

Государственное научно-образовательное учреждение
Факультет вычислительной математики и кибернетики
Московского государственного университета имени М.В.Ломоносова

УДК _____
ГРНТИ _____
№ госрегистрации _____
Инв. № _____

УТВЕРЖДАЮ
Руководитель НИР
_____ Смелянский Р.Л.
«_____» _____ 2010 г.

ОТЧЕТ
О ПАТЕНТНЫХ ИССЛЕДОВАНИЯХ

по теме:

«Создание прототипа интегрированной среды и методов комплексного анализа функционирования распределённых вычислительных систем реального времени (РВС РВ)»

Шифр «2010-1.1-215-138»

Ответственный за проведение
патентного исследования

личная подпись

Савенков
К.О.

расшифровка
подписи

дата

Москва, 2010

СПИСОК ОСНОВНЫХ ИСПОЛНИТЕЛЕЙ

Руководитель группы	_____	_____	Савенков К.О. (разделы 1,2)
	подпись	дата	
<i>Исполнители темы:</i>			
1. Тематический поиск документов	_____	Дата	Савенков К.О. (раздел 1)
	подпись	Дата	
2. Исследование новизны и аналогов разрабатываемого объекта	_____	Дата	Антоненко Виталий Александрович (раздел 2)
	подпись	Дата	Пашков Василий Николаевич (раздел 2)
	_____	_____	
	_____	_____	

РЕФЕРАТ

Отчет 119 с., 1 ч., 31 рис., 27 табл., 4 прил.

Патентные исследования, патентный поиск, комплексный анализ функционирования, распределённые вычислительные системы, интегрированная среда разработки, имитационное моделирование.

Проект: «Создание прототипа интегрированной среды и методов комплексного анализа функционирования распределённых вычислительных систем реального времени (РВС РВ)».

Объектом исследования являются программы для ЭВМ, используемые для анализа функционирования РВС РВ, а также обеспечивающие процесс комплексного анализа функционирования РВС РВ в составе интегрированной среды.

Задачи патентных исследований: исследование направлений научно-исследовательской и производственной деятельности организаций и фирм, применяющих анализ функционирования РВС РВ в своих производственных процессах; выявление ближайших существующих аналогов разрабатываемых программных средств; исследование охраноспособности объекта.

Виды патентных исследований:

1. Тематический поиск публикаций о результатах научно-исследовательских работ и информации о существующих программных средствах, охраняемых документах.
2. Исследование новизны и поиск аналогов разрабатываемого объекта.

Патентные исследования проводились в соответствии с условиями, указанными в Регламенте поиска №1.

В рамках настоящего патентного исследования произведен тематический поиск охраняемых документов (свидетельств о государственной регистрации программы для ЭВМ), публикаций о результатах научно-исследовательских работ и информации о существующих программных средствах. В результате выявлены программы для ЭВМ, наиболее близкие к разрабатываемому объекту и его составным частям.

Анализ и сопоставление программ для ЭВМ, являющихся наиболее близкими аналогами разрабатываемого объекта, позволил сформировать группу отличительных признаков предлагаемого программного решения, а именно:

- узкая специализация на предметной области – анализ поведения распределённых вычислительных систем реального времени, с учётом

особенностей, связанных с обеспечением функционирования, мониторинга и анализа поведения таких систем;

- возможность организации программных средств в интегрированную среду, обеспечиваемая согласованными (или настраиваемыми) интерфейсами обмена данными;
- открытость исходного кода, что позволит добиться широкого распространения программного продукта и обеспечит возможность его модернизации (создания новых версий программ) под конкретные РВС РВ;
- возможность работы под операционной системой (Linux), распространяемой на основе «открытой» лицензии, что позволит пользователям экономить существенные денежные средства на приобретение программного обеспечения.

Наличие данных отличительных особенностей подчеркивает новизну разрабатываемого объекта, что свидетельствует о его охраноспособности.

Ряд выявленных программных средств, в силу лицензии, под которой выполняется их распространение, могут быть использованы в составе разрабатываемого объекта.

Таким образом настоящее патентное исследование проведено в полном соответствии с Заданием №1. Задачи, поставленные перед патентными исследователями, выполнены.

СОДЕРЖАНИЕ

ОБЩИЕ ДАННЫЕ ОБ ОБЪЕКТЕ ИССЛЕДОВАНИЙ.....	7
ОСНОВНАЯ ЧАСТЬ.....	9
1 Тематический поиск патентной документации	9
2 Исследование новизны разрабатываемого объекта.....	10
ЗАКЛЮЧЕНИЕ	42
ПРИЛОЖЕНИЕ А – Задание № 1 на проведение патентных исследований.....	48
ПРИЛОЖЕНИЕ Б – Регламент поиска №1	50
ПРИЛОЖЕНИЕ В – Отчет о поиске	51
ПРИЛОЖЕНИЕ Г – Дополнительные информационные материалы	59

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем отчете о патентных исследованиях применяются следующие термины, обозначения и сокращения с соответствующими определениями:

- Программа для ЭВМ - представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств в целях получения определенного результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения (ГК РФ часть 4 ст. 1261);
- РВС РВ - распределённая вычислительная система (системы) реального времени;
- РИМ - распределённое имитационное моделирование;
- БРЭО - бортовое радиоэлектронное оборудование;
- ЭО ТПК - экспериментальный образец типового программного комплекса;
- СПМ - стенд полунатурного моделирования;
- ИФТ - интеграционное и функциональное тестирование;

ОБЩИЕ ДАННЫЕ ОБ ОБЪЕКТЕ ИССЛЕДОВАНИЙ

В качестве объекта исследования выступают программы для ЭВМ, используемые для анализа функционирования РВС РВ, а также обеспечивающие процесс комплексного анализа функционирования РВС РВ в составе интегрированной среды.

Период проведения работ: 20 сентября 2010 года – 30 ноября 2010 года

Область применения: создаваемый ЭО ТПК предназначен для комплексного анализа функционирования РВС РВ.

Современные распределённые вычислительные системы реального времени (РВС РВ), и бортовые вычислительные системы в особенности, являются чрезвычайно сложными, многокомпонентными объектами [1]. На практике разработка компонентов РВС РВ происходит одновременно, распределённо и выполняется различными организациями. При этом комплексирование (сборочное тестирование) необходимо проводить на всех стадиях готовности РВС РВ [1]. Это позволяет своевременно обнаруживать и исправлять программные и аппаратные ошибки, включая выбор аппаратной конфигурации РВС РВ, уменьшая тем самым срок и стоимость разработки РВС РВ.

Для отработки совместного функционирования компонентов РВС РВ необходимы инструментальные средства, охватывающие все стадии разработки и интеграции бортового оборудования. Одной из важных задач таких средств является комплексный анализ функционирования РВС РВ, в том числе – анализ поведения РВС РВ в ходе совместного функционирования на этапе, когда многие компоненты системы еще не изготовлены. Указанная задача требует применения средств моделирования, позволяющих организовать работу РВС РВ в отсутствие натуральных образцов ряда компонентов и подыграть необходимые для проведения эксперимента условия окружающей среды. В случае разработки моделей компонентов РВС РВ поставщиками компонентов, как правило, приходится иметь дело с разнородными моделями. Это означает, что должен быть разработан язык спецификации интерфейсов таких моделей для задач совместного распределённого исполнения.

Средства комплексного анализа поведения РВС РВ должны, тем самым, обеспечивать решение следующих задач:

1. **Спецификация интерфейсов моделей компонентов РВС РВ.** Помимо спецификации интерфейса обмена модельными данными средства должны позволять задавать интерфейс распределённого имитационного моделирования: обмена событиями и синхронизации времени. Для обеспечения возможности отладки необходимы также средства описания интерфейса удалённой отладки компонентов РВС РВ (доступа к внутреннему состоянию компонента).
2. **Распределённое совместное имитационное моделирование (РИМ), допускающее, в том числе, совместное выполнение имитационных моделей и натуральных компонентов РВС РВ.** Из последнего следует, что средства РИМ должны обеспечивать возможность моделирования в реальном времени, т.е. с гарантированно низкими затратами на обмен событиями между моделями.
3. **Обеспечение хранения и доступа к поведению РВС РВ, записанному в ходе функционирования.** В таких случаях принято говорить о трассах поведения – последовательностях событий с параметрами и временными метками. Подобные средства должны обеспечивать высокую скорость записи событий (иначе при проведении эксперимента в реальном времени часть событий может пропасть), компактное представление трассы и эффективный доступ к её элементам. Последнее требует применения технологий семантического сжатия информации и поддержание системы индексов доступа, поскольку объём даже сжатой трассы может измеряться терабайтами.
4. **Анализ трасс для задач проверки выполнения требований к РВС РВ.** Средства анализа трасс должны обеспечивать возможность автоматического и интерактивного анализа, включая проверку темпоральных спецификаций, навигацию, поиск и визуализацию.

ОСНОВНАЯ ЧАСТЬ

В соответствии с задачами патентного поиска (Приложение А) проведены мероприятия:

1. тематический поиск патентной документации, публикаций о результатах научно-исследовательских работ и информации о существующих программных средствах;
2. выполнен обзор направлений научно-исследовательской и производственной деятельности организаций и фирм, применяющих анализ функционирования РВС РВ в своих производственных процессах;
3. проведён анализ существующих программных средств, применяемых в ходе анализа функционирования РВС РВ;
4. проведён анализ охраноспособности программных средств для комплексного анализа функционирования РВС РВ.

1 Тематический поиск патентной документации и публикаций

В результате произведенного тематического поиска в соответствии с Регламентом поиска №1 (Приложение Б) за период с 2000 по 2010 года выявлен 71 документ отвечающих по ключевым параметрам тематическим условиям поиска (см. Список литературы и приложение Г.5):

13 охранных документов (свидетельств о государственной регистрации программы для ЭВМ),

9 отчётов о проведении научно-исследовательских работ,

36 публикаций о результатах научно-исследовательских работ,

13 источников с описанием разработок в сети Интернет.

Из них для дальнейшего анализа отобрано 69 документов (Таблица В.6.1 и Таблица В.6.2 Приложение В). Расширенные материалы о программах для ЭВМ, выбранных для дальнейшего анализа, а также документы, которые не вошли в выборку, приводятся в Приложении Г, т.к. могут представлять практический интерес для разработчика ЭО ТПК.

В результате анализа документов выявлен ряд программных средств, являющихся аналогами разрабатываемого программного средства. Ряд выявленных программных средств, в силу лицензии, под которой выполняется их распространение, могут быть использованы в составе разрабатываемого объекта. Подробности и результаты анализа представлены в разделе 2 настоящего отчета.

2 Исследование новизны и существующих аналогов разрабатываемого объекта

Исследование новизны разрабатываемого объекта осуществляется посредством анализа и сопоставления программ для ЭВМ, являющихся наиболее близкими аналогами разрабатываемого объекта, а также изучения основных направлений научно-исследовательской деятельности организаций, применяющих анализ поведения РВС РВ в своих производственных процессах.

Исследование выполнено по 5 направлениям:

1. Существующие средства комплексного анализа поведения РВС РВ (раздел 2.1),
2. Методы спецификации интерфейсов моделей компонентов РВС РВ (раздел 2.2),
3. Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ (раздел 2.3),
4. Средства записи и хранения информации о поведении РВС РВ (раздел 2.4),
5. Средства анализа данных о поведении РВС РВ (раздел 2.5).

2.1 Существующие средства комплексного анализа поведения РВС РВ

В настоящем разделе представлены результаты анализа существующих средств комплексного анализа поведения РВС РВ. Как правило, данные средства применяются в ходе тестирования РВС РВ для сопоставления наблюдаемого поведения РВС РВ предъявляемым требованиям.

2.1.1 Средства интеграционного и функционального тестирования

К большинству РВС РВ предъявляются повышенные требования к безопасности и производительности, в силу того, что в большинстве случаев цена дефектов и отказов в таких системах очень велика (вред здоровью или гибель людей, утрата дорогостоящего оборудования и т.д.). Поэтому в цикле разработки РВС РВ одними из важнейших являются этапы функционального и интеграционного тестирования.

Под функциональным тестированием (ФТ) РВС РВ подразумевается проверка соответствия системы функциональным требованиям, то есть её способностей в определенных условиях выполнять необходимые функции. Завершающим этапом ФТ является интеграционное тестирование (ИТ), предполагающее тестирование и анализ производительности группы компонентов или же всей системы в целом. Основной целью ИТ является нахождение дефектов интерфейсов между модулями. Важнейшее значение

данный вид тестирования имеет на этапе комплексирования РВС РВ – сборки отдельных модулей системы вместе.

Основными и традиционными методами проверки РВС РВ являются имитационное и полунатурное моделирование. Соответственно, средства интеграционного и функционального тестирования (ИФТ) РВС РВ неразрывно связаны со средствами, используемыми на этих этапах моделирования.

Выделим основные задачи ИФТ, возникающие на этапе комплексирования и требующие инструментальной поддержки [1]:

- проверка соответствия приборов РВС РВ требованиям технического задания, в том числе, в части приёма и передачи данных по внешним интерфейсам;
- отработка взаимодействия между приборами РВС РВ по каналам передачи данных;
- комплексное тестирование и отладка ПО РВС РВ, в том числе, ПО, выполняемого распределённо на различных узлах;
- оценка надёжности архитектуры РВС РВ, в том числе, наличия резерва пропускной способности каналов передачи данных и устойчивость аппаратно-программных средств РВС РВ к сбоям при передаче данных;
- построение расписаний обмена данными по бортовым каналам, а также проверка правильности отработки этого расписания приборами в составе РВС РВ.

В зависимости от степени готовности приборов РВС РВ, в составе стенда моделирования могут использоваться полностью программные имитационные модели всех приборов, либо комплекс из натуральных образцов и программных моделей, собранных в единый стенд и сопряжённых через аппаратные каналы бортовых интерфейсов.

Также может варьироваться уровень детальности моделирования - от так называемых "интервальных" моделей, которые отрабатывают заданные циклограммы обменов, не выполняя вычисление передаваемых прибором данных, до полных функциональных моделей, эквивалентных реальным приборам по составу и значениям выдаваемых в бортовые каналы данных и включающих в свой состав реальное ПО приборов.

Вследствие сложности и указанной специфики разработки большинства РВС РВ, средства, осуществляющие инструментальную поддержку их ИФТ, также представляют собой весьма нетривиальные системы, которые должны удовлетворять ряду критериев.

2.1.2 Критерии сравнения средств ИФТ

В силу того что ИФТ встроенных систем реального времени неразрывно связано с процессами моделирования и комплексирования таких систем, приведенные ниже критерии сравнения в значительной мере составлены в соответствии с этим фактом.

Моделирование поведения системы в реальном времени. Большинство коммерческих средств, предоставляющих решения в области моделирования и интеграции РВС РВ, действуют по методу имитации и анализа поведения системы в реальном времени [2] [3] [4] [5] [6]. Подобные средства представляют наибольший интерес, т.к. ориентируются в основном на полунатурное моделирование и пока дают наибольшие гарантии того что реальная система будет соответствовать заданным требованиям к качеству, отказоустойчивости, производительности и безопасности. Однако, существует ряд подходов [7] [8] позволяющих осуществить верификацию чисто формальными методами без какого-либо моделирования в реальном времени. Подобные методы могут быть исключительно полезны на самых ранних стадиях разработки РВС РВ, когда реальные компоненты РВС РВ еще не готовы и не учитывают многих тонкостей которые возникают на более поздних этапах. В данной работе основной целью рассмотрения являются средства полунатурного моделирования, поэтому мы не будем детально рассматривать данные формальные методы.

Возможность и способы интеграции моделей с реальными компонентами. Полунатурное моделирование подразумевают возможность интеграции с реальными компонентами по мере их поступления. Все рассматриваемые далее системы удовлетворяют данному критерию.

Комплексное имитационное моделирование РВС РВ. Возможность имитации поведения РВС РВ как единого целого. На этапе интеграции, это позволяет обнаружить многие проблемы, воспроизводимые исключительно в масштабах всей системы: проблемы с производительностью отдельных компонентов, ошибки в протоколах передачи данных между устройствами, ошибки в интерфейсах по которым взаимодействуют отдельные компоненты и т.п. На стадии математического моделирования используются чисто программная имитация поведения отдельных модулей, на стадии полунатурного – часть

модулей заменяется реальными устройствами. Все рассматриваемые системы поддерживают данную возможность.

Поддержка постепенной замены математических моделей реальными модулями PBC

PВ. Вследствие того, что реальные модули разрабатываемой PBC PВ поступают отдельно, по мере их готовности, а организация и калибровка среды моделирования поведения распределенной системы представляет собой достаточно сложную задачу, наличие возможностей прозрачной замены математических моделей на реальные компоненты существенно сокращает затраты на реорганизацию. На сегодняшний день невыполнение этого критерия в какой-либо из систем полунатурного моделирования существенно бы сократило возможность её применимости на практике. В результате, все рассматриваемые системы позволяют заменить математическую модель реальным компонентом и отличаются лишь удобством конфигурирования параметров соединения с подключаемым модулем.

Как правило, обеспечение совместимости интерфейсов ввода/вывода между симулятором и подключаемым устройством реализуется с использованием технологии FPGA, и наличие средств быстрого и удобного перепрограммирования FPGA-чипов существенно упрощает подключение нового устройства. Все рассматриваемые средства, кроме [3] используют для этих целей основную исходную среду разработки моделей: Simulink [2] [4] или LabVIEW [5] [6].

Стандартные библиотеки компонентов для определённых отраслей. В силу того, что каждая область в которой применяется PBC PВ имеет свою специфику (протоколы передачи данных, ориентированные на определенный вид вычислений контроллеры, и др.), от которой зависит сложность разработки математических моделей и сложность вычислений в результирующем коде, большое значение имеет наличие специфических библиотек, упрощающих разработку моделей и обеспечивающих генерацию эффективного кода для целевых архитектур.

Наибольшее количество библиотек для этих целей предоставляет компания Oral-RT [2]. Среди них библиотеки для нужд электроэнергетики, авиации, моделирования автомобильных электродвигателей и др. Наличие данных библиотек, повышающих производительность позволяет выполнять моделирование поведения PBC PВ даже на кластере состоящем из обычных ПК. Это является одним из основных достоинств решений компании Oral-RT по сравнению с аналогами. Компания NLR предоставляет широкие средства поддержки моделирования PBC PВ в области авиации [3]. Компания

dSPACE – в области моделирования встроенных автомобильных систем [4]. National Instruments – в обеих этих областях [6]. Недостатком является то что, все эти компании (и их программное обеспечение) ориентированы исключительно на специализированные, весьма дорогие, аппаратные средства (симуляторы).

Среда разработки моделей. Процесс разработки РВС РВ в большинстве случаев строится на моделировании (model-based development). Лидирующее положение на рынке систем разработки математических моделей в данный момент занимает MathWorks® Simulink™. В результате, все рассматриваемые в данном обзоре средства в той или иной мере интегрированы с этим программным продуктом, позволяющим разрабатывать модели графическим способом. Конкурентным аналогом Simulink является графическая среда разработки моделей National Instruments SystemBuild™, интеграция с которой поддерживается в [2], [5], [6]. Однако, во многих проектах (например военной авиации) не используются подобные средства и процесс математического моделирования построен иначе, с ориентацией в большей степени на конечный исходный код компонентов РВС РВ. Среди всех рассмотренных средств, работу с реальным исходным кодом как с моделью поддерживает только ADI ADvantage Framework [5]. Тем не менее, указанное и все остальные средства позволяют осуществить генерацию кода, например, модели Simulink (используя MathWorks RTW™) и загружать этот код на реальные модули РВС РВ. При полунатурном моделировании это позволяет быстро заменить математическую модель реальным устройством при его поступлении.

Основной пользовательский интерфейс системы выполнения имитационного эксперимента. Важнейшими компонентами пользовательского интерфейса системы, осуществляющей поддержку полунатурного моделирования, являются: средства визуализации процесса моделирования и средства для распределения частных моделей и вычислений по узлам распределенной системы. От степени развитости этих средств напрямую зависит удобство и эффективность проведения имитационного эксперимента.

Наиболее развитые средства визуализации и мониторинга для полунатурного моделирования предоставляет инструмент NI LabVIEW™, который используется как стандартное средство для этих целей во всех решениях компании National Instruments [6], а также за счет возможностей интеграции может быть использован в [2] [5]. LabVIEW по сути представляет собой виртуальный прибор (англ. virtual Instrument) и состоит из двух частей:

- блочной диаграммы, описывающей логику работы виртуального прибора;

- лицевой панели, описывающей внешний интерфейс виртуального прибора.

Виртуальные приборы могут использоваться в качестве составных частей для построения других виртуальных приборов. Графический язык программирования «G», используемый в LabVIEW, основан на архитектуре потоков данных. Последовательность выполнения операторов в подобных языках определяется не порядком их следования (как в императивных языках программирования), а наличием данных на входах этих операторов. Операторы, не связанные по данным, выполняются параллельно в произвольном порядке [6].

Следует также отметить, что компании Opal-RT [2], NLR [3] и dSPACE [4] предоставляют специализированные программные средства для расширенной поддержки определенных областей (авиация, автомобилестроение и др.). Наличие таких средств обеспечивает на порядок большее число успешных проектов по интеграции, выполненных при участии данных компаний в этих конкретных областях. Сходные с LabVIEW возможности предоставляет среда ADI ADvantageVI [5], однако компания ADI не предоставляет специальных библиотек для конкретных областей, что существенно увеличивает трудозатраты на разработку среды визуализации.

Рассматривая вопрос наличия средств для распределения частных моделей по вычислительным узлам распределенной системы, необходимо отметить, что большинство рассматриваемых по данному критерию оставляют желать лучшего. Относительно прозрачную схему создания карты вычислений обеспечивают [2] [6]. Наиболее удобные же средства для указанных целей имеются только у компании ADI [5] в составе среды ADvantageDE. В неоднородных вычислительных средах наличие таких средств является весьма существенным достоинством, которое позволяет быстро подобрать максимально эффективную карту вычислений.

Наличие средств автоматизации ИФТ и анализа. В силу того что эффективное ИФТ и анализ таких сложных систем как РВС РВ практически невозможно без средств автоматизации данного процесса. По отношению к данному критерию, все рассматриваемые среды поддержки полунатурного моделирования можно разделить на две группы:

- предоставляющие API доступа к среде имитационного моделирования и ориентированные в первую очередь на средства тестирования третьих сторон,
- предоставляющие собственные интегрированные средства автоматизации тестирования.

К первой группе относится средство [2], ко второй все остальные средства. Средство ADvantage [5] относится к обеим группам в равной мере.

Решения компании Opal-RT [2] предоставляют лишь интерфейс для взаимодействия с тестовыми средами третьих сторон, написанными на языке Python и, в целом, отличаются скудностью возможностей автоматизации тестирования.

Продукт TVE компании NLR [3] предоставляет очень удобные средства тестирования с использованием сценариев на специальном языке PLUTO, однако их достоинством и одновременно недостатком является ориентированность исключительно на область тестирования авиационных РВС РВ.

Средства компаний dSPACE и NI предоставляют средства создания графических параметризованных тестовых последовательностей в рамках единой методологии разработки на основе моделей [4] [6]. В ряде случаев это оказывается весьма удобным решением, однако, как уже упоминалось выше, данная парадигма разработки РВС РВ применяется далеко не всегда, что существенно сужает область применимости подобных средств.

Наиболее развитые средства автоматизации ИФТ предоставляет компания dSPACE что является её бесспорным достоинством [4]. Решения dSPACE позволяют полностью автоматизировать процесс ИФТ, за счет создания тестовых последовательностей, которые могут быть запущены и выполняться без какого-либо участия человека, осуществляя сбор необходимых данных, получаемых в ходе эксперимента. Это становится особенно полезным для методов регрессионного тестирования. Кроме того dSPACE позволяет связать средства управления требованиями (DOORS™ или Excel™) со средствами ИФТ для отслеживания выполнимости требований и тестового покрытия.

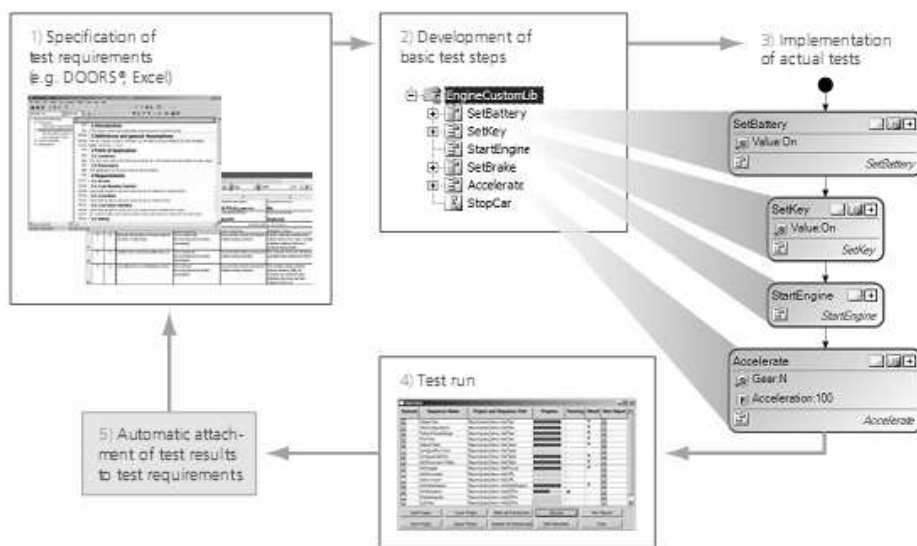


Рисунок 1. Подход к ИФТ компании dSPACE.

Сходные по своей функциональности возможности предоставляет компания NI [6] за счет использования NI TestStand™ вкуче с NI Requirements Gateway™.

Все рассмотренные средства, кроме [2] и [3], предоставляют возможности создания конфигурируемых отчетов о выполнении тестов в формате XML. Наиболее развитые средства создания отчетов предоставляет компания dSPACE [4].

Для оценки производительности и хода эксперимента постфактум, во всех рассмотренных средствах (кроме NLR TVE [3]) предоставляются возможности логирования и сбора трасс при выполнении кода. В силу закрытости рассмотренных средств, формат трасс во всех них неизвестен и детальные описания возможностей трассировки отсутствуют.

Параметры целевого симулятора (комплекса аппаратных средств) и связанных с ним программных возможностей. Базовые целевые архитектуры всех рассмотренных средств (кроме [3]) имеют сходный вид и могут быть представлены в виде изображенном на рисунке 2.

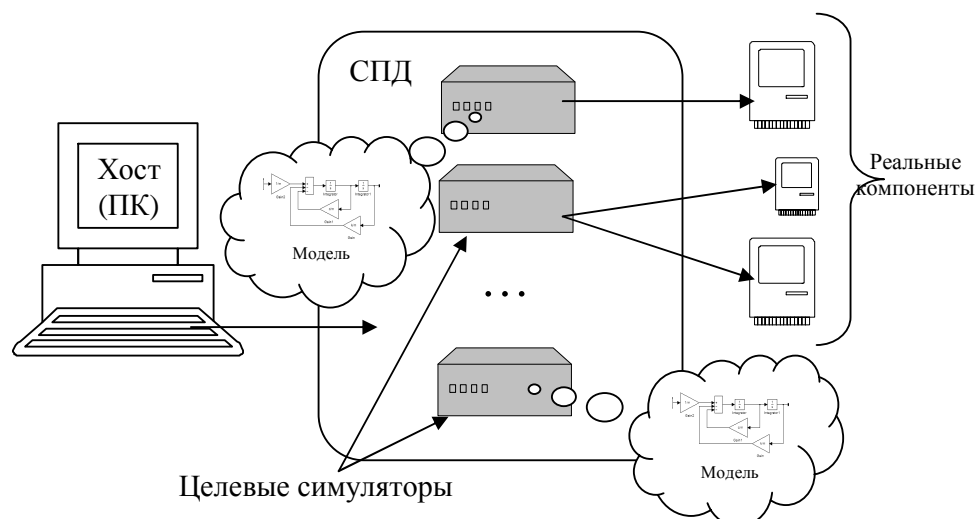


Рисунок 2. Базовая целевая архитектура рассматриваемых средств полунатурного моделирования.

Целевые симуляторы, как правило, представляют собой поставляемые соответствующими компаниями многопроцессорные системы, использующие FPGA на общей памяти, имеющие в своем составе специальные интерфейсы ввода/вывода, аппаратные средства имитации отказов, и имеющие возможность объединения группы таких симуляторов в единую распределенную систему (как правило, при помощи Ethernet). В качестве целевой операционной системы используется QNX [2][5], VME [3][5], OSEK[4], PXI RTOS [6]. Характеристики среды выполнения моделей и коммуникационного интерфейса рассмотренных проприетарных решений в открытом доступе не имеются.

Недостатком данных симуляторов является то, что они представляют собой весьма дорогие устройства, и их использование не всегда оправдано. Поддержку проведения полунатурного моделирования на кластерах из разнородных машин осуществляют только Opal-RT [2] и ADI [5].

2.1.3 Сводная таблица результатов анализа

Сравнение рассмотренных средств по всем перечисленным критериям представлено в сводной таблице 1.

#	Критерий	OPAL-RT (RT-LAB™, eMEGAsim™ etc.)	NLR (Test and Verification Equipment for Spacecraft)	dSPACE (dSPACE Simulator)	ADI (The ADvantage Framework)	NI (NI HIL Test Platform)	ЛБК (Dyana)
1	Моделирование поведения системы в	+	+	+	+	+	+

	реальном времени						
2	Способы (и вообще возможность) интеграции моделей с аппаратурой	+	+	+	+	+	+
3	Комплексное имитационное моделирование РВС РВ	+	+	+	+	+	+
4	Поддержка постепенной замены математических моделей реальными модулями РВС РВ.	+	+	+	+	+	+
5	Стандартные библиотеки компонентов для определённых отраслей	+	+(авионика)	+(автомобили)	-	+/- (авионика, автомобили, поставляются вместе с аппаратными средствами)	-
6	Среда разработки моделей	-	-	-	-	SystemBuild, LabVIEW	+
6.1	Язык моделирования или способ задания моделей	Графический	Графический	Графический	Графический/код	Графический	Сиподобный язык
6.2	Интеграция с другими средствами разработки	+(Simulink, SystemBuild)	+(Simulink)	+(Simulink)	+(Simulink, SystemBuild, C/C++, Ada, Fortran)	+(Simulink)	-
6.3	Имитационное моделирование с подключением исходного кода	-	-	+/- (код может выполняться только на спец. симуляторе)	+	-	+
6.4	Возможность генерации кода по модели и перенос сгенерированного кода в реальные компоненты	+(RTW)	+(RTW)	+(RTW)	+(RTW)	+(LabVIEW C Generator / RTW)	-
7	Основной ГПИ системы выполнения имитационного эксперимента	+	+(eurosим)	+(ControlDesk)	+(ADvantage)	+(NI VeriStand)	+
7.1	Наличие средств для распределения симулируемых моделей и	+	+/-	-	+(ADvantageDE)	+	+/-

	вычислений по узлам распределенной системы							
7.2	Средства визуализации процесса моделирования	+	+	+	+	+	+	+
		(LabVIEW, RT-SCOPE)	(SCOS-2000)	(ModelDesk)	(ADvantageVI, SIMplotter, LabVIEW)	(NI VeriStand, LabVIEW)		
8	Наличие средств автоматизации ИФТ и анализа		+	+	+	+	+	+/-
				(AutomationDesk)	(ADvantageVI)	(NI TestStand, LabVIEW)		
8.1	Формат/язык входных данных тестов		Сценарий на языке PLUTO, использующий набор базовых операций для авионики	Графический	Сценарии на большом количестве поддерживаемых языков: Python, C/C++, Cosint	Графический	Сиподобный язык (то же что и для разработок и моделей)	
		-						
8.2	Средства для разработки параметризованных тестовых последовательностей		+	+	+	+	+	+/-
8.3	Наличие средств создания отчетов о выполнении тестов		+/-	+	+	+	+	+
8.4	Трассировка при выполнении кода для оценки производительности	+	-	+	+	+	+	+
8.5	Открытая среда для интеграции со средствами тестирования третьих сторон	+	-	+	+	+	+	-
		(Python API)						
9	Целевой симулятор (комплекс аппаратных средств) и связанные с ним программные возможности	+	+	+	+	+	+	+/-

9.1	Целевая архитектура (ПК / Многоядерная система на общей памяти (ОП) / Распределенная система (РП))	ПК + Один или несколько eMEGAsim, eDRIVEsim симуляторов (или обычных ПК) + набор различных аппаратных средств в зависимости от задач;	VME-фронтенд + 2 хоста для исполнения моделей (ПК) + устройства моделирующие бортовые интерфейсы	ПК + Один или несколько dSPACE Simulatorов + набор различных аппаратных средств в зависимости от задач)	ПК + Один или несколько RTS Simulatorов + набор различных аппаратных средств в зависимости от задач Кластер ПК	ПК + Один или несколько NI PXI Simulatorов + набор различных аппаратных средств в зависимости от задач	Кластер ПК
9.2	Полунатурное моделирование, включающее набор реальных устройств и моделей реальных устройств	+	+	+	+	+	+
9.3	Поддержка стандартных технологий, таких как PCI Express и новейших архитектур процессоров и памяти	+	-	+/- (используется исключительно специальный симулятор dSPACE)	+/- (используется исключительно специальный симулятор ADI RTS)	+/- (используется исключительно специальный симулятор NI PXI)	-
9.4	Поддержка FPGA и ориентация на него	+	-	+	+	+	-

2.1.4 Выводы

Как показали результаты обзора, в настоящее время существует ряд средств ИФТ, обладающих сходными характеристиками и областью применения. Ни одно из этих средств не является программным средством с открытым исходным кодом. Тем самым, полных аналогов разрабатываемых средств не выявлено.

Сравнение характеристик программного средства «Стенд ПНМ», разработанного в ЛВК ВМК МГУ имени М.В.Ломоносова, с его ближайшими аналогами показывает, что характеристики данного программного средства отвечают современным требованиям. В то же время, средства, существующие на рынке, ориентированы на интеграцию с распространёнными средствами разработки моделей, синтеза кода и системами автоматического тестирования.

Таким образом, средства «Стенда ПНМ» подходят в качестве основы для разработки прототипа ПО для комплексного анализа поведения РВС РВ с открытым кодом. При этом необходимо проанализировать ПО третьих сторон, применяемые потенциальными пользователями в своих производственных процессах и либо обеспечить интеграцию с ними разрабатываемого ПО, либо предусмотреть возможность такой интеграции в ходе разработки интерфейсов ПО.

2.2 Методы спецификации интерфейсов моделей компонентов РВС РВ

В настоящее время существует единственный стандарт, разработанный для описания интерфейсов моделей для распределённого гетерогенного моделирования. Таким стандартом является High Level Architecture (HLA). Архитектура HLA была разработана по заказу Министерства обороны США для унификации и повторного использования моделей, применяемых в военных целях. В настоящее время эта архитектура стандартизована институтом IEEE (стандарт 1516) [9]. HLA представляет объектно-ориентированную систему, которая может применяться для построения моделей на различных языках, поддерживающих концепцию ООП (C++, Java и т.п.).

Модель, основанная на архитектуре HLA, называется федерацией и состоит из независимых федератов – основных строительных блоков. Требования HLA ограничивают возможности взаимодействия между федератами только средствами, предоставляемыми HLA RTI (Run-Time Infrastructure). Из этого следует, что всё необходимое для работы отдельно взятого федерата в составе любой модели может быть описано и документировано в терминах, определяемых стандартом HLA и не должно зависеть от конкретной реализации RTI. HLA не накладывает никаких ограничений на внутреннюю

организацию федератов – каждый из них представляет из себя независимо исполняемую программу или аппаратуру, реализующую требования архитектуры.

Для взаимодействия между федератами HLA предоставляет два механизма: экземпляры объектов (objects) с атрибутами (attributes) и взаимодействия (interactions) с набором параметров (parameters). С точки зрения потоков информации, федераты задают поведение объектов через изменение их свойств-атрибутов, устанавливают зависимости между свойствами различных объектов. Объекты HLA логически отражают появление, существование и исчезновение в моделируемой среде объектов реального мира. Взаимодействия служат для представления событий, не связанных с конкретными объектами моделируемой системы, но оказывающих влияние на федераты. Весь обмен информацией в федерации осуществляется через централизованный процесс, реализующий службы RTI.

На сегодняшний день существует множество реализаций RTI в той или иной степени соответствующей стандарту HLA IEEE P1516 (см. раздел 3). Задача данной части исследования – определить степень реализации интерфейсов подключения моделей в реализациях HLA с открытым исходным кодом, лицензия распространения которых позволяет их использование в качестве составной части другого ПО (CERTI [10], PORTICO [11]). Для сравнения также приводятся данные по наиболее популярной коммерческой HLA (МАК [12]).

2.2.1 Базовые понятия HLA

В данной главе подробнее опишем интерфейсы FOM, исходя из стандарта HLA.

Для спецификации как интерфейсов между федератами и RTI, так и служб RTI, необходимо точное и строгое определение концепции объектной модели. Правила и терминология, используемые для описания объектной модели федерации (federation object model FOM), описаны в документе High Level Architecture, IEEE P1516.2 [13]. Объектная модель имитационного моделирования (simulation object model - SOM) описывает существенные характеристики федерата с целью его повторного использования и другой деятельности, связанной с внутренними подробностями его функционирования. Модель SOM, как таковая, не используется RTI и ее службами. В отличие от SOM, модель FOM имеет дело с взаимодействием федератов и используется инфраструктурой RTI.

Модель FOM определяет:

- набор классов объектов, представляющих моделируемый мир в создаваемой федерации;
- набор классов взаимодействий, выбранных для представления взаимодействий между объектами моделируемого мира;
- атрибуты и параметры упомянутых классов;
- уровень детальности, на котором эти классы представляют моделируемый мир.

Любой объект должен быть экземпляром класса объектов, присутствующего в модели FOM. Классы объектов выбираются разработчиком объектной модели. Каждый класс объектов имеет связанный с ним набор атрибутов.

С точки зрения федерации, набор значений атрибутов некоторого объекта полностью определяет состояние этого объекта. Федераты имеют право связывать с объектом дополнительную информацию о состоянии, которая не подлежит передаче между федератами, это вне компетенции объектной модели федерации.

Федераты используют состояние объектов как одно из основных средств взаимодействия. В любой момент времени только один федерат должен отвечать за изменение заданного атрибута объекта. Такой федерат обязан предоставлять другим федератам новые значения этого атрибута объекта посредством служб RTI.

Федерат, поставляющий новые значения атрибута объекта, называется *обновляющим* значение этого атрибута. Федераты, получающие эти значения, называются *отражающими* этот атрибут.

В любой момент времени право на обновление значения атрибута объекта должно сохраняться за единственным федератом. Федерат, имеющий право на обновление значения атрибута объекта, называется *владеющим* этим атрибутом. Инфраструктура RTI предоставляет службы, позволяющие федератам передавать владение атрибутами объектов.

Концепция модели FOM также позволяет определять в объектной модели классы взаимодействий. Типы возможных взаимодействий и их параметры специфицируются в FOM.

Федерация - это объединение конкретной модели FOM, конкретного набора федератов и служб RTI. Федерация создается для конкретной цели на основе единообразно понимаемой объектной модели и набора федератов, которые могут связывать свою специфичную семантику с этой объектной моделью.

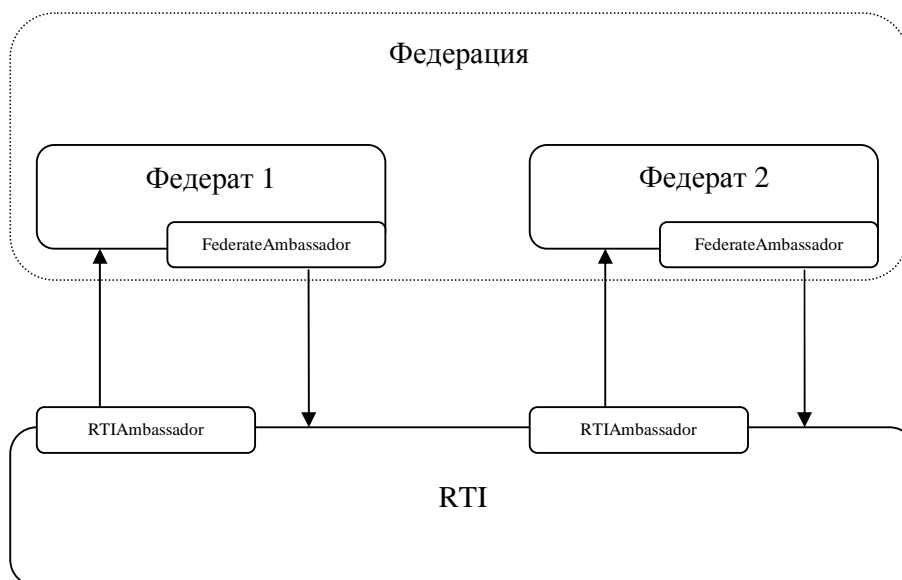


Рисунок № 3 Схема взаимодействия федератов и RTI.

Далее описываются стандартизированные интерфейсы для взаимодействия федератов в федерации и приведена схема жизненного цикла федерата (см. рис.3).

2.2.2 Критерии сравнения

Требования стандарта HLA сформулированы в виде набора сервисов, которые должны предоставлять компоненты гетерогенной имитационной модели и описания объёма предоставления данных сервисов. Сравнение упомянутых выше реализаций HLA предлагается выполнять на основе набора критериев, соответствующих данным сервиса. Ниже приводится список сервисов, выступающих в качестве критериев, с пояснениями.

Исполнение процесса федерации

Create Federation Execution

Служба *Create Federation Execution* создает процесс исполнения федерации и добавляет его ко множеству поддерживаемых процессов исполнения федерации. Каждый процесс исполнения федерации, созданный этой службой, должен быть независимым от остальных процессов исполнения федерации, а также между процессами исполнения федерации не должно быть взаимодействия через инфраструктуру RTI. Параметр “описатель FED” идентифицирует данные FED, которые требуются для создания процесса исполнения федерации.

Destroy Federation Execution

Служба Destroy Federation execution удаляет процесс исполнения федерации из множества процессов исполнения федерации, поддерживаемых инфраструктурой RTI. Перед вызовом этой службы все операции федерации должны быть завершены, и все федераты должны отсоединиться.

Join Federation Execution

Служба Join Federation Execution присоединяет федерат к процессу исполнения федерации. Вызов службы Join Federation Execution должен выражать намерение участвовать в заданной федерации. Параметр «тип федерата» различает категории федератов для целей сохранения/восстановления федераций. Возвращаемый описатель федерата должен быть уникальным для всех федератов в процессе исполнения федерации.

Resign Federation Execution

Служба Resign Federation Execution запрашивает прекращение участия в федерации. Перед отсоединением должно быть принято решение о владении атрибутами объекта, принадлежащими федерату. Федерат может передать владение такими атрибутами объекта другим федератам, освободить их для приобретения прав владения позднее, или удалить объект, частью которого являются атрибуты (при условии достаточных привилегий для удаления таких объектов. Для удобства федерата служба Resign Federation Execution должна принимать аргумент, требующий от инфраструктуры RTI выполнения нуля или более из следующих действий:

- Освободить все атрибуты объекта для приобретения прав владения позднее. Это означает, что у атрибутов объекта не будет владельца (и их значения не будут обновляться), что сделает возможным приобретение прав владения;
- Удалить все объекты, для которых федерат имеет право на удаление (неявный вызов службы Delete Object Instance).

Работа с точками синхронизации

Register Federation Synchronization Point

Служба Register Federation Synchronization Point используется, чтобы инициировать регистрацию метки для будущей точки синхронизации. После успешной регистрации метки точки синхронизации (о чем сообщает служба Confirm Synchronization Point), инфраструктура RTI должна известить все или некоторые федераты о существовании метки вызовом службы Announce Synchronization Point в этих федератах. Множество описателей федератов, являющееся необязательным параметром, используется федератом

для указания, какие федераты в исполняемой федерации должны быть проинформированы о существовании метки. Если это множество пусто или не предоставлено, то о метке должны быть проинформированы все федераты. В противном случае, все указанные федераты должны быть членами процесса исполнения федерации. Тег пользователя позволяет связать с точкой синхронизации дополнительную информацию и должен быть сообщен вместе с меткой синхронизации. Возможно одновременное существование нескольких точек синхронизации, зарегистрированных одним или несколькими федератами. Однако, синхронизационные метки должны быть уникальными.

Confirm Synchronization Point Registration

Служба Confirm Synchronization Point Registration сообщает федерату результат запрошенной регистрации точки синхронизации. Он должен быть вызван в ответ на вызов службы Register Federation Synchronization Point. Индикатор успеха сообщает федерату либо об успешной регистрации синхронизации синхронизационной метки либо о неудаче из-за уже использованной метки или по иной причине. Попытка регистрации, закончившаяся неудачно, не должна никак влиять на процесс исполнения федерации.

Announce Synchronization Point

Служба Announce Synchronization Point информирует федерат о существовании новой точки синхронизации. После того, как метка точки синхронизации будет зарегистрирована службой Register Federation Synchronization Point, инфраструктура RTI вызывает службу Announce Synchronization Point, либо в каждом из федератов процесса исполнения федерации, либо только в федератах, указанных при регистрации метки. Федераты, проинформированные о существовании точки синхронизации с помощью вызова службы Announce Synchronization Point, образуют множество синхронизации для этой точки. Если при регистрации синхронизационной метки множество описателей федератов не было указано или было пусто, то инфраструктура RTI должна вызвать службу Announce Synchronization Point во всех федератах, присоединяющихся во время существования точки синхронизации, т.е. в промежутке от ее регистрации до вызова службы Synchronization Point Achieved всеми федератами, проинформированными о существовании этой точки синхронизации. Эти вновь присоединённые федераты становятся частью множества синхронизации этой точки. Федераты, вышедшие из процесса исполнения федерации после объявления точки синхронизации, но до синхронизации федерации в этой точке, удаляются из множества синхронизации. Тег

пользователя в вызове службы *Announce Synchronization Point* должен совпадать с тем, переданным соответствующему вызову службы *Register Federation Synchronization Point*.

Synchronization Point Achieved

Служба *Synchronization Point Achieved* информирует инфраструктуру RTI о достижении федератом указанной точки синхронизации. После того, как все федератами в множестве синхронизации для какой-либо точки вызвали эту службу, RTI не имеет права вызывать *Announce Synchronization Point* у вновь присоединяющихся федератов.

Federation Synchronized

Служба *Federation Synchronized* информирует федерата о том, что все федераты из множества синхронизации для заданной точки синхронизации вызвали службу *Synchronization Point Achieved* для этой точки. Эта служба будет вызвана во всех федератах из множества синхронизации для этой точки, сообщая, что федераты из этого множества синхронизованы в этой точке. После синхронизации в точке (служба *Federation Synchronized* вызвана во всех федератах во множестве), эта точка перестаёт быть зарегистрированной, а множество синхронизации для этой точки больше не существует.

Сохранения состояния

Request Federation Save

Служба *Request Federation Save* указывает на необходимость сохранения федерации. При отсутствии необязательного аргумента «федеративное время», после вызова службы *Request Federation Save* инфраструктура RTI должна потребовать от всех участников процесса исполнения федерации скорейшего сохранения состояния. Если аргумент «федеративное время» присутствует, то каждому управляемому временем участнику федерации RTI указывает сохранить состояние в момент достижения логическим временем этого федерата значения, переданного службе; всем федератам, не управляемым временем, должно быть указано сохранить состояние после достижения последним из управляемых временем федератов значения логического времени, переданного службе. RTI указывает федерату о сохранении состояния с помощью вызова службы *Initiate Federate Save* в этом федерате. В любой момент времени может быть только один ожидающий запрос на сохранение. Новый запрос замещает любой ранее ожидавший запрос. Однако, запрос на сохранение не может возникнуть во время сохранения, то есть в промежутке между вызовом инфраструктурой RTI службы *Initiate Federation Save* и вызовом службы *Federation Saved*.

Initiate Federate Save

Служба `Initiate Federate Save` указывает федерату сохранить состояние. После вызова службы `Initiate Federate Save` федерат должен сохраниться так быстро, как только возможно. Метка, переданная инфраструктуре RTI в момент запроса сохранения службой `Request Federation Save`, передается федерату. Федерат должен использовать эту метку, имя исполняемой федерации, свой описатель федерата, а также тип федерата, который он передал при вызове службы `Join Federation Execution`, чтобы идентифицировать информацию о сохранённом состоянии. Если федерат не управляется временем, то он должен быть готов к получению службы `Initiate Federate Save` в любой момент времени. Если федерат управляется временем, то вызов службы `Initiate Federate Save` может произойти только при ожидании завершения одного из следующих служб: `Time Advance Request`, `Time Advance Request Available`, `Next Event Request`, `Next Event Request Available`, `Flush Queue Request`. После получения вызова службы `Initiate Federate Save` федерат должен немедленно прекратить передачу новой информации в федерацию. Федерат может возобновить передачу новой информации в федерацию только после получения вызова службы `Initiate Federate Save`.

Federate Save Begun

Служба `Federate Save Begun` информирует инфраструктуру RTI о начале сохранения федератом своего состояния.

Federate Save Complete

Служба `Federate Save Complete` сообщает инфраструктуре RTI о завершении федератом попытки сохранения. Индикатор успешности сохранения сообщает RTI об успехе или неуспехе сохранения.

Federation Saved

Служба `Federation Saved` сообщает федерату о завершении процесса сохранения федерации и указывает, успешно ли оно завершилось. Успешное завершение означает, что все федераты, в которых была вызвана служба `Initiate Federation Save`, успешно вызвали службу `Federate Save Complete`. Неуспешное завершение означает, что один или более федератов, в которых была вызвана служба `Initiate Federation Save`, неуспешно вызвали службу `Federate Save` с указанием неудачи, или что инфраструктура RTI обнаружила ошибку в одном или нескольких из таких федератов. Все федераты, получившие вызов службы `Initiate Federation Save` должны получить вызов службы `Federation Saved`. Если федерат, получивший вызов службы `Initiate Federation Save`, выходит из процесса исполнения федерации до вызова службы `Federation Saved`, то это должно считаться

ошибкой сохранения федерации и служба Federation Saved должна быть вызвана с указанием неудачи.

Восстановление состояния

Request Federation Restore

Служба Request Federation Restore указывает инфраструктуре RTI начать восстановление процесса исполнения федерации. Восстановление федерации должно быть начато сразу же после проверки допустимости вызова службы Request Federation Restore. Допустимость запроса на восстановление федерации сообщается вызовом службы Confirm Federation Restoration Request.

Confirm Federation Restoration Request

Служба Confirm Federation Restoration Request указывает федерату статус запрошенного восстановления федерации. Эта служба должна быть вызвана в ответ на вызов службы Request Federation Restore. Положительное значение индикатора успеха сообщает федерату о том, что инфраструктура RTI обнаружила сохраненную информацию состояния, соответствующую указанной метке и имени процесса исполнения федерации, совпадении количество и типы присоединенных федератов совпали с количеством и типами, имевшимися на момент сохранения, и ни один иной федерат не пытается в данный момент восстановить федерацию. Если несколько федератов одновременно пытаются восстановить состояние, то один из них должен получить положительное значение индикатора успеха, а все остальные – отрицательное значение. Попытка восстановления, закончившаяся неудачно, не должна иметь никакого влияния на процесс исполнения федерации.

Federation Restore Begun

Служба Federation Restore Begun информирует федерат о начале восстановления. После получения вызова службы Federation Restore Begun федерат должен немедленно прекратить предоставление новой информации в федерацию. Федерат может продолжить предоставление новой информации в федерацию только после получения вызова службы Federation Restored.

Initiate Federate Restore

Служба Confirm Federate Restoration Request указывает федерату о возвращении к ранее сохраненному состоянию. Федерат должен выбрать подходящую информацию для восстановления на основании имени текущего процесса исполнения федерации, переданной метки сохранения федерации и переданного описателя федерата. В результате

вызова этой службы, описатель федерата может принять значение, отличное от возвращенного службой Join Federation Execution.

Federate Restore Complete

Служба Federate Save Complete уведомляет инфраструктуру RTI о том, что федерат завершил попытку восстановления. Если восстановление произошло успешно, то федерат будет находиться в состоянии, которое либо он, либо какой-то другой федерат его типа, имел в момент сохранения с указанной меткой, с тем отличием, что федерат будет ожидать вызова службы Federation Restored.

Federation Restored

Служба Federation Restored информирует федерата о завершении процесса восстановления федерации и указывает успешность или не успешность завершения. Успешность означает, что все федераты, у которых была вызвана служба Federation Restore Begun, вызвали службу Federate Restore Complete с признаком успеха. Неуспешность означает, что один или несколько федератов у которых была вызвана служба Federation Restore Begun вызвали службу Federate Restore Complete с указанием неуспеха, или что инфраструктура RTI обнаружила ошибку в одном или нескольких из таких федератов. Все федераты, получившие вызов службы Federation Restore Begun, должны получить вызов службы Federation Restored. Если федерат, получивший вызов службы Federation Restore Begun, покидает процесс исполнения федерации до вызова службы Federation Restored, это должно считаться ошибкой восстановления федерации, и служба Federation Restored должна быть вызвана с указанием неудачи.

На рис.4 представлен жизненный цикл федерата.

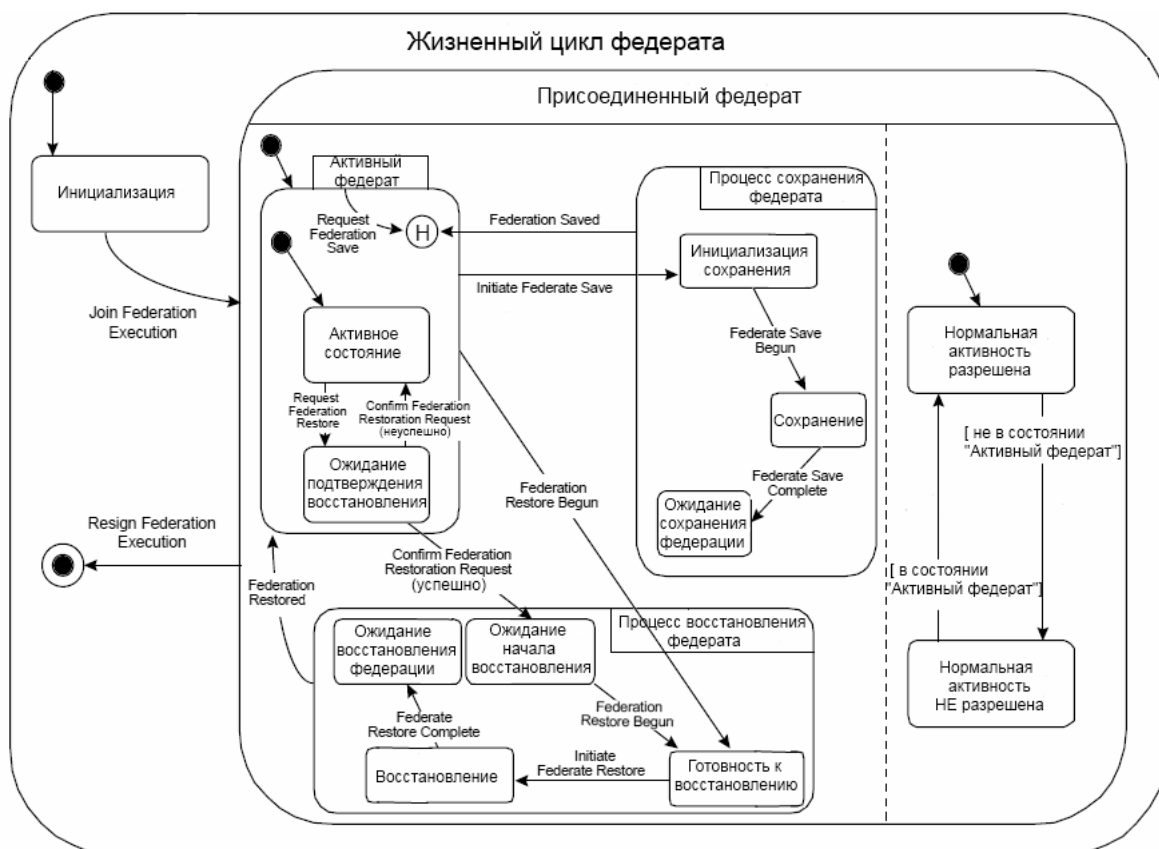


Рисунок №4 Жизненный цикл федерата.

2.2.3 Сводные результаты обзора

Результаты изучения документации (МАК) и исходных кодов (CERTI, PORTICO) реализаций RTI, на предмет соответствия стандарту FOM HLA, представлены в таблице.

	CERTI	PORTICO	МАК
Create Federation Execution	V		V
Destroy Federation Execution	V		V
Join Federation Execution	V		V
Resign Federation Execution	V		V
Register Federation Synchronization Point	V	V	V
Confirm Synchronization Point Registration	V	V	V
Announce Synchronization Point	V	V	V
Synchronization Point Achieved	V	V	V
Federation Synchronized	V	V	V
Request Federation Save	V		V

Initiate Federate Save	V	V	V
Federate Save Begun	V		V
Federate Save Complete	V	V	V
Federation Saved	V	V	V
Request Federation Restore	V		V
Confirm Federation Restoration Request			V
Federation Restore Begun	V	V	V
Initiate Federate Restore	V	V	V
Federate Restore Complete	V	V	V
Federation Restored	V	V	V

Таблица №2 Реализация интерфейсов FOM в различных RTI.

2.2.4 Выводы

Как видно из результатов, приведённых в таблице 2, среди реализаций HLA FOM с открытым исходным кодом наиболее полной является CERTI. Более того, по степени соответствия стандарту в части средств описания интерфейса моделей CERTI не уступает лучшим из коммерческих систем (МАК RTI).

2.3 Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ

Существующие средства распределённого имитационного моделирования в реальном времени основаны на одной из двух технологий. Первая – это HLA Run-Time Infrastructure (HLA RTI, см. раздел 2.2), составная часть стандарта IEEE по организации распределённого гетерогенного моделирования, Вторая – Data Distribution Service (DDS), стандарт передачи данных в реальном времени, разработанный консорциумом OMG для организации передачи данных в РВС РВ.

Цель данного раздела исследования – провести анализ существующих реализаций HLA RTI и DDS, а также описать технологию, используемую для организации полунатурного моделирования в реальном времени, разработанную в ЛВК ВМК МГУ имени М.В.Ломоносова. На основании такого анализа будет выявлено ПО, потенциально пригодное для включения в состав прототипа разрабатываемых средств. Дальнейший анализ и сравнение образцов ПО предполагается провести в рамках научно-исследовательской работы.

2.3.1 Стенд ПНМ, разработанный в ЛВК МГУ

Стенд ПНМ поддерживает выполнение имитационной модели одновременно на нескольких инструментальных машинах, объединённых сетью. При этом, для обмена информацией используется несколько физических каналов связи:

1. Ethernet со стеком протоколов tcp/ip,
2. Ethernet для поддержки моделирования в реальном времени,
3. LPT для точной синхронизации.

К каждой инструментальной машине может быть подключён набор тестируемых устройств и привязан набор моделей таких устройств. Модели и устройства обмениваются между собой сообщениями.

Детальное описание стенда ПНМ приводится в дополнительных материалах (Приложение Г).

2.3.2 Сравнение существующих реализаций HLA RTI

Исследование проводилось среди реализаций RTI, удовлетворяющих (хотя бы частично) следующим критериям:

1. Доступен исходный код продукта.
2. Существует описание архитектуры и принципов работы реализации;
3. Продукт продолжает поддерживаться и развиваться;
4. Реализация используется для проведения моделирования реального времени;
5. Реализация основана на стандарте High-Level Architecture (HLA);

Таблица 1. Реализации HLA RTI.

Средство	Разработчик	Доступность кода	Поддержка HLA
NCWare & SimWare	Nextel[21]	Закрытый	1516-2000
EODiSP	P&P Software[22]	Открытый	1516-2000
CERTI	ONERA[23]	Открытый	1516-2000
MAK RTI	MAK Technologies[24]	Закрытый	1516-2010 (Evolved)
pRTI	Pitch Technologies[25]	Закрытый	1516-2010 (Evolved)
GAIA	University of Bologna [26]	Открытый	1516-2000
Portico	Portico[27]	Открытый	1.3

Большинство найденных реализаций RTI имеют закрытый исходный код, и их польза при построении новой системы распределённого моделирования ограничивается теоретической базой. Проект с открытым исходным кодом EODiSP прекратил своё развитие в 2006 году и имеет ограниченную функциональность. Реализация Portico совместима лишь со стандартом HLA 1.3 и использует Java, что плохо совместимо с системами реального времени. Более подходящей является реализация ARTIS системы GAIA, но лицензия этого продукта не позволяет его свободное использование (хотя заявлено, что в будущем проект станет полностью открытым).

Таким образом, лучшей базовой RTI для разработки системы моделирования априорно становится реализация CERTI. Поэтому анализ существующих реализаций RTI описывается главным образом в виде сравнения идей, применённых в CERTI и их аналогов из других реализаций.

В результате исследования научных статей, посвящённых системам имитационного моделирования на основе стандарта HLA, за последнее десятилетие был выделен набор признаков для сравнения выделенных реализаций таких систем между собой.

Результаты тестов показывают, что реализация RTI CERTI отстаёт от коммерческих решений [14]. Однако проект активно развивается, и существует множество прецедентов использования CERTI в качестве базы для новых разработок. Подобная задача ставится перед RTI и в рамках настоящего проекта.

2.3.3 Сравнение существующих реализаций стандарта DDS

Спецификации OMG DDS определяют стандарт взаимодействия процессов, применимый к широкому классу распределённых систем реального времени и встроенных систем (DRE). В основе DDS лежит ориентированная на данные модель с архитектурой вида издатель-подписчик (DCPS). Процессы, использующие DDS, могут производить чтение и запись типизированных данных. При этом модель DCPS образует прослойку, которая позволяет читающим данные процессам получить доступ к самой последней их версии. В рамках DCPS определяются глобальное пространство данных, которые можно читать и изменять, и глобальное пространство имён, позволяющее создавать новые и искать существующие разделяемые объекты [18].

Процесс-издатель (publisher), желающий создать разделяемый объект, делает соответствующие записи в глобальных пространствах имён и данных. Аналогично, процесс-подписчик (subscriber) может найти интересующие его объекты в пространстве имён и получить доступ к соответствующим данным. При этом разделяются объявление о необходимости использовать разделяемые данные и непосредственное их использование, что позволяет применять в рамках DDS соединения с контролем качества.

Рассмотренные реализации DDS

Для исследования было выбрано несколько реализаций DDS:

Средство	Разработчик	Доступность кода
NDDS	Real-Time Innovations[28]	Закрытый
OpenSPlice	PrismTechnologies[29]	Открытый
Open DDS	Object Computing, Inc.[30]	Открытый

Таблица 3. Реализации стандарта DDS.

Детальный анализ реализаций DDS приводится в Приложении Г.

2.3.4 Выводы

Наиболее полезными проектами для создания новой распределённой системы реального времени в силу своей открытости являются OpenSPlice и Open DDS. При этом по результатам тестов реализация OpenSPlice выглядит предпочтительнее. В дальнейшем данное средство может быть модифицировано и переведено, например, на децентрализованную архитектурную модель NDDS, показавшую абсолютный результат в ходе тестирования.

2.4 Средства записи и хранения информации о поведении PBC PB

Результатом имитационного эксперимента является трасса имитационной модели. Под трассой понимают журнал событий какого-либо типа, произошедших во время работы системы. Соответственно под трассировкой понимают процесс сбора информации о системе во время ее функционирования. Для того чтобы получить трассу в исследуемую систему встраиваются "профилировочные" вызовы, которые фиксируют наступление определенных событий, их продолжительность и фиксируют эту информацию в трассе.

По полученным в ходе экспериментов трассам с помощью специальных программных инструментов осуществляется анализ и визуализация производительности и поведения исследуемой системы (с целью улучшения её производительности, либо других интересующих характеристик системы).

Можно выделить следующие общие проблемы, связанные с трассировкой программно-аппаратных систем и относящиеся к недостаткам существующих средств анализа:

- Формат трасс не унифицирован и обычно ориентирован на конкретную библиотеку передачи сообщений.
- Слабые возможности настройки фильтров событий (какие события и какую информацию включать в трассы). Нет возможности варьировать объем трассы, в связи с чем возникает проблема сжатия и хранения больших трасс.
- Не учитывается эффект замера - средство трассировки достаточно сильно изменяет поведение программы.

В данном разделе приводятся результаты исследования существующих форматов хранения трасс и ПО, обеспечивающего их использование. Исследование проведено с учётом цели записи трасс: фиксация поведения PBC PB для последующего анализа и визуализации.

2.4.1 Существующие форматы хранения трасс событий

Были рассмотрены основные существующие форматы трасс, применяемые в настоящее время для трассировки поведения различных вычислительных систем или приложений.

Список форматов трасс, средства анализа и возможности конвертации форматов приведены в таблице 4.

№	Формат	Средство анализа трассы	Конвертеры в другие форматы трасс
1	OTF [36-37]	Vampir, VNG, VITE	VTF
2	из Дианы [33]	Vis	нет
3	из V-Ray [34]	V-Ray средство	нет
4	Paje [35]	Paje, VITE	нет
5	EPILOG [38]	VNG	VTF
6	SLOG-2 [39, 41]	JumpShot-4	нет
7	CLOG [39, 41]	JumpShot-4	
8	CCG [31, 40]	нет	нет
9	STF [31]	VNG, ITA 6.0, Vampir	VTF
10	TAU Trace[32]	VNG, VITE	Paraver, SLOG-2, ALOG, EPILOG, OTF, VTF
11	VTF [43]	VNG, ITA 4.0	OTF
12	ALOG [39, 41]	JumpShot-4	

Таблица 4 Форматы трасс, средства анализа трасс, возможность конвертирования трасс.

Как можно заметить по таблице 4, в основном каждое средство анализа и визуализации ориентировано на работу с определенным форматом трасс.

2.4.2 Выводы

Можно сформулировать следующие требования к формату трасс для PBC PB:

1. **Документированность формата** - наличие описания трассы, событий, определений и принципа их записи в трассу.
2. **Наличие реализации формата** означает, что данный формат, помимо теоретического описания, имеет реализацию на одном из языков программирования и ранее применялся для какого-либо проекта.

3. **Открытость API** означает открытость библиотек чтения и записи для данного формата.
4. **Открытость средств работы с трассами** связана со специальными средствами анализа и визуализации трасс данного формата, которые должны относиться к программному обеспечению с открытым исходным кодом (opensource).
5. **Компактность трассы** связана с объемом памяти, необходимой для хранения трассы данного формата. Поскольку объем трассы непосредственно связан с кодировкой её данных, трассы можно разделить на 3 класса:
 - a. Текстовые.
 - b. Бинарные.
 - c. Со специальной кодировкой.
6. **Наличие специализированных запросов к трассе**, которые отсутствуют в других форматах.

С учетом сформулированных требований можно сделать следующие выводы по каждому из форматов трасс.

Формат	Документированность (описание)	Реализация формата	Открытость API	Открытость средств работы	Компактность сжатия (Кодировка, класс)	Наличие спец. запросов
OTF	да	да	да	да	ASCII (c)	есть
«Диана»	да	да	ЛВК МГУ	ЛВК МГУ	бинарный (b)	есть
«VD-Ray»	да	да	да	да	текстовый (a)	есть
Raje	да	да	да	да	текстовый (a)	нет
EPILOG	да	да	да	да	бинарный (b)	нет
SLOG-2	да	да	да	да	бинарный (b)	нет
CLOG	да	да	да	да	бинарный (b)	нет
CCG	да	нет	да	да	не реализован	есть
STF	нет	да	Intel	Intel	бинарный (b)	есть
TAU Trase	да	да	да	да	бинарный (b)	нет
ALOG	да	да	да	да	бинарный (b)	нет

Таблица 5. Сравнение форматов трасс.

На основе приведенной таблицы можно сделать следующее заключение: наиболее приемлемыми являются форматы OTF (Open Trase Format) и формат проекта «Диана». Однако в дальнейших практических исследованиях с форматами трасс предлагается рассмотреть не только перечисленные форматы, но и форматы, конвертируемые друг в друга - TAU, CLOG, SLOG-2 и EPILOG.

2.5 Средства анализа данных о поведении PBC PB

Результаты проведенного обзора средств визуализации и анализа для рассмотренных ранее форматов трасс приведены в таблице:

Средство	Поддерживаемые форматы	Открытость ПО (opensource)
Vampir NG (VNG)	EPILOG, VTF3, OTF	да
JumpShot-4	*LOG	да
ViTE	TAU, OTF, Paje	да
ITA 7.0	STF	Intel
Vis	формат "Дианы"	ЛВК МГУ
VD-Ray	V-Ray	НИВЦ МГУ
КОЖАК	EPILOG	да
Paje	Paje	нет

Таблица 6. Сравнение средств анализа трасс.

Исходя из критерия открытости средств визуализации и анализа трасс и на основе выводов из обзора форматов трасс, можно сделать следующее заключение: наиболее приемлемыми для дальнейшего практического исследования являются средства

- ViTE
- Vis
- КОЖАК
- Vampir NG
- JumpShot-4

2.6 Анализ охраноспособности

Для анализа охраноспособности разрабатываемого ПО был проведён анализ патентной документации, а именно рефератов из официального бюллетеня «Программы для ЭВМ. Базы данных. Топологии интегральных микросхем» Федеральной службы по интеллектуальной собственности, патентам и товарным знакам (ФИПС).

Не было найдено ни одного полного аналога, однако был найден ряд средств, являющихся аналогами отдельных компонентов разрабатываемого ПО. Основная масса средств разработана в Федеральном государственном унитарном предприятии "Государственный научно-исследовательский институт авиационных систем" (ФГУП "ГосНИИАС"). Это ПО, предназначенное для решения отдельных задач, возникающих при моделировании РВС РВ:

- средства имитации полетной обстановки при полунатурном моделировании (SAMVME, регистрационный номер 2010613286),

- средства задания начальных условий (НУ) полетной обстановки, режимов работы комплекса полунатурного моделирования (Startset, регистрационный номер 2010613287),
- средства регистрации и отображения на экране монитора в реальном масштабе времени и после сеанса моделирования в удобной для оператора графической форме информации, передаваемой по каналам (RTM, регистрационный номер 2010613288),
- средства централизованного управления процессом полунатурного моделирования и отображения состояния программ вычислительного центра (DISPATCH, регистрационный номер 2010613289),
- средства визуализации моделируемой тактической обстановки в реальном масштабе времени при полунатурном моделировании (ИТО, регистрационный номер 2010613290),
- средства обработки бортовых систем управления подвесными авиационными средствами в реальном масштабе времени (SSS, регистрационный номер 2010613291).

Также обнаружена информация по программно-математическому обеспечению лётно-моделирующего комплекса для проведения исследований и сопровождения лётных испытаний пилотажно-навигационного оборудования летательных аппаратов (Федеральное государственное унитарное предприятие "Лётно-исследовательский институт имени М.М. Громова", регистрационный номер 2009610697) и экспертной системе постанализа вычислительных процессов в приборах навигационного комплекса (Открытое акционерное общество "Концерн "Центральный научно-исследовательский институт "Электроприбор", регистрационный номер 2010613553).

Ни одно из упомянутых средств, ни все они в комплексе не удовлетворяют всем требованиям к разрабатываемому ЭО ТПК. Более того, ни одно из них не является ПО с открытыми исходными кодами. Детальная информация по найденным патентам представлена в приложении Г.

2.6.1 Выводы

Проведенный анализ и сопоставление программ для ЭВМ, являющихся наиболее близкими аналогами разрабатываемого объекта, позволяет заключить, что основными преимуществами создаваемого ЭО ТПК являются:

- узкая специализация на предметной области – анализ поведения распределённых вычислительных систем реального времени, с учётом особенностей, связанных с обеспечением функционирования, мониторинга и анализа поведения таких систем;
- открытость исходного кода, что позволит добиться широкого распространения программного продукта и обеспечит возможность его модернизации (создания новых версий программ) под конкретные задачи пользователей;
- возможность работы под операционной системой (Linux), распространяемой на основе «открытой» лицензии, что позволит пользователям экономить существенные денежные средства на приобретение программного обеспечения.

Наличие данных отличительных особенностей характеризует новизну разрабатываемого объекта, что свидетельствует о его охраноспособности.

ЗАКЛЮЧЕНИЕ

В рамках настоящего патентного исследования произведен тематический поиск российской и зарубежной охранной документации, а также мониторинг прочих информационных ресурсов, в результате которого выявлены наиболее близкие к разрабатываемому объекту аналоги (программы для ЭВМ).

С учётом того, что в рамках НИР «Создание прототипа интегрированной среды и методов комплексного анализа функционирования распределённых вычислительных систем реального времени (РВС РВ)» планируется разработка ЭО ТПК с открытым исходным кодом, все исследованные образцы ПО можно условно разделить на две группы. Первая группа – это аналоги разрабатываемого ЭО ТПК, присутствующие на рынке. Анализ такого ПО позволяет оценить актуальные требования к средствам разработки РВС РВ и возможности, востребованные рынком, а также новизну разрабатываемого ЭО ТПК.

Вторая группа – ПО с открытым исходным кодом, лицензия которого позволяет использовать его в качестве составной части ЭО ТПК. Исследованы образцы подобного ПО, пригодные для использования в составе ЭО ТПК.

Анализ и сопоставление программ для ЭВМ, являющихся наиболее близкими аналогами разрабатываемого объекта, позволил заключить, что основными преимуществами создаваемого ЭО ТПК являются:

- узкая специализация на предметной области – анализ поведения распределённых вычислительных систем реального времени, с учётом особенностей, связанных с обеспечением функционирования, мониторинга и анализа поведения таких систем;
- открытость исходного кода, что позволит добиться широкого распространения программного продукта и обеспечит возможность его модернизации (создания новых версий программ) под конкретные задачи пользователей;
- возможность работы под операционной системой (Linux), распространяемой на основе «открытой» лицензии, что позволит пользователям экономить существенные денежные средства на приобретение программного обеспечения.

Наличие данных отличительных особенностей характеризует новизну разрабатываемого объекта, что свидетельствует о охраноспособности объекта разработки.

Аналогов ЭО ТПК, распространяемых по лицензии с открытым исходным кодом, выявлено не было, однако был выявлен ряд средств, которые могут быть использованы как составные части ЭО ТПК.

В части средств описания интерфейсов подключаемых моделей рекомендуется опираться на раздел FOM стандарта HLA [13], как единственного существующего стандарта на интерфейсы подключения гетерогенных моделей для распределённого имитационного моделирования. В ходе исследования было установлено, что ПО CERTI [23] включает в себя реализацию HLA FOM, пригодную для использования в ЭО ТПК.

В части средств организации распределённого моделирования в реальном времени был выявлен ряд средств, удовлетворяющих поставленным требованиям и пригодных для использования в составе ЭО ТПК. Нужно отметить, что при использовании любого из выявленных средств потребуется его доработка под задачи проекта. Это средства Open DDS [30] и OpenSplice [29], реализующие стандарт OMG DDS [17] и распространяемые по свободной лицензии с открытым исходным кодом. Данные средства разрабатывались без учёта особенностей применения в области имитационного моделирования, поэтому их применение в ЭО ТПК возможно лишь в сопряжении со специализированными средствами РИМ, которые можно заимствовать из проекта «Стенд ПНМ» [1]. Также может быть использована среда распределённого моделирования CERTI [25], её требуется доработать для поддержки моделирования в реальном времени.

В части форматов хранения и средств анализа трасс были выделены наиболее приемлемые форматы трасс (OTF, формат трасс проекта «Диана», TAU, CLOG, SLOG-2 и EPILOG) для решения задач анализа поведения PCB PB и соответствующие средства для работы с ними (ViTE, Vis, KOJAK, Vampir NG, JumpShot-4) с открытыми исходными кодами. Данные форматы и средства в основном применяются для анализа поведения параллельных приложений, поэтому потребуется их доработка с учетом особенностей PCB PB и решаемых задач анализа.

В дальнейшем необходимо детально проанализировать перечисленные образцы ПО и принять решение о целесообразности использования их в составе ЭО ТПК и необходимых доработках.

Таким образом, настоящее патентное исследование проведено в полном соответствии с Заданием №1. Задачи, поставленные перед патентными исследователями, выполнены.

СПИСОК ЛИТЕРАТУРЫ

- [1] Балашов В.В., Бахмуров А.Г., Волканов Д.Ю., Смелянский Р.Л., Чистилинов М.В., Ющенко Н.В., “Стенд полунатурного моделирования для разработки встроенных вычислительных систем”, 2009.
- [2] RT-LAB Professional // Официальный сайт компании Opal-RT Technologies. URL: <http://opal-rt.com/product/rt-lab-professional> (дата обращения: 24.11.10)
- [3] Brouwer, M.P.A.M., Castelijns, A.A., van Ingen Schenau, H.A., Oving, B.A., Timmermans, L.J. and Zwartbol, T. Developments in Test and Verification Equipment for Spacecraft. Technical report NLR-TP-2000-658. National Aerospace Laboratory, Netherlands, 2000.
- [4] ECU Testing with dSPACE Simulator // Официальный сайт компании dSPACE GmbH. URL: <http://www.dspace.com/en/pub/home/products/systems/ecutest.cfm> (дата обращения: 23.11.10).
- [5] The ADvantage Framework // Официальный сайт компании Applied Dynamics International. URL: http://www.adi.com/products_sim.htm (дата обращения: 23.11.10).
- [6] Hardware-in-the-Loop Test System // Официальный сайт компании National Instruments. URL: <http://www.ni.com/embedded/hil.htm> (дата обращения: 23.11.10).
- [7] Manfred Broy. Architecture Based Specification and Verification of Embedded Software Systems // ISoLA 2008, CCIS 17, pp. 1–13, 2008.
- [8] GaborMadl, Nikil Dutt, Sherif Abdelwahed. Performance Estimation of Distributed Real-time Embedded Systems by Discrete Event Simulations // EMSOFT’07, September 30 – October 3, 2007.
- [9] Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules // IEEE, 2010.
- [10] CERTI Summary // Документация по проекту, 2010 URL: <http://savannah.nongnu.org/cookbook/?group=certi> (дата обращения: 20.10.2010).
- [11] Developer Documentation – Portico // Официальная документация по проекту, 2010 URL: http://porticoproject.org/index.php?title=Main_Page (дата обращения 20.10.2010).
- [12] MAK RTI – High Performance RTI // официальный сайт проекта, 2010 URL: <http://www.mak.com/products/rti.php#dlform> (дата обращения: 21.10.2010).
- [13] Modeling and Simulation (M&S) High Level Architecture (HLA) — Object Model Template (OMT) Specification // IEEE, 2000, с 260.
- [14] L. Malinga, Willem H. le Roux. HLA RTI Performance Evaluation // Council for Scientific and Industrial Research, 2009.
- [15] P. Knight, A. Corder, R. Liedell. Evaluation of Run Time Infrastructure (RTI) Implementations // US Army SMDC Wynn Drive, Huntsville, 2002.

- [16] Portico Performance Reports – Protico // сайт, 2010 URL: http://www.porticoproject.org/index.php?title=Portico_Performance_Reports (дата обращения: 22.10.2010).
- [17] Object Management Group; Object Interface Systems, Inc; Real-Time Innovations, Inc; THALES, 2007. Data Distribution Service for Real-time Systems, version 1.2.
- [18] Real-Time Innovations, Inc. (RTI), 2004. OMG Data-Distribution Service (DDS): Architectural Overview.
- [19] Schmidt, D. POLLUX: ENHANCING THE QUALITY OF SERVICE OF THE GLOBAL INFORMATION GRID (GIG). The Vanderbilt University, Nashville TN, USA, 2009.
- [20] M. Xiong, J. Parsons, J. Edmondson, H. Nguyen, D. Schmidt. Evaluating the Performance of Publish/Subscribe Platforms for Information Management in Distributed Real-time and Embedded Systems. Vanderbilt University, Nashville TN, USA, 2006.
- [21] NCWare & SimWare // Официальный сайт компании Nextel. URL: <http://www.nexteleng.es/> (дата обращения 30.10.2010).
- [22] EODiSP // Официальный сайт компании P&P Software. URL: <http://www.pnp-software.com/> (дата обращения 30.10.2010).
- [23] CERTI // Официальный сайт лаборатории ONERA. URL: <http://www.cert.fr/> (дата обращения 30.10.2010).
- [24] MAK RTI // Официальный сайт компании MAK Technologies. URL: <http://www.mak.com/> (дата обращения 30.10.2010).
- [25] pRTI // Официальный сайт компании Pitch Technologies. URL: <http://www.pitch.se/> (дата обращения 30.10.2010).
- [26] GAIA // Официальный сайт разработчика (University of Bologna). URL: <http://pads.cs.unibo.it/> (дата обращения 30.10.2010).
- [27] Portico // Официальный сайт проекта Portico. URL: <http://porticoproject.org/> (дата обращения 30.10.2010).
- [28] NDDS // Официальный сайт компании Real-Time Innovations. URL: <http://www.rti.com/> (дата обращения 30.10.2010).
- [29] OpenSPlice // Официальный сайт компании PrismTechnologies. URL: <http://www.opensplice.com/> (дата обращения 30.10.2010).
- [30] Open DDS // Официальный сайт проекта Portico. URL: <http://www.ociweb.com/> (дата обращения 30.10.2010).
- [31] Andreas Knupfer. Advanced Memory Data Structures for Scalable Event Trace Analysis. Dissertation, Dresden University of Technology, March 2009.

- [32] Sameer Shende. TAU Source Code, Version 2.13.5. Personal communications, 2004.
- [33] Среда моделирования ДИАНА: Визуализатор временной диаграммы. Руководство оператора. Редлаб, 2009. 37 с.
- [34] Проект V-Ray // Официальный сайт НИВЦ МГУ, URL: <http://parallel.ru/v-ray/>
- [35] O. Stein, J. Chassin de Kergommeaux. Paje trace file format. March 2003.
- [36] A. Knupfer, H. Brunst, A. D. Malony, S. S. Shende. Open Trace Format API Specification Version 1.1. Dresden University of Technology, Germany, 2006.
- [37] A. Knupfer, H. Brunst, A. D. Malony, S. S. Shende. Open Trace Format (OTF) Tutorial. Presentation. University of Dresden, Germany, 2006.
- [38] F. Wolf, B. Mohr, N. Bhatia, M.-A. Hermanns, M. Geimer. Epilog Binary Trace-Data Format. University of Tennessee, 2006.
- [39] A. Chan, W. Gropp, E. Lusk. Scalable Log Files for Parallel Program Trace Data. Mathematics and Computer Science Division. ANL/MCS-TM-249.
- [40] A. Knupfer, W. E. Nagel. New Algorithms for Performance Trace Analysis Based on Compressed Complete Call Graphs // Computational science – ICCS 2005. Part2, pp. 116-123.
- [41] A. Chan, B. Gropp, R. Lusk. Logfile Format // Mathematics and Computer Science Division, Argonne National Laboratory, URL: http://www.mcs.anl.gov/research/projects/perfvis/software/log_format/index.htm (время обращения: 30.10.2010).
- [42] Performance API // URL: http://icl.cs.utk.edu/projects/papi/wiki/Main_Page (время обращения: 30.10.2010).
- [43] S. Seidl, VTF3 - A Fast Vampir Trace File Low-Level Management Library // Dresden University of Technology, Center for High-Performance Computing, Report ZHR-R-0304, Nov 2003.
- [44] M. Adelantado, P. Siron, J.-B. Chaudron. Towards an HLA Run-time Infrastructure with Hard Real-time Capabilities. Toulouse, France, 2010.
- [45] P. Bieber, D. Raujol, P. Siron. Security Architecture for Federated Cooperative Information Systems. Toulouse, France, 2002.
- [46] I. Birrer, B. Carnicero-Dominguez, M. Egli, A. Pasetti. EODiSP – AN OPEN AND DISTRIBUTED SIMULATION PLATFORM. Noordwijk, the Netherlands, 2006.
- [47] L. Bononi, G. D'Angelo, L. Donatiello. HLA-BASED ADAPTIVE DISTRIBUTED SIMULATION OF WIRELESS MOBILE SYSTEMS, 2003.

- [48] B. d'Ausbourg, P. Siron, E. Noulard. Running Real Time Distributed Simulations under Linux and CERTI. Toulouse, France, 2008.
- [49] R. D. Fujimoto. Parallel and Distributed simulation systems, 2000.
- [50] M. Karlsson, P. Karlsson. An In-Depth Look at RTI Process Models. Pitch AB, Linköeping, Sweden, 2003.
- [51] B. A. Mello, F. R. Wagner. A Standardized Co-simulation Backbone. (б.д.).
- [52] B. Moeller, F. Antelius. Object-Oriented HLA - Does One Size Fit All?, 2010.
- [53] B. Moeller, M. Karlsson. Making RTI Tuning Easy with Performance Profiles. Pitch Technologies, Linköping, Sweden, 2005.
- [54] E. Noulard, J.-Y. Rousselot. CERTI, an Open Source RTI, why and how. Toulouse, France, 2009.
- [55] L. M. Roux. HLA RTI Performance Evaluation. Pretoria, South Africa, 2009.
- [56] K. J. Rycerz. Grid-based HLA Simulation Support. Krakow, Poland, 2006.
- [57] D. D. Wood. Implementation of DDM in the MAK High Performance RTI. Cambridge, 2002.
- [58] Y. Xie, Teo Y. Meng, W. Cai, S. Turner. A Distributed Simulation Backbone for Executing HLA-based Simulation Over the Internet, 2004.

ПРИЛОЖЕНИЕ А – Задание № 1 на проведение патентных исследований

УТВЕРЖДАЮ

Руководитель НИР

_____ Смелянский Р.Л.
«20» сентября 2010 г.

**ЗАДАНИЕ № 1
на проведение патентных исследований**

Наименование работы (темы): Создание прототипа интегрированной среды и методов комплексного анализа функционирования распределённых вычислительных систем реального времени (РВС РВ)

Шифр работы (темы): 2010-1.1-215-138

Этап работы: I, сроки его выполнения: сентябрь 2010 г. - ноябрь 2010 г.

Задачи патентных исследований:

- исследование направлений научно-исследовательской и производственной деятельности организаций и фирм, применяющих анализ функционирования РВС РВ в своих производственных процессах;
- выявление новизны и ближайших существующих аналогов разрабатываемых программных средств;
- исследование охраноспособности объекта.

КАЛЕНДАРНЫЙ ПЛАН

Виды патентных исследований	Подразделения-исполнители (соисполнители)	Ответственные исполнители (Ф.И.О.)	Сроки выполнения патентных исследований. Начало. Окончание	Отчетные документы
1. Тематический поиск документов	ВТК	Савенков Константин Олегович	с 20 сентября 2010 г. по 30 ноября 2010 года	Отчет о патентных исследованиях
2. Исследование новизны и аналогов разрабатываемого объекта	ВТК	Савенков Константин Олегович	с 20 сентября 2010 г. по 30 ноября 2010 года	Отчет о патентных исследованиях

Руководитель НИР

_____	Смелянский Р.Л	_____
личная подпись	расшифровка подписи	дата

ПРИЛОЖЕНИЕ В – Отчет о поиске

V.1 Поиск проведен в соответствии с заданием Руководителя НИР Смелянского Р.Л. №1 от 20.09.2010 г. и Регламентом поиска №1 от 20.09.2010 г.

V.2 Этап работы: -

V.3 Начало поиска: 20 сентября 2010 г. Окончание поиска: 30 ноября 2010 г.

V.4 Сведения о выполнении регламента поиска (указывают степень выполнения регламента поиска, отступления от требований регламента, причины этих отступлений): требования регламента поиска соблюдены.

V.5 Предложения по дальнейшему проведению поиска и патентных исследований: провести патентный поиск на более поздних этапах работы с целью выявления аналогов разрабатываемых программных средств.

V.6 Материалы, отобранные для последующего анализа

Таблица В.6.1 – Охранная документация (свидетельств о государственной регистрации программы для ЭВМ) отобранные для последующего анализа

Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа.	Заявитель (правообладатель), страна. Дата регистрации в государственном Реестре программ для ЭВМ	Название программы	Сведения о действии охранного документа или причина его аннулирования (только для анализа охраноспособности)
1	2	3	4	5
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Россия. Свидетельство о государственной регистрации программы для ЭВМ. №: 2003611671	Общество с ограниченной ответственностью «Редлаб ЛТД». Россия. (29 мая 2003)	Среда моделирования ДИАНА	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Россия. Свидетельство о государственной регистрации программы для ЭВМ. №: 2010612305	Общество с ограниченной ответственностью «Редлаб ЛТД». (26 марта 2010)	Анализатор ДПК	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010612304	Общество с ограниченной ответственностью «Редлаб ЛТД». (28 января 2010)	Среда моделирования «ДИАНА» версия 7.2	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010612303	Общество с ограниченной ответственностью «Редлаб ЛТД». (26 марта 2010)	Анализатор МКЮ	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010612302	Общество с ограниченной ответственностью «Редлаб ЛТД». (26 марта 2010)	Средства функционального тестирования комплексов бортового оборудования	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2009610697	Россия. Федеральное государственное унитарное предприятие "Лётно-исследовательский институт имени М.М.	Программно-математическое обеспечение лётно-моделирующего комплекса для проведения	Действует

Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа.	Заявитель (правообладатель), страна. Дата регистрации в государственном Реестре программ для ЭВМ	Название программы	Сведения о действии охранного документа или причина его аннулирования (только для анализа охраноспособности)
1	2	3	4	5
		Громова" (29 января 2009)	исследований и сопровождения лётных испытаний пилотажно- навигационного оборудования летательных аппаратов	
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010613286	Россия.Федеральное государственное унитарное предприятие "Государственный научно- исследовательский институт авиационных систем" (ФГУП "ГосНИИАС") (19 мая 2010)	Динамическое моделирование полетной обстановки (SAMVME)	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010613287	Россия.Федеральное государственное унитарное предприятие "Государственный научно- исследовательский институт авиационных систем" (ФГУП "ГосНИИАС") (19 мая 2010)	"Задание условий работы комплекса полунатурного моделирования" (Startset)	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010613288	Россия.Федеральное государственное унитарное предприятие "Государственный научно- исследовательский институт авиационных систем" (ФГУП "ГосНИИАС") (19 мая 2010)	"Тестирование информационных каналов последовательного кода" (RTM)	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010613289	Россия.Федеральное государственное унитарное предприятие "Государственный научно- исследовательский институт авиационных систем" (ФГУП "ГосНИИАС") (19 мая 2010)	"Диспетчер работы вычислительного центра" (DISPATCH)	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010613290	Россия.Федеральное государственное унитарное предприятие "Государственный научно- исследовательский институт авиационных систем" (ФГУП "ГосНИИАС") (19 мая 2010)	"Отображение тактической обстановки" (ИТО)	Действует
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010613291	Россия.Федеральное государственное унитарное предприятие "Государственный научно- исследовательский институт авиационных систем" (ФГУП "ГосНИИАС") (19 мая 2010)	"Программа управления имитатором подвесных авиационных средств" (SSS)	Действует

Предмет поиска (объект исследования, его составные части)	Страна выдачи, вид и номер охранного документа.	Заявитель (правообладатель), страна. Дата регистрации в государственном Реестре программ для ЭВМ	Название программы	Сведения о действии охранного документа или причина его аннулирования (только для анализа охраноспособности)
1	2	3	4	5
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Свидетельство о государственной регистрации программы для ЭВМ. №: 2010613553	Россия. Открытое акционерное общество "Концерн "Центральный научно-исследовательский институт "Электроприбор" (28 мая 2010)	Экспертная система постанализа вычислительных процессов в приборах навигационного комплекса	Действует

Таблица В.6.2 – Научно-техническая, конъюнктурная, нормативная документация и материалы государственной регистрации (отчеты о научно-исследовательских работах)

Предмет поиска	Наименование источника информации с указанием страницы источника	Автор, фирма (держатель) технической документации	Год, место и орган издания (утверждения, депонирования источника)
1	2	3	4
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	RT-LAB Professional // Официальный сайт компании Opal-RT Technologies URL: http://opal-rt.com/product/rt-lab-professional (дата обращения: 24.11.10)	OPAL-RT	2010, OPAL-RT
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Real-Time Simulation: Applications to More Electric Aircrafts	Christian Dufour	2010, OPAL-RT
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	The What, Where and Why of Real-Time Simulation	Jean Bélanger, Jean-Nicolas	2010, OPAL-RT
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Developments in Test and Verification Equipment for Spacecraft. Technical report NLR-TP-2000-658.	Brouwer, M.P.A.M., Castelijin, A.A., van Ingen Schenau, H.A., Oving, B.A., Timmermans, L.J. and Zwartbol, T.	2000, National Aerospace Laboratory, Netherlands.
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	ECU Testing with dSPACE Simulator // Официальный сайт компании dSPACE GmbH URL: http://www.dspace.com/en/pub/home/products/systems/ecutest.cfm (дата обращения: 23.11.10)	dSPACE	2010, dSPACE
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Model-Based Software Development According to ISO 26262	Michael Beine	2009, dSPACE
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Applying Model-Based Design and Automatic Production Code Generation to Safety-Critical System Development	Michael Beine, Dr. Ulrich Eisemann, Dirk Fleischer	2009, dSPACE

Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	The ADvantage Framework // Официальный сайт компании Applied Dynamics International URL: http://www.adi.com/products_sim.htm (дата обращения: 23.11.10)	Applied Dynamics International (ADI)	2010, Applied Dynamics International (ADI)
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Predictor Methods in Real-time Simulation	R.M. Howe	2007, Applied Dynamics International (ADI)
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Development Of A Real-Time Simulation System	J. Boyd	2007, Applied Dynamics International (ADI)
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Hardware-in-the-Loop Test System // Официальный сайт компании National Instruments URL: http://www.ni.com/embedded/hil.htm (дата обращения: 23.11.10)	National Instruments	2010, National Instruments
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Accelerating Embedded System Design	B. MacCleery	2007, National Instruments
Программы для ЭВМ, обеспечивающие комплексный анализ функционирования РВС РВ	Стенд полунатурного моделирования для разработки встроенных вычислительных систем	Балашов В.В., Бахмуров А.Г., Волканов Д.Ю., Смелянский Р.Л., Чистилинов М.В., Ющенко Н.В.,	2009, ЛВК
Средства РИМ, допускающие совместное выполнение моделей и натурных компонентов РВС РВ	CERTI Summary // Документация по проекту, 2010 URL: http://savannah.nongnu.org/cookbook/?group=certi (дата обращения: 20.10.2010)	CERTI	2010, CERTI
Средства РИМ, допускающие совместное выполнение моделей и натурных компонентов РВС РВ	CERTI, an Open Source RTI, why and how	Eric Noulard and Jean-Yves Rousselot	2009, Toulouse, France
Средства РИМ, допускающие совместное выполнение моделей и натурных компонентов РВС РВ	Developer Documentation – Portico // Официальная документация по проекту, 2010 URL: http://porticoproject.org/index.php?title=Main_Page (дата обращения 20.10.2010)	Portico Project	Portico Project
Средства РИМ, допускающие совместное выполнение моделей и натурных компонентов РВС РВ	МАК RTI – High Performance RTI // официальный сайт проекта, 2010 URL: http://www.mak.com/products/rti.php#dlform (дата обращения: 21.10.2010).	МАК	2010, МАК
Средства РИМ, допускающие совместное выполнение моделей и натурных компонентов РВС РВ	HLA RTI Performance Evaluation	L. Malinga Willem H. le Roux	2009, Council for Scientific and Industrial Research, Pretoria, South Africa

Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов PBC PB	Evaluation of Run Time Infrastructure (RTI) Implementations	Knight, A. Corder, R. Liedell	2002, US Army SMDC Wynn Drive, Huntsville
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов PBC PB	Evaluating the Performance of Publish/Subscribe Platforms for Information Management in Distributed Real-time and Embedded Systems	Ming Xiong, Jeff Parsons, James Edmondson, Hieu Nguyen, and Douglas C. Schmidt	2006, Vanderbilt University, Nashville TN, USA
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов PBC PB	POLLUX: ENHANCING THE QUALITY OF SERVICE OF THE GLOBAL INFORMATION GRID (GIG)	Douglas C. Schmidt	2009, The Vanderbilt University, Nashville TN, USA
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов PBC PB	An In-Depth Look at RTI Process Models	Mikael Karlsson and Peter Karlsson	2003, Pitch AB, Linköping, Sweden
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов PBC PB	Making RTI Tuning Easy with Performance Profiles	Björn Möller and Mikael Karlsson	2005, Pitch Technologies, Linköping, Sweden
Средства записи и хранения информации о поведении PBC PB	Advanced Memory Data Structures for Scalable Event Trace Analysis.	Andreas Knupfer	2009, Dissertation, Dresden University of Technology
Средства записи и хранения информации о поведении PBC PB	Open Trace Format API Specification Version 1.1.	Andreas Knupfer, Holger Brunst, Allen D. Malony, Sameer S. Shende	2006, Dresden University of Technology, Germany
Средства записи и хранения информации о поведении PBC PB	Epilog Binary Trace-Data Format	Felix Wolf, Bernd Mohr, Nikhil Bhatia, Marc-André Hermanns, Markus Geimer	2006, University of Tennessee
Средства анализа данных о поведении PBC PB	New Algorithms for Performance Trace Analysis Based on Compressed Complete Call Graphs.	Andreas Knupfer, Wolfgang E.Nagel	2005, Computational science – ICCS
Методы спецификации интерфейсов моделей компонентов PBC PB	Architecture Based Specification and Verification of Embedded Software Systems	Manfred Broy	2008, ISoLA 2008, CCIS 17, pp. 1–13.
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов PBC PB	Performance Estimation of Distributed Real-time Embedded Systems by Discrete Event Simulations	GaborMadl, Nikil Dutt, Sherif Abdelwahed	2007, EMSOFT'07, September 30 – October 3
Средства РИМ, допускающие совместное выполнение моделей	Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules	IEEE	2010, IEEE

и натуральных компонентов РВС РВ			
Методы спецификации интерфейсов моделей компонентов РВС РВ	Modeling and Simulation (M&S) High Level Architecture (HLA) — Object Model Template (OMT) Specification	IEEE	2000, IEEE
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	Portico Performance Reports – Protico // сайт, 2010 URL: http://www.porticoproject.org/index.php?title=Portico_Performance_Reports (дата обращения: 22.10.2010).	Portico	2010, Portico
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	Data Distribution Service for Real-time Systems, version 1.2.	Object Interface Systems, Inc; Real-Time Innovations, Inc; THALES.	2007, THALES
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	OMG Data-Distribution Service (DDS): Architectural Overview.	Real-Time Innovations, Inc.	2004, Real-Time Innovations, Inc.
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	NCWare & SimWare // Официальный сайт компании Nextel. URL: http://www.nexteleng.es	Nextel	2010, Nextel
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	EODiSP // Официальный сайт компании P&P Software. URL: http://www.pnp-software.com/	P&P Software	2010, P&P Software
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	CERTI // Официальный сайт лаборатории ONERA. URL: http://www.cert.fr/ (дата обращения 30.10.2010).	ONERA	2010, ONERA
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	MAK RTI // Официальный сайт компании MAK Technologies. URL: http://www.mak.com/ (дата обращения 30.10.2010).	MAK Technologies	2010, MAK Technologies
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	pRTI // Официальный сайт компании Pitch Technologies. URL: http://www.pitch.se/ (дата обращения 30.10.2010).	Pitch Technologies	2010, Pitch Technologies
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	GAIA // Официальный сайт разработчика (University of Bologna). URL: http://pads.cs.unibo.it/ (дата обращения 30.10.2010).	University of Bologna	2010, University of Bologna
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	Portico // Официальный сайт проекта Portico. URL: http://porticoproject.org/ (дата обращения 30.10.2010).	Portico	2010, Portico
Средства РИМ, допускающие совместное	NDDS // Официальный сайт компании Real-Time Innovations. URL: http://www.rti.com/ (дата обращения 30.10.2010).	Real-Time Innovations	2010, Real-Time

выполнение моделей и натуральных компонентов PBC PB			Innovations
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов PBC PB	OpenSPlice // Официальный сайт компании PrismTechnologies. URL: http://www.opensplice.com/ (дата обращения 30.10.2010).	PrismTechnologies	2010, Prism Technologies
Средства анализа данных о поведении PBC PB	Среда моделирования ДИАНА: Визуализатор временной диаграммы. Руководство оператора. Редлаб, 2009.	Редлаб	2010, Редлаб
Средства записи и хранения информации о поведении PBC PB	Проект V-Ray // Официальный сайт НИВЦ МГУ, URL: http://parallel.ru/v-ray/	НИВЦ МГУ	2010, НИВЦ МГУ
Средства записи и хранения информации о поведении PBC PB	Paje trace file format.	O. Stein, J. Chassin de Kergommeaux	2003, Universidade Federal de Santa Maria, RS, Brazil
Средства записи и хранения информации о поведении PBC PB	Open Trace Format (OTF) Tutorial. Presentation.	A. Knupfer, H. Brunst, A. D. Malony, S. S. Shende	2006, University of Dresden, Germany
Средства записи и хранения информации о поведении PBC PB	Epilog Binary Trace-Data Format.	F. Wolf, B. Mohr, N. Bhatia, M.-A. Hermanns, M. Geimer.	2006, University of Tennessee
Средства записи и хранения информации о поведении PBC PB	Scalable Log Files for Parallel Program Trace Data	A. Chan, W. Gropp, E. Lusk.	2010, Scalable Log Files for Parallel Program Trace Data
Средства записи и хранения информации о поведении PBC PB	Logfile Format // Mathematics and Computer Science Division, Argonne National Laboratory, URL: http://www.mcs.anl.gov/research/projects/perfvis/software/log_format/index.htm	A. Chan, B. Gropp, R. Lusk.	2010, Mathematics and Computer Science Division, Argonne National Laboratory
Средства записи и хранения информации о поведении PBC PB	Performance API // URL: http://icl.cs.utk.edu/projects/papi/wiki/Main_Page	Innovating Computing Laboratory	2010, Innovating Computing Laboratory
Средства записи и хранения информации о поведении PBC PB	VTF3 - A Fast Vampir Trace File Low-Level Management Library // Dresden University of Technology, Center for High-Performance Computing, Report ZHR-R-0304, Nov 2003.	S. Seidl	2003, Dresden University of Technology, Center for High-Performance Computing
Методы спецификации интерфейсов моделей компонентов PBC PB	Towards an HLA Run-time Infrastructure with Hard Real-time Capabilities	M. Adelantado, P. Siron, J.-B. Chaudron	2010, Toulouse, France
Методы спецификации интерфейсов моделей компонентов PBC PB	Security Architecture for Federated Cooperative Information Systems	P. Bieber, D. Raujol, P. Siron	2002, Toulouse, France

Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	EODiSP – AN OPEN AND DISTRIBUTED SIMULATION PLATFORM	I. Birrer, B. Carnicero-Dominguez, M. Egli, A. Pasetti	2006, Noordwijk, the Netherlands
Методы спецификации интерфейсов моделей компонентов РВС РВ	HLA-BASED ADAPTIVE DISTRIBUTED SIMULATION OF WIRELESS MOBILE SYSTEMS	L. Bononi, G. D'Angelo, L. Donatiello	2003, Universita degli Studi di Bologna
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	Running Real Time Distributed Simulations under Linux and CERTI	B. d'Ausbourg, P. Siron, E. Noulard.	2008, Toulouse, France
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	Parallel and Distributed simulation systems	R. D. Fujimoto	2000, Wiley Series on Parallel and Distributed Computing, ISBN-10: 0471183830
Методы спецификации интерфейсов моделей компонентов РВС РВ	Object-Oriented HLA - Does One Size Fit All?	B. Moeller, F. Antelius.	2010, Simulation Interoperability Standards Organization
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	HLA RTI Performance Evaluation	L. M. Roux	2009, Pretoria, South Africa
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	Grid-based HLA Simulation Support	K. J. Rycerz	2006, Krakow, Poland
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	Implementation of DDM in the MAK High Performance RTI	D. D. Wood	2002, Cambridge
Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ	A Distributed Simulation Backbone for Executing HLA-based Simulation Over the Internet // Proceedings of the 2nd International Conference on Scientific and Engineering Computation, pp. 96-103.	Y. Xie, Teo Y. Meng, W. Cai, S. Turner	2004, Workshop on Grid Computing & Applications

ПРИЛОЖЕНИЕ Г – Дополнительные информационные материалы

Ниже представлены расширенные материалы о программах для ЭВМ, перечень которых был указан в таблицах В.6.1 и В.6.2.

Г.1 Средства РИМ, допускающие совместное выполнение моделей и натуральных компонентов РВС РВ

Г.1.1 Стенд ПНМ, разработанный в ЛВК ВМК МГУ имени М.В.Ломоносова

Система полунатурного моделирования с поддержкой исполнения имитационной модели одновременно на нескольких инструментальных машинах, объединённых сетью. При этом, для обмена информацией используется несколько физических каналов связи:

1. Ethernet со стеком протоколов tcp/ip,
2. Ethernet для поддержки моделирования в реальном времени,
3. LPT для точной синхронизации.

К каждой инструментальной машине может быть подключён набор тестируемых устройств и привязан набор моделей таких устройств. Модели и устройства обмениваются между собой сообщениями.

Стек протоколов для поддержки реального времени

Во время прогона имитационной модели инструментальные машины связываются между собой через отдельный канал Ethernet по протоколу реального времени сетевого уровня RTEth. Протокол способен объединить до 31 машины. В качестве пакетов транспортного уровня используются пакеты rtms. Протокол RTEth поддерживает передачу транспортных пакетов, превышающих по размеру кадры Ethernet. Однако, в целях повышения производительности, используются пакеты меньшей длины (Концепция моделирования (по проекту с ОКБ "Сухого"), 2010).

Следует отметить, что если адресат сообщения (существующее устройство или его модель) расположен на той же инструментальной машине, что и отправитель, то на уровень RTEth rtms-пакет не передаётся.

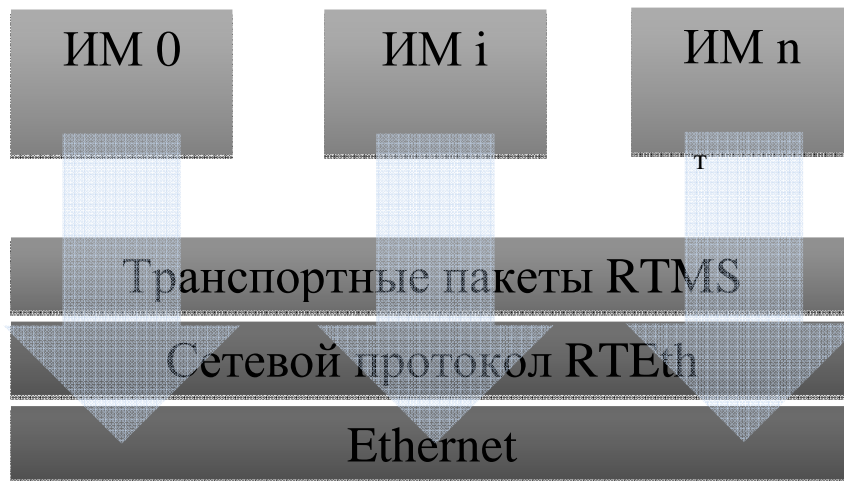


Рисунок 5. Стек протоколов обмена данными в сети инструментальных машин.

Средства управления экспериментом

Контроль над прогоном модели осуществляется с помощью средств оперативной визуализации. Одна из инструментальных машин сети, мастер эксперимента, получает через «обычный» Ethernet управляющие команды от средств визуализации: начать, приостановить, возобновить или завершить моделирование. Прогон моделей на вспомогательных машинах, в свою очередь, контролируется машиной-мастером путём передачи служебных сообщений через Ethernet канал реального времени. В текущей реализации мастером эксперимента всегда является машина с нулевым адресом. Она будет использоваться даже в том случае, если к ней не будет привязано ни одной устройство или модель. В дальнейшем ситуация, вероятно, изменится.

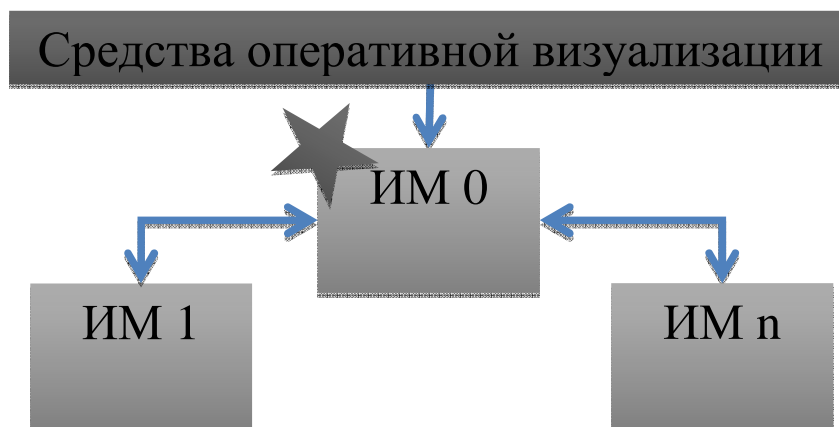


Рисунок 6. Управление выполнением имитационной модели.

Средства управления параметрами модели

На каждой инструментальной машине запускаются основной и вспомогательный процесс моделирования. Основной процесс обеспечивает моделирование целевой системы и заносит события результирующей трассы в разделяемую память. Вспомогательный процесс периодически сохраняет собранные файлы на диск и пересылает данные об изменении состояния модели средствам оперативной визуализации. После завершения моделирования файлы собранных трасс объединяются.

Существует так же возможность изменения значения конкретных параметров во время прогона с помощью средств оперативной визуализации. При этом новое значение сначала передаётся по сети вспомогательному процессу моделирования, а затем основному процессу через разделяемую память.

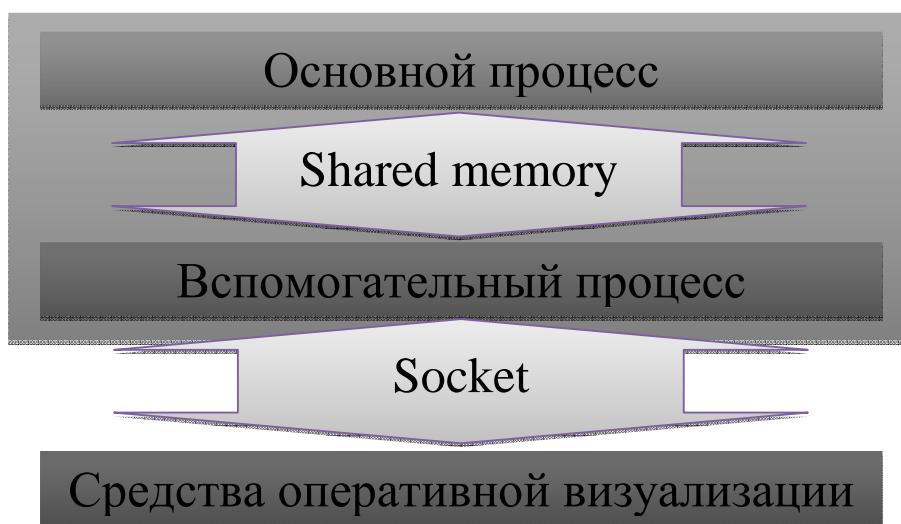


Рисунок 7. Процессная архитектура ПО инструментальной машины.

Схема работы среды выполнения системы «Стенд»

В рамках среды выполнения системы моделирования «Стенд» выделяют реальные устройства, модели (логические объекты) для имитации поведения таких устройств и монитор (диспетчер), обеспечивающий синхронизацию и взаимодействие процессов системы. Для элементов имитационной модели и диспетчера в рамках основного процесса моделирования создаются свои нити в соответствии с иерархией приоритетов:

1. драйверы устройств,
2. монитор,
3. модели устройств.

Инициализация моделирования

Основной процесс моделирования на каждой инструментальной машине распределённой сети получает конфигурацию ИМ и инициализирует соответствующие

компоненты: интерфейсы подключённых реальных устройств и модели таких устройств. При этом создаются таблицы, по которым во время прогона определяется машина, к которой привязан конкретный компонент модели. Прогон начинается после получения мастером моделирования соответствующей команды от средств оперативной визуализации. Непосредственно перед началом прогона между инструментальными машинами происходит двухэтапная синхронизация времени: предварительная синхронизация через канал Ethernet и более точная через LPT.

Планирование события

В системе «стенд» планировать события способны как модели устройств, так и мониторы. Существует несколько типов событий:

1. События управления процессом моделирования;
2. Служебные события keep-alive, для поддержки «холостой» работы монитора;
3. События, обеспечивающие обмен сообщениями между моделями устройств. При этом данные, соответствующие телу сообщения, передаются одним пакетом вместе с событием.

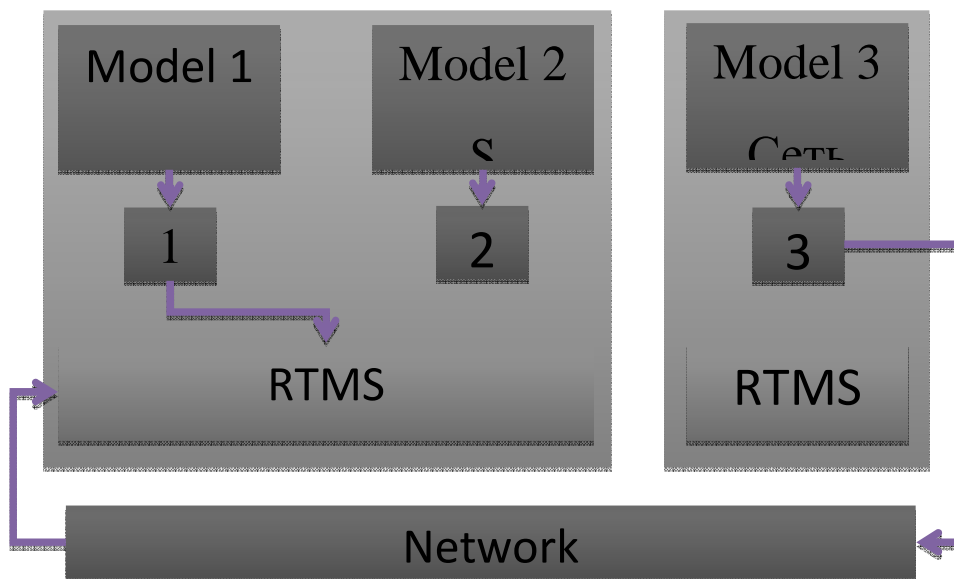


Рисунок 8. Передача сообщений в сети инструментальных машин.

Планирование любого события предполагает создание соответствующего rtms пакета через интерфейс, предоставляемый монитором и вне зависимости от взаимного расположения отправителя и адресата относительно инструментальных машин. Однако пакеты обрабатываются по-разному в зависимости от заданного адресата (цифры на рисунке соответствуют номерам перечисленных пунктов):

1. локальные пакеты помещаются в очередь пакетов для обработки,
2. пакеты, предназначенные для выключенных адресатов, игнорируются,
3. внешние пакеты передаются на соответствующую инструментальную машину.

Обработка событий

В нити монитора происходит циклическая проверка поступивших rtms-пакетов и выполняется их обработка. При этом могут, например, изменяться значения параметров моделей устройств.

Каждая модель устройства содержит очередь сигналов, которая пополняется во время обработки пакетов. Из моделей устройств с изменившимися очередями сигналов строится очередь изменённых моделей.

После завершения обработки поступивших rtms пакетов проверяются управляющие события от средств визуализации. В частности, переменным могут быть присвоены пользовательские значения.

Далее в нити монитора запускается проверка сигналов для каждой модели из очереди изменённых моделей. Из буфера выбирается сигнал с наибольшим приоритетом. Если данный сигнал способен вывести модель из текущей точки ожидания, то нить модели пробудится и снова сможет планировать события.

Завершение моделирования

Существует несколько вариантов остановки монитора:

1. Эксперимент остановлен пользователем,
2. Работа прекращена в связи с истечением времени моделирования,
3. Получен соответствующий управляющий сигнал от мастера эксперимента,
4. Произошла ошибка (например, устройство перестало посылать сообщения).

В текущей реализации проверка достижения модельным временем порога моделирования осуществляется только на мастере эксперимента, который при этом рассылает соответствующие управляющие пакеты остальным инструментальным машинам. Однако в дальнейшем, вероятно, каждая машина будет осуществлять подобный контроль.

Г.1.2 Существующие реализации HLA RTI

В результате исследования научных статей, посвящённых системам имитационного моделирования на основе стандарта HLA, за последнее десятилетие был выделен набор признаков для сравнения выделенных реализаций таких систем между собой.

1. Поддерживаемая версия HLA

За время существования стандарта HLA было выпущено несколько основных его версий: HLA 1.3, HLA 1516-2000 и HLA 1516-2010 (Evolved). На данный момент большинство реализаций поддерживают большую часть функциональности стандарта HLA 1516-2000. HLA 1516-2010 вышел сравнительно недавно, и большинство продуктов его не поддерживают или поддерживают лишь некоторые нововведения.

1.1. Поддержка моделирования в реальном времени

Как показали проведённые исследования [44], стандарт HLA может быть использован для построения средств моделирования реального времени. Многие из рассмотренных RTI послужили базой для разработки таких средств [48]. Однако, основные ветви продуктов, как правило, не предполагают его поддержки. Исключение составляет средство NCWare, основанное на стандарте DDS.

1.2. Поддержка технологии GRID

Существует множество примеров использования GRID в качестве сети для распределённого моделирования для стандарта HLA [56, 58]. Однако подобные продукты специфичны и не являются основной ветвью разработки систем RTI.

1.3. Object-Oriented HLA (OO-HLA)

Обычно взаимодействие федерата и RTI осуществляется через предоставляемый RTI унифицированный интерфейс. Часто разработчики пытаются упростить использование RTI с помощью объектно-ориентированной обёртки, специально созданной для конкретной модели FOM. Существуют средства автоматизированного создания подобных обёрток. Подобное средство входит, например, в пакет программ моделирования от компании Pitch [52].

2. Поддержка моделирования в реальном времени

2.1. Механизмы контроля качества соединения

Существуют реализации, основанные на CERTI и поддерживающие моделирование в жёстком реальном времени, однако это коммерческие продукты [48]. Для обеспечения качества сервиса используются протоколы RSVP и VRTP.

2.2. Поддержка гибридной синхронизации

Существуют средства (Mello & Wagner) позволяющие части федераций работать в синхронном режиме, а части – в асинхронном, одновременно.

2.3. Поддержка синхронизации федератов в цикле моделирования

Традиционно считается, что на каждом шаге моделирования в реальном времени федерат проходит через 4 фазы: получение, обработка, отправка и простой [49]. Существуют реализации RTI, позволяющие для увеличения точности вычислений проводить синхронизацию федератов перед каждой итерацией описанного цикла [44].

2.4. Поддержка распределённого (географически) полунатурного моделирования

Система EODiSP разрабатывалась для моделирования стыковки кораблей к Международной Космической Станции (МКС), и предполагает использование полунатурных моделей, удалённых друг от друга. Передача данных между моделями осуществляется через сеть и требует определённое время, что порождает ошибки моделирования. Однако использование механизма предсказания текущего значения данных по уже полученным результатам позволяет минимизировать ошибку и достичь приемлемой точности моделирования [46].

3. Внутренняя архитектура RTI

3.1. Разделение RTI на компоненты

Как правило, реализация RTI разделена на несколько компонентов. Так для обеспечения федератов стандартным интерфейсом HLA обычно используются компоненты, размещаемые на инструментальных машинах локально. В то же время для реализации глобальных алгоритмов работы с федерацией обычно предоставляется отдельный компонент RTI.

Архитектура CERTI

На каждой инструментальной машине работает локальный процесс RTI Ambassador (RTIA). Федераты, выполняемые на данной инструментальной машине, общаются с RTIA через библиотеку libRTI, обладающей HLA-совместимым интерфейсом. Взаимодействие федерата и RTIA осуществляется через сокет Unix. Кроме того, существует выделенный центральный компонент - глобальный процесс RTI Gate (RTIG), отвечающий за распределённые алгоритмы реализации некоторых служб RTI [54].

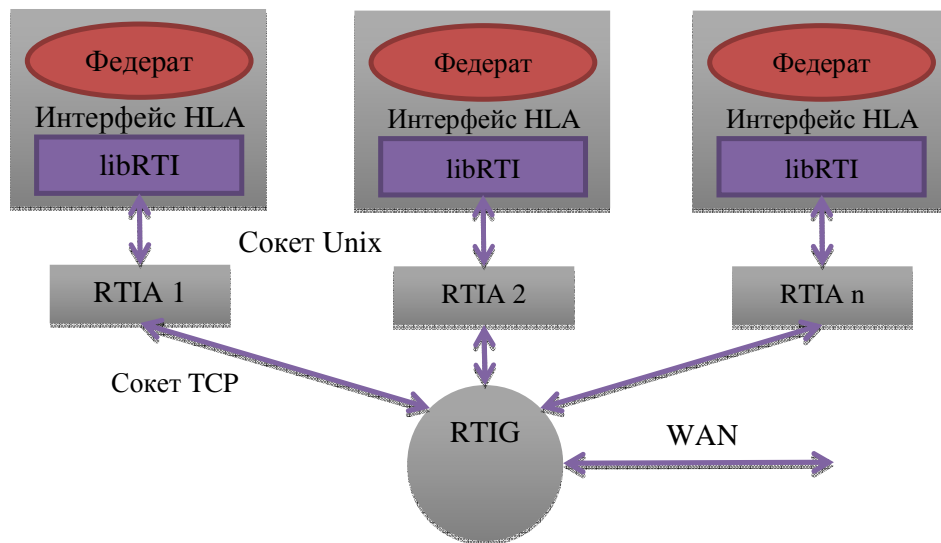


Рисунок 9. Архитектурная модель CERTI.

3.2. Типы используемых процессов

Существуют как средства, использующие для разделения задач моделирования полновесные процессы (CERTI), и так и средства, основанные на процессах-нитех (МАК). Использование независимых процессов позволяет разделить настройки соединения между инструментальными машинами и настройки средства моделирования, однако влечёт за собой лишние расходы на обмен данными. Использование же нескольких нитей позволяет ускорить обмен данными, но усложняет установку и настройку приложения в целом.

3.3. Модель используемых процессов

Стандарт HLA не задаёт спецификации на используемую модель процессов для взаимодействия RTI и федерата, и среди современных реализаций существует несколько вариантов таких моделей [50].

3.3.1. single-threaded (поддерживается в RTI:NG)

Федерат посылает запрос RTI. Запрос обрабатывается в той же нити (возможно, с использованием обратных вызовов) и возвращает управление федерату.

3.3.2. asynchronous (реализовано в RTI:NG и pRTI)

Внутри RTI работает несколько нитей, которые возвращают управление нити федерата сразу после получения её запроса.

3.3.3. multi-threaded (поддерживается в pRTI)

Внутри RTI работает несколько нитей, самостоятельно совершающие вызовы к федерату. При этом федерат никаких запросов не делает: он лишь должен быть готов к обработке вызова со стороны RTI.

3.4. Цикл работы федерата

Традиционно федерат в моделировании работают циклически [49]. Однако цикл работы федерата может быть реализован по-разному [50].

3.4.1. Активное ожидание (polling)

Процесс-федерат циклически засыпает на предустановленное время, и обрабатывает пришедшие события.

3.4.2. Пассивное ожидание (Interrupt-driven)

Процесс-федерат в цикле ожидает поступления новых событий и, если такие появились, сразу их обрабатывает.

3.5. Одновременный доступ федератов к сервисам RTI

Во время прогона имитационной модели часто случается ситуация, когда несколько федератов пытаются получить доступ к сервисам RTI одновременно. Возможны несколько реализаций обработки такой ситуации [50].

3.5.1. Выбрасывание исключения (balking mode)

Когда федерат пытается получить сервис RTI, предоставленный в данный момент другому федерату, выбрасывается исключение. При этом управление сразу возвращается нити федерата, что теоретически позволяет достичь большей эффективности использования процессора.

3.5.2. Синхронная обработка

При использовании этого подхода RTI задерживает нить второго федерата, и предоставляет ему сервис после его освобождения первым федератом. Преимуществом данного решения является сравнительная простота реализации как RTI, так и процесса федерата (используется в rRTI).

3.6. Поддержка использования разделяемой памяти

Если несколько федератов работают на одной инструментальной машине, то обмен данными между ними может быть произведён через разделяемую память, что эффективнее передачи через сетевой интерфейс. Подобной функциональностью обладают многие рассмотренные системы [47].

3.7. Сетевая передача данных

3.7.1. Стек протоколов

Некоторые реализации HLA ориентированы на конкретные задачи (например, поддерживают распределённое моделирование реального времени) и используют особые протоколы передачи данных между инструментальными машинами. Базовая версия CERTI поддерживает взаимодействие RTIA и RTIG только через сокет TCP и UDP [54].

3.7.2. Оптимизации сетевого взаимодействия

Некоторые RTI (например, pRTI) поддерживают передачу нескольких событий одним пакетом (piggy-backing). Кроме того, существуют механизмы ограничения числа пакетов одновременно находящихся в сети и сбрасывания пакетов, невостребованных до установленного временного порога [53].

4. Возможность защиты федератов и их компонентов

Часто в распределённом моделировании участвуют коммерческие системы. Через RTI может быть получена некоторая информация о ключевых идеях их реализации. Существуют средства, позволяющие контролировать доступ к подобной информации.

Защита данных в CERTI

Несколько механизмов защиты частных данных было предложено для системы моделирования CERTI [45]. Для осуществления контроля над федератами и другими элементами федераций используются специальные метки безопасности. При этом доступ к данным контролируется на нескольких уровнях: возможна иерархическая организация меток.

Контроль безопасной передачи данных между федератами осуществляет RTIG. Для каждого отправителя RTIG строит список получателей. При этом производится проверка меток безопасности данных и федератов на совместимость друг с другом.

Важно отметить, что контроль осуществляется на этапе подписки на получение данных, что не влияет на эффективность передачи данных (что наиболее критично с точки зрения общей производительности системы).

Производители могут запускать модели своих продуктов на собственных серверах и проводить моделирования с участием сторонних разработчиков через сеть интернет. Для защиты информации при передаче через глобальную сеть при этом используется Generic Security Services Application Programming Interface (GSS-API). В случае распределённого моделирования на локально соединённых инструментальных машинах, предоставленных различными компаниями участниками, для построения безопасной локальной сети предлагается использовать протокол Secure Medium Access Control (SMAC).

5. Возможность динамического перераспределения нагрузки между машинами

Общая производительность распределённого моделирования может быть увеличена за счёт динамического перераспределения нагрузки (исполняемых федератов) между инструментальными машинами сети. Значительные исследования в этой области были проведены в процессе разработки системы GAIA, основанной на RTI ARTIS [47].

6. Механизмы оптимизации HLA Data Distribution Management (DDM)

DDM отвечает за распределение (передачу) данных между федератами. Используя сервисы DDM, федерат может указать, насколько важны для него конкретные данные. На основе полученной информации RTI может оптимизировать использование сетевых ресурсов, и, тем самым, увеличить общую производительность системы [57].

6.1. Фильтрация данных на получателе (receive filtering)

Данный механизм может использоваться как самостоятельно, так и после применения грубых первичных фильтров.

6.2. Фиксированная сеть (fixed grid)

Широковещательная передача данных производится не всем инструментальным машинам, а лишь небольшой их подгруппе (применяется в CERTI). Машины объединяются по диапазону адресов, заданному в конфигурационных файлах до начала моделирования. Данный метод требует дополнительной фильтрации данных на получателе.

6.3. Распределённые области (distributed region)

Группы для широковещательной передачи определяются динамически во время моделирования. Требуется дополнительная фильтрация данных на получателе.

7. Переносимость

Средство Portico кроссплатформенно и поставляется в двух вариантах: на языке C++ и на языке Java. Однако использование версии C++ требует установки Java Run-Time Environment (JRE).

Реализация CERTI написана на языке C++. На данный момент она успешно применяется на нескольких платформах, включая разные типы Linux, Windows, Solaris и IRIX.

Г.1.3 Существующие реализации стандарта DDS

Спецификации OMG DDS определяют стандарт взаимодействия процессов, применимый к широкому классу распределённых систем реального времени и встроенных систем (DRE). В основе DDS лежит ориентированная на данные модель с архитектурой вида издатель-подписчик (DCPS). Процессы, использующие DDS, могут производить чтение и запись типизированных данных. При этом модель DCPS образует прослойку, которая позволяет читающим данные процессам получить доступ к самой последней их версии. В рамках DCPS определяются глобальное пространство данных, которые можно читать и изменять, и глобальное пространство имён, позволяющее создавать новые и искать существующие разделяемые объекты [18].

Процесс-издатель (publisher), желающий создать разделяемый объект, делает соответствующие записи в глобальных пространствах имён и данных. Аналогично, процесс-подписчик (subscriber) может найти интересующие его объекты в пространстве имён и получить доступ к соответствующим данным. При этом разделяются объявление о необходимости использовать разделяемые данные и непосредственное их использование, что позволяет применять в рамках DDS соединения с контролем качества.

К основным понятиям DDS относятся:

Домен. Процессы DDS могут обмениваться между собой информацией лишь через общий домен данных, что позволяет оптимизировать передачу информации внутри соответствующей группы процессов и изолировать её от процессов, которым эта информация не нужна.

Пары читатель/писатель и издатель/подписчик. Процессы используют писателей данных, чтобы поместить информацию в глобальное пространство данных, и читателей данных – чтобы получить эту информацию. Издатель создаёт и управляет набором писателей с одинаковым поведением и качеством соединения. Подписчик создаёт и управляет группой читателей.

Топик (Topic). Топик служит для соединения писателя и читателей. Взаимодействие между процессами внутри домена происходит лишь в том случае, если топик, объявленный писателем, соответствует топику, к получению которого присоединился читатель. Взаимодействие через топики анонимно и прозрачно, то есть издатели и подписчики не заботятся о том, как создаются и управляются топики, и кто записывает/считывает их данные. Решением этих вопросов занимается слой DCPS [17].

Стандарт DDS определяет большое количество политик обеспечения качества соединения и способов обмена топиками между взаимодействующими процессами. Однако стандарт не специфицирует то, как должно быть реализовано внутреннее управление данными, оставляя разработчикам широкую область для реализации своих идей. Там не менее, модель взаимодействия, архитектура, реализация DDS и её конфигурация оказывают значительное влияние на поведение процессов системы и определяют применимость к конкретному типу систем DRE [20].

Сравнение различных реализаций DDS между собой

Для исследования было выбрано несколько реализаций DDS:

Таблица 5. Реализации стандарта DDS.

Средство	Разработчик	Доступность кода
NDDS	Real-Time Innovations [28]	Закрытый
OpenSPlice	PrismTechnologies [29]	Открытый
Open DDS	Object Computing, Inc. [30]	Открытый

В результате изучения трудов конференций посвящённых предметной области было выделено несколько критериев для сравнения различных реализаций между собой.

1. Взаимодействие участников обмена данными

1.1. Стандарт DDS позволяет реализациям использовать преимущества различных моделей обмена данными (одноадресной, многоадресной и широковещательной)

Таблица 6. Поддерживаемые модели обмена данными.

Средство DDS	Модель обмена данными		
	Одноадресная (unicast)	Многоадресная (multicast)	Широковещательная (broadcast)
NDDS	Да (по умолчанию)	Да	Нет
OpenSPlice	Нет	Да	Да (по умолчанию)
Open DDS	Да (по умолчанию)	Нет	Нет

1.2. Используемые протоколы передачи данных

Все исследуемые реализации DDS используют для поддержки различных моделей обмена данными сетевой протокол IP. Но возможны и другие решения – например, основанные на протоколе Ricochet, который совмещает поддержку взаимодействия группы IP и упреждающее исправление ошибок, что позволяет достичь высокого уровня надёжности передачи с настраиваемыми и фиксированными накладными расходами [19].

2. Каждая из рассматриваемых реализаций DDS использует свою архитектурную модель

Таблица 7. Архитектурные модели реализаций DDS.

Средство	Версия	Архитектура
NDDS	4.1c	Децентрализованная
OpenSPlice	2.0 Beta	Федеративная
Open DDS	8.0	Централизованная

2.1. Федеративная модель

Для каждого сетевого интерфейса инструментальной машины создаётся отдельный процесс-демон DDS (см. рис.10). Демон отвечает за реализацию модели DSPC и запускается на каждом узле до того, как пользовательские процессы начинают обмен данными. Начав работу, он устанавливает каналы передачи данных с другими компьютерами сети в соответствии с заданными настройками (например, контроль качества и адреса узлов). Каждый конкретный канал отвечает за обработку всех запросов подходящей конфигурации.

Использование фонового процесса позволяет отделить пользовательский процесс от деталей настройки соединения и конфигурации модели DCPS. Например, демон может

сохранять параметры соединения в отдельном файле. Тогда изменение конфигурации сети не потребует изменения кода пользовательского процесса и/или параметров его запуска.

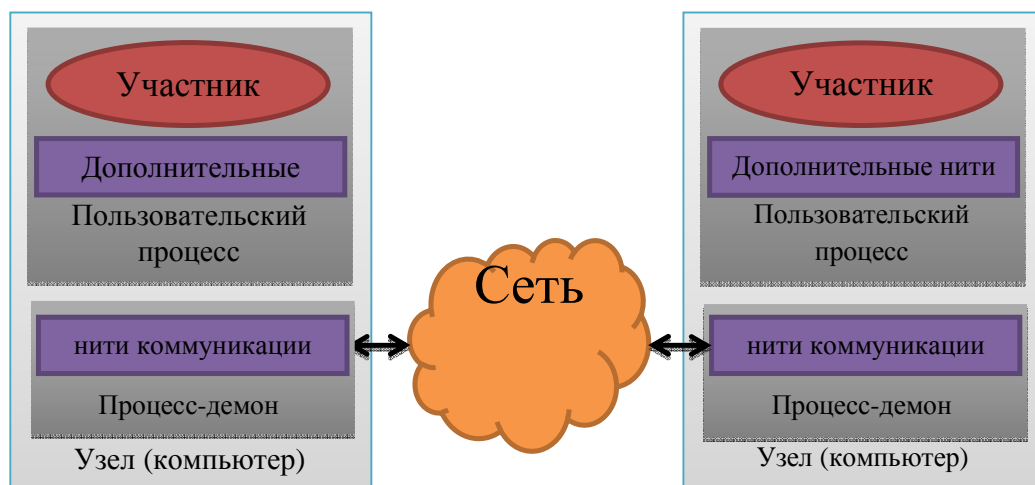


Рисунок 20. Федеративная модель.

Использование федеративной архитектуры позволяет легко изменять число участников, расположенных на одной машине (процесс-демон может эффективно обрабатывать сообщения от нескольких пользовательских процессов одновременно). Кроме того, использование отдельного процесса упрощает настройку пользовательских процессов, использующих один и тот же сетевой интерфейс, и позволяет задавать приоритет сообщениям из разных каналов.

Минусом архитектуры являются необходимость двухэтапной настройки (процесса-демона и пользовательского процесса) и дополнительные расходы на межпроцессное взаимодействие.

2.2. Децентрализованная модель

Децентрализованная модель (см. рис.11) совмещает средства коммуникации и конфигурации в одном полномесном процессе, разделяя их, однако, по разным нитям (в отличие от федеративной архитектуры, использующей отдельные процессы-демоны).

Плюсом децентрализованной архитектуры является самодостаточность приложения (не нужно дополнительно использовать отдельный процесс демон). Как следствие, упрощается модель обмена данными, и уменьшается время передачи между приложениями, использующими сервисы DDS, и его разброс.

Существуют в децентрализованной архитектуре и недостатки. Так модель предполагает, что настройка параметров соединения (например, адреса для отправки многоадресных сообщений, номера портов, надёжность) производится на уровне приложения, что усложняет приложение и ведёт к лишним ошибкам. Кроме того, данная

архитектура усложняет взаимодействие через DDS нескольких приложений на одной машине и приводит к дополнительным сложностям при масштабировании.

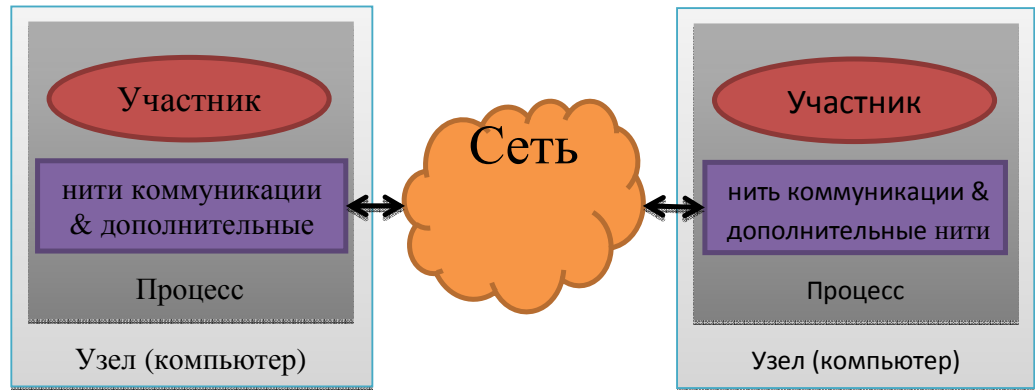


Рисунок 11. Децентрализованная модель.

2.3. Централизованная модель

Централизованная архитектура (см. рис.12) предполагает наличие машины-сервера. Сервер хранит информацию, необходимую для создания и управления соединениями между пользовательскими процессами внутри домена данных. Данные при этом передаются напрямую между процессами, в то время как управление и инициализация (например, регистрация данных, создание топиков, установка контроля качества) требует взаимодействия с сервером.

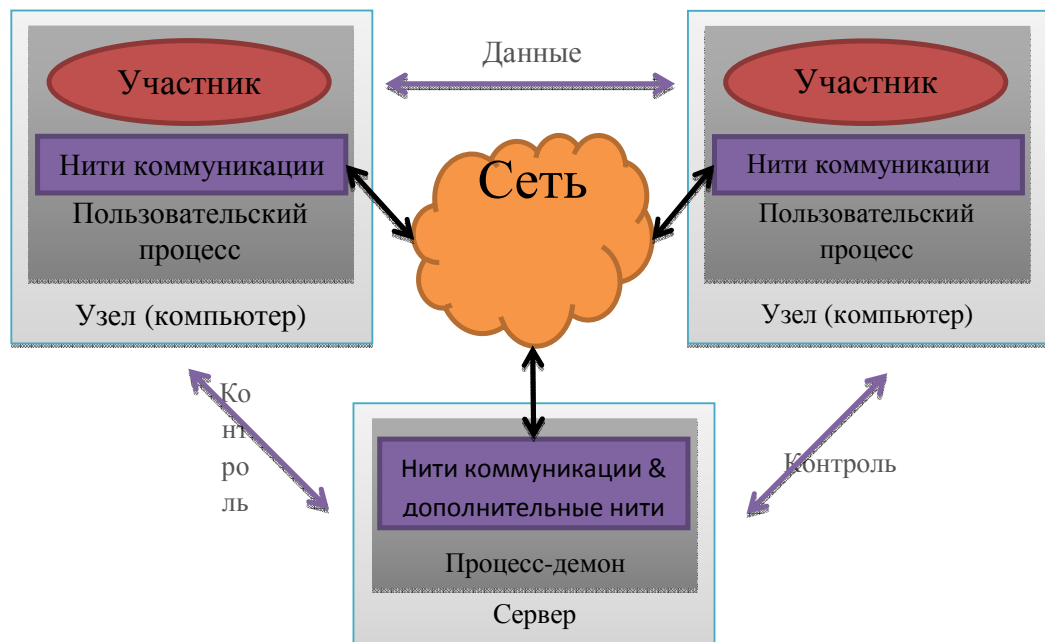


Рисунок 12. Централизованная модель.

Основное преимущество централизованной архитектуры – простота её реализации и настройки (вся необходимая для контроля информация сосредоточена в одной точке).

Здесь же кроется и основной недостаток данной реализации – сервер часто оказывается «узким местом».

3. Экспериментальное исследование архитектур DDS

Для сравнения эффективности существующих реализаций между собой было проведено несколько тестов [20] со следующим сценарием: издатель записывает данные заданного размера в пространство данных, а подписчик считывает данные и посылает издателю уведомление. Были проведены пробы для размера данных от 4 до 16384 байт с шагом, соответствующим степени двойки. Кроме того, тестовые данные отличались своей структурой: были выделены «простой» и «сложный» (требует дополнительной обработки) типы данных.

В первом виде тестов (рисунки 13-16) издатель и подписчик располагались на одной инструментальной машине. При этом на издателе замерялось время, прошедшее с момента отправки данных до получения уведомления. Реализации NDDS и OpenSplice при этом переключились на использование разделяемой памяти и показали значительно лучший результат в сравнении с Open DDS, передававшей данные через сетевой интерфейс.

В большинстве тестов абсолютный результат показала децентрализованная архитектура NDDS, обрабатывающая данные в рамках одного полновесного процесса. Однако OpenSplice лучше проявила себя на тестах с большими объёмами (более 512 байт) сложных данных. Такие результаты обусловлены тем, что NDDS проводит полный разбор данных даже в рамках одной машины, в то время как OpenSplice проводит лишь необходимую часть разбора.

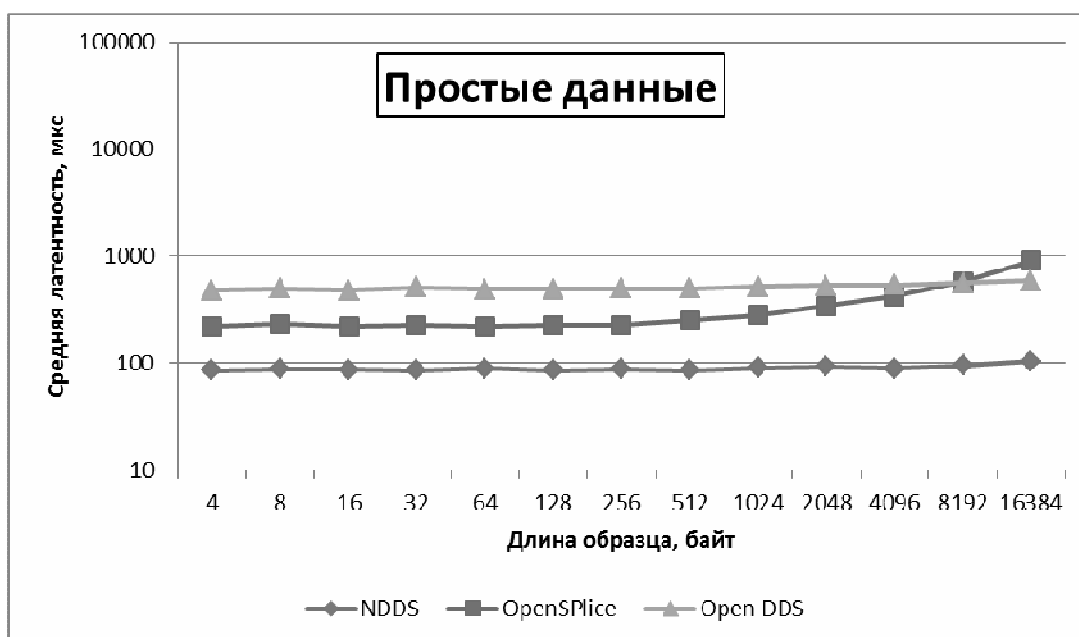


Рисунок 13. Латентность различных реализаций DDS на тестах с «простыми» данными.

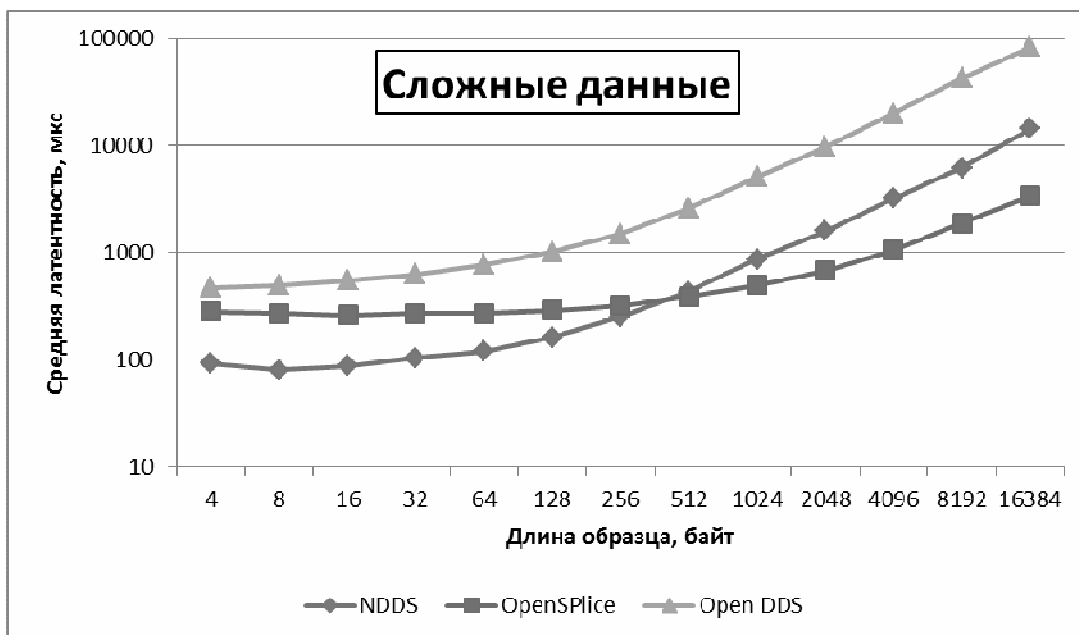


Рисунок 14. Латентность различных реализаций DDS на тестах со "сложными" данными.

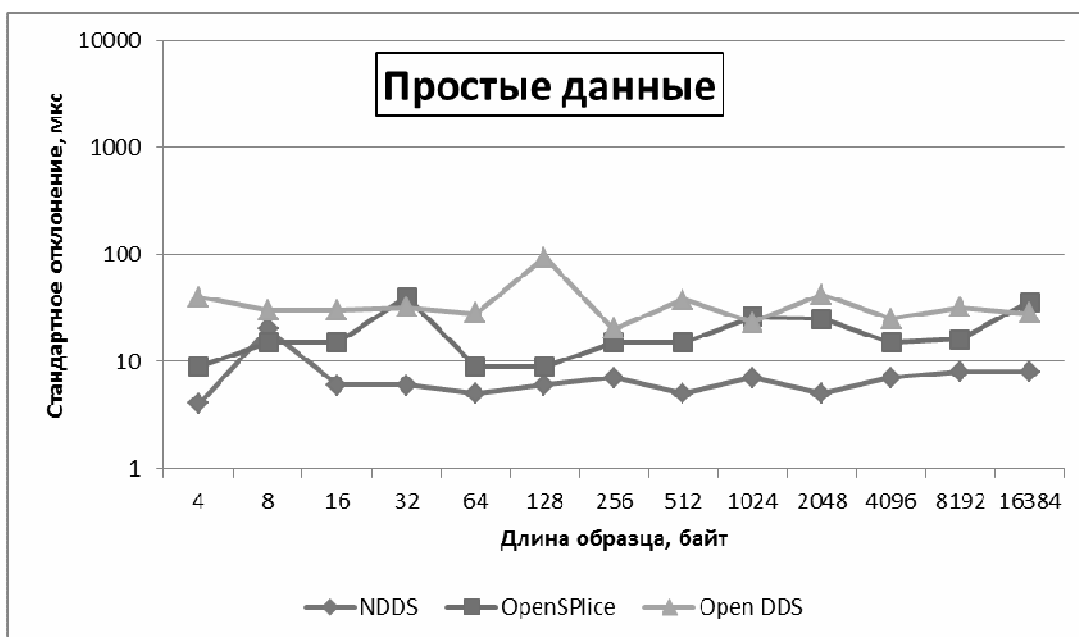


Рисунок 15. Стандартное отклонение различных реализаций DDS на тестах с "простыми" данными.

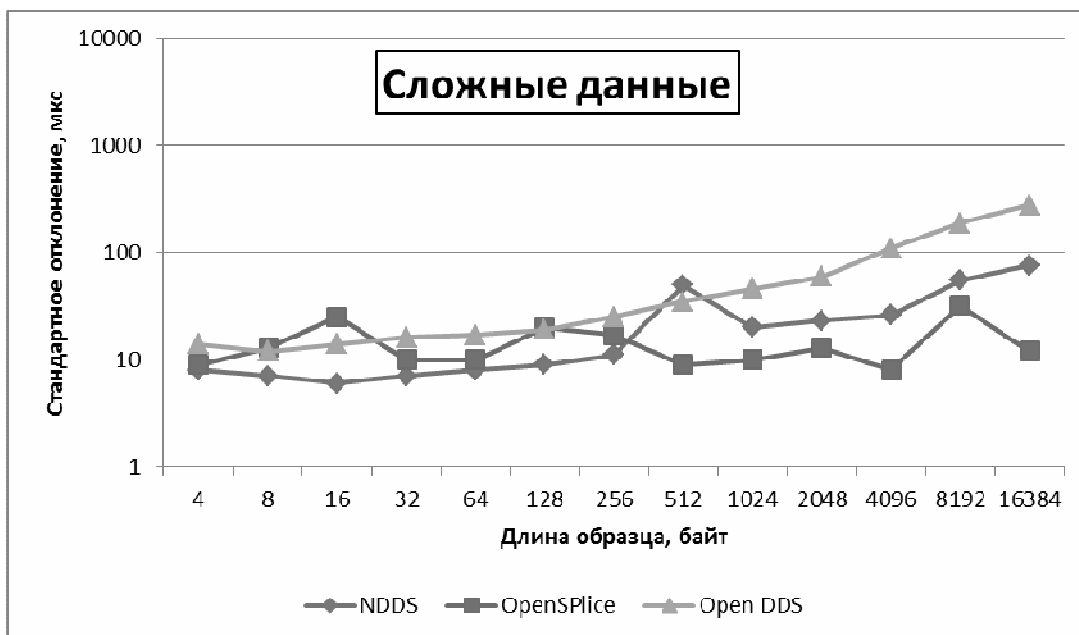


Рисунок 16. Стандартное отклонение различных реализаций DDS на тестах с "простыми" данными.

Другим важным показателем эффективности систем DRE может служить её пропускная способность. Поэтому был проведён ряд тестов для выявления зависимостей между пропускной способностью и различными конфигурациями DDS (были проведены пробы с числом процессов-подписчиков равным 4 и 12, различными моделями передачи данных и способами её доставки). Чтобы упростить масштабируемость тестовой системы, издатель и каждый подключаемый подписчик были запущены на своей отдельной машине.

Реализация NDDS поддерживает одноадресную и многоадресную модели передачи данных. Был поставлен соответствующий тест производительности, показавший, что с ростом числа подписчиков, пропускная способность системы с одноадресной моделью падает значительно быстрее, чем системы, основанной на многоадресной модели (рисунок 17). Это обусловлено необходимостью издателя передавать несколько (по числу подписчиков) копий данных в одноадресном режиме, в то время как использование многоадресной модели позволяет передать данные единственный раз.

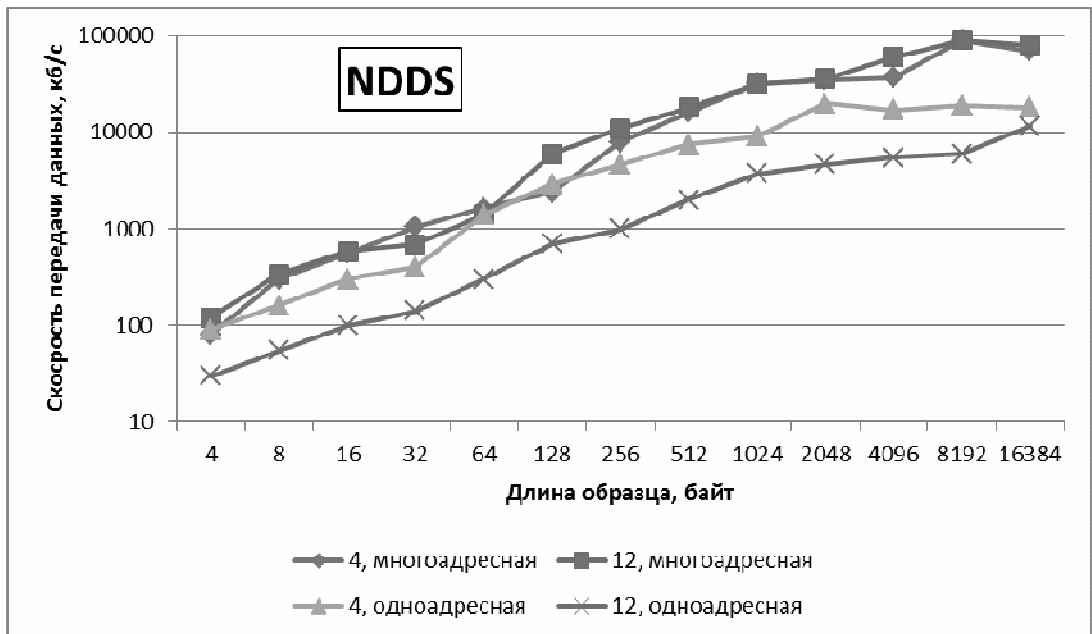


Рисунок 17. Сравнение скорости передачи данных в одноадресном и многоадресном режимах реализации NDDS.

Реализация OpenSPlice поддерживает многоадресную и широковещательную передачу данных. Проведённые тесты (рисунок 18) показали, что с увеличением числа подписчиков многоадресная модель передачи данных даёт несколько лучшую производительность. Более того, использование широковещательной передача данных сопряжено с риском излишней нагрузки на сеть и процессоры вычислительных узлов.

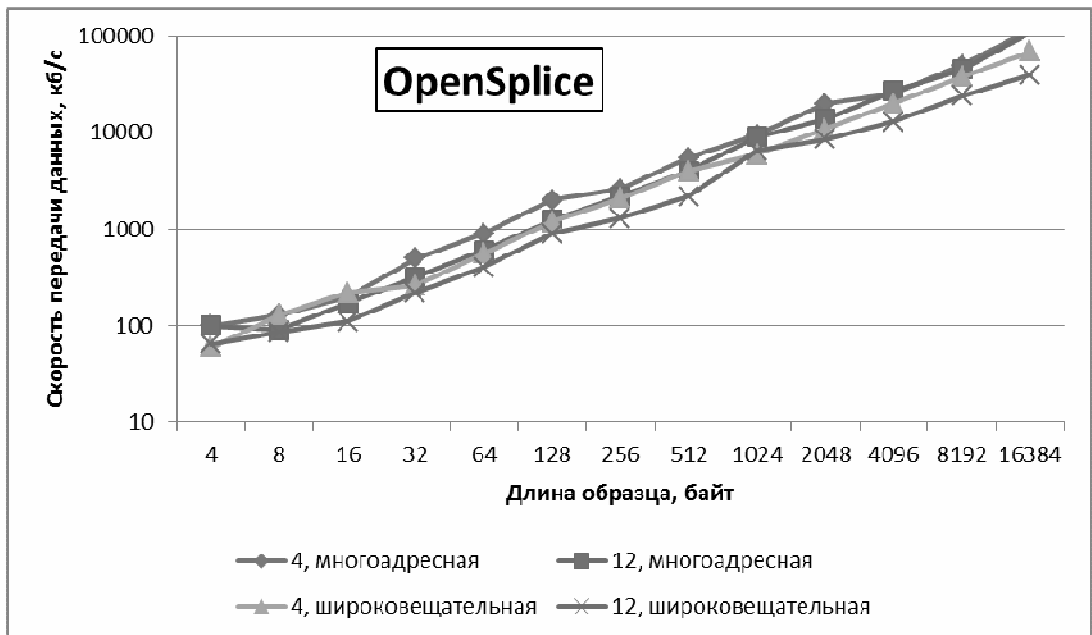


Рисунок 18. Сравнение скорости передачи данных в многоадресном и широковещательном режимах реализации OpenSPlice.

Проведённые тесты так же позволяют сравнить между собой пропускную способность различных реализаций DDS с одинаковыми моделями передачи данных

(образуются пары NDDS/OpenSPlice для многоадресной передачи (рисунок 19) и NDDS/Open DDS для одноадресной передачи данных (рисунок 20)). Тесты для многоадресной передачи продемонстрировали, что средство NDDS имеет лучшие показатели на данных размером до 8 килобайт. На данных размером 16 килобайт и более OpenSPlice опережает NDDS. При использовании одноадресного режима передачи NDDS немного опережает Open DDS на большинстве тестов.

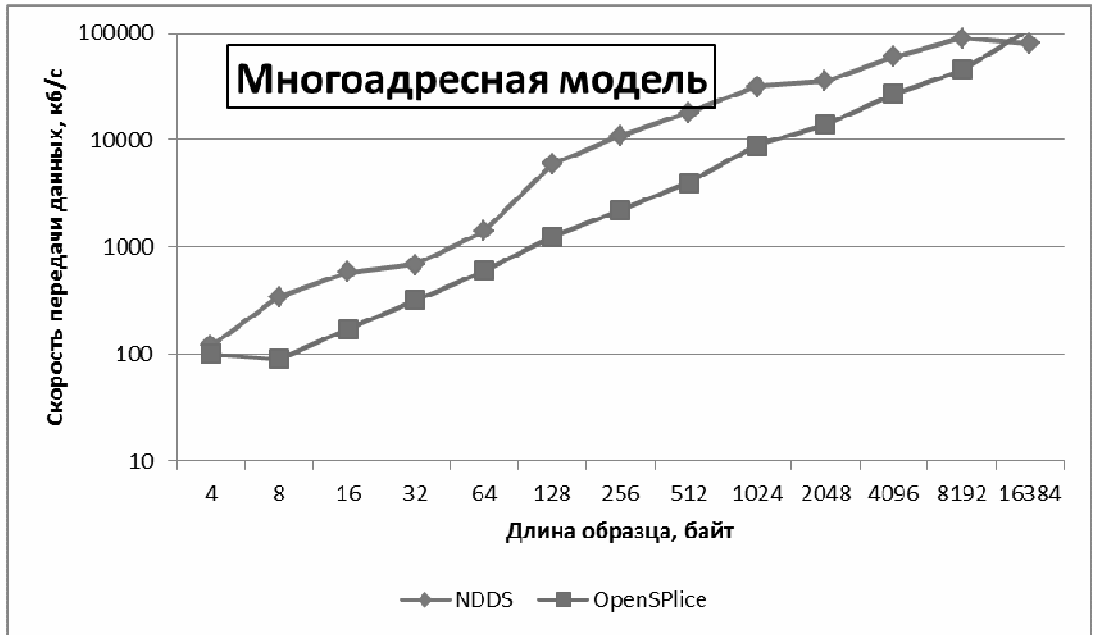


Рисунок 19. Сравнение скорости передачи данных в многоадресном режиме для реализаций NDDS и OpenSPlice.

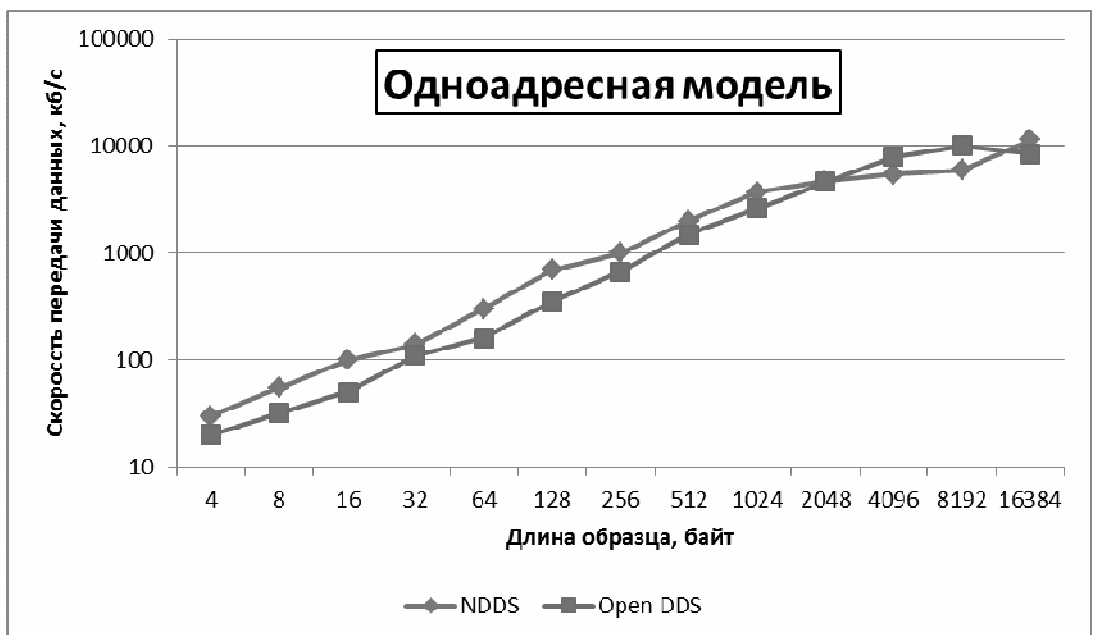


Рисунок 20. Сравнение скорости передачи данных в одноадресном режиме для реализаций NDDS и Open DDS.

Г.2 Существующие форматы записи хранения трасс событий

Трассы можно разделить по типу используемых структур данных для хранения информации о событиях:

- Непоследовательные структуры данных (графы) – CCG формат,
- Последовательные структуры данных - остальные.

Все форматы файлов трасс разделяют похожий базовый дизайн и имеют много общего. Прежде всего, они хранят информацию в (более или менее) чистом и необработанном виде. Кроме того, события трассировки представляются с помощью, так называемых записей трассы (trace records), которые покрывают одно событие в некоторый момент времени и являются наименьшей единицей информации во всех форматах. Все типы записей можно разделить на две категории [31]:

- типы записей событий,
- типы записей определений.

Как правило, существуют отдельные типы записей для всех различных типов событий.

Записи событий

Типы записей событий описывают реальные события, т. е. отличительные события во время выполнения. Все типы записей событий содержат, по крайней мере, временную метку, которая указывает момент, когда событие произошло. Кроме того, большинство записей событий поддерживают локализацию места (возникновения события). Это либо логическое расположение по отношению к структуре приложения (например, ID потока или процесса) или физическое расположение по отношению к вычислительной системе (например, машина, вычислительный узел, процессора, ядра, и т.д.). Кроме времени и места, есть дополнительные свойства событий, которые являются специфическими для определенных типов событий.

Записи событий составляют подавляющее большинство записей в больших трассах. Таким образом, эффективное кодирование записи событий очень важно, поскольку это напрямую влияет на общий размер трассы. Записи событий располагаются в последовательном потоке или в несколько параллельных потоков. Всегда, события сортируются в соответствии с их первоначальным временным порядком.

Наиболее важные типы записей событий поддерживаются всеми основными форматами трасс в очень похожем виде, такие как:

- Вход и выход из функций/узлов,
- Отправка и получение сообщений типа «точка - точка»,
- Групповые операции связи MPI, и

- Замеры аппаратных счетчиков производительности.

Другими примерами являются:

- операции ввода/вывода,
- блокирование и освобождение мьютексов,
- начало и конец параллельных OpenMP областей.

Записи определений

Кроме записей событий все форматы поддерживают ряд так называемых типы записей определений (definition record types). Эти записи обеспечивают различные глобальные свойства, необходимые для последующего анализа и для удобства. К ним относится, шаг таймера для всех временных меток, информация о дате и времени создания трассы и информация о платформе (имя хоста, тип процессора, скорость процессора, объем памяти и т.д.).

Специальные типы записей определений необходимы для повышения эффективности хранения, в частности, для меток, названий или описаний, таких как имена функций, имена вычислительных узлов, пути к исходным файлам и т.д. Такие названия являются произвольным длинным текстом и на них могут ссылаться очень часто. В целях повышения эффективности кодирования, эти метки отображаются в лексемы целого типа (идентификаторы) фиксированного размера, определенного записями. Лексемы используются всякий раз, когда ссылка на метку необходима в записях событий или в других записях определений. Это экономит место для хранения нескольких ссылок.

Типичными примерами для лексем записей определений являются:

- определения процессов,
- определения групп процессов,
- определения функций,
- определения групп функций,
- определения места в исходном коде,
- определения счетчиков производительности.

В отличие от записей событий, записи определений, как правило, не являются критическими с точки зрения их количества и размера их хранения. Даже если есть достаточно много определений, число событий, ссылающихся на эти определения, как правило, намного выше.

Библиотечные интерфейсы

Для всех форматов трасс есть поддержка библиотек, которые предоставляют интерфейсы для чтения и записи в файлы трассировки, такие, что проблемы кодирования и синтаксического анализа (декодирования), могут быть скрыты от программиста и

проблемы платформ, такие как размеры типов переменных, могут быть прозрачно решены.

Все библиотеки форматов трасс поддерживают стандартную схему доступа, где события могут быть прочитаны и записаны во временном порядке. Как правило, библиотеки предоставляют набор методов для написания отдельных типов записей. Для чтения, события трассы доставляются потребителю через обратный вызов обработчиков, которые должны быть зарегистрированы в библиотеке заранее. Опять же, есть отдельные обработчики для различных типов записей.

В дополнение к этому, некоторые библиотеки форматов трасс обеспечивают более или менее эффективный селективный доступ на чтение, либо в отношении некоторых типов записей или касающихся отдельных процессов в параллельных трассах или, касающихся отдельных временных интервалов. Кроме того, все форматы трасс включают в себя средства поддержки, которые, в частности, необходимые для сбора трасс нескольких процессов в единую параллельную трассу.

1 Формат трассы TAU

Формат TAU нацелен на анализ производительности параллельных программ, написанных на языках Fortran, C, C++, Java, Python [31].

TAU поддерживает свой собственный формат файлов трасс. Трассы формата TAU могут быть преобразованы в другие формат трасс.

Формат трасс TAU поддерживает наиболее широко используемые типы событий и обеспечивает поддержку библиотеки для записи и чтения. Трассы TAU состоят из небольшого файла определений и потенциально большого файла событий. Трассы параллельных процессов или потоков могут быть сначала записаны в несколько локальных трасс. Позже они объединяются в единую глобальную трассу с помощью инструмента `tau_merge`.

Формат TAU кодирует записи в бинарном виде и явным образом зависит от платформы, в частности очередность байт на платформе байт не корректируется. Все типы записей потребляют некоторое фиксированное количество байт [32]. Это упрощает внутреннее управление, но приводит к определенным накладным расходам хранения и к ограничениям расширяемости.

Нужно отметить, TAU имеет целый ряд инструментов, позволяющих конвертировать трассы в формате TAU в форматы Paraver, SLOG-2, ALOG, EPILOG, OTF, VTF.

Состав и порядок записи трассы

Счётчики производительности существуют во многих современных микропроцессорах. Они могут рассчитывать события, связанные с производительностью аппаратного обеспечения, такие как кэш-промахи, операции с плавающей точкой и т.д., пока программа выполняется на процессоре. Стандарт данных о производительности и API (PAPI) пакет предоставляют единый интерфейс для доступа к этим счетчикам производительности[42]. Для того чтобы использовать эти счетчики, необходимо сначала узнать, какие события PAPI ваша система поддерживает. В PAPI описано более 100 различных стандартных событий. Более подробно для разных архитектур этот список можно посмотреть в [42].

2 Формат трассы в среде моделирования «Диана»

2.1 Цели сбора и анализа трасс в проекте «Диана»

Цель проекта: разработка стенда моделирования бортовых вычислительных комплексов.

Цель сбора и анализа трасс: сбор информации и анализ функционирования частных моделей (ЧМ) и их взаимодействия на основе трассы, собранной в ходе имитационного эксперимента.

По трассе исследуются:

- состояния частных моделей и каналов бортовых интерфейсов (БИ);
- происходящие в частных моделях события;
- взаимосвязи между событиями.

2.2 Описание формата трассы событий

Событие в среде моделирования Диана характеризуется [33]:

- типом;
- временем возникновения;
- местом возникновения;
- набором дополнительных атрибутов, в зависимости от типа события.

Трасса событий представляет собой совокупность нескольких файлов, в каждом из которых содержится последовательность записей, описывающая выполнение имитационного эксперимента (таблица 8). Основной файл, в котором содержатся записи о событиях, - файл с расширением trc.

Таблица 8. Составляющие трассы событий в среде моделирования «Диана».

Составляющие трассы событий		
№	Название	Расширение файлов

1	Основная трасса	*.trc
2	Трасса групп событий	*.gro и *.get
3	Трасса состояний	*.sta

2.2.1 Основная трасса

В основной трассе в хронологическом порядке записаны записи о событиях, произошедших в имитационной модели в ходе эксперимента:

```
struct event_data
{
    t_time EvTime;    // время возникновения события
    t_numb Proc;     // номер компонента
    short Oper;      // номер оператора события
    short SubType;   // подтип события
    char Type;       // тип события
    ...
}
```

Основные параметры события – время возникновения (EvTime) , компонент, в котором произошло событие (Proc) и его тип (Type) . Также у событий некоторых типов могут быть дополнительные параметры, которые также записаны в структуре event_data.

Таким образом, основная трасса имеет следующий формат, описанный в таблице 9.

Таблица 9. Описание основной трассы.

Описание параметров события в Основной трассе *.trc			
Значимость	Поле	Тип данных	Перемен.
Основные	Тип события	char	Type
	Время возникновения события	t_time	EvTime
	Место возникновения события (номер компонента)	t_numb	Proc
дополнительные (зависят от типа события)	Подтип события	short	SubType
	Номер оператора события	short	Oper
	Продолжительность события (Время)		
	другие		

2.2.2 Трасса групп событий

Некоторые события могут быть логически связаны между собой. Одно событие может быть причиной другого события (следствия).

Информация о связях между событиями записана в трассе групп событий, которая состоит из двух файлов с расширениями gro и get. В файле с расширением gro хранится информация о каждой группе событий (таблица 10), а в get файле – информация о каждом событии каждой группы (таблица 11).

Описание *.gro файла

В gro файле последовательно записаны структуры следующего вида:

```
class Group {
    unsigned long long Id;    // идентификатор группы
    short Type;              // тип группы
}
```

```

t_time BeginTime;      // время начала
t_time EndTime;       // время конца
int NEvents;          // количество событий
int Head;             // номер головного события
long FirstEv;        // номер первого события данной группы в get файле
...
};

```

На данный момент в стенде существует только два типа группы, а максимальное количество событий в группе равно 3.

Таблица 10. Описание трассы группы событий *.gro.

Описание параметров события в Трассе групп событий *.gro		
Поле	Тип данных	Перемен.
Идентификатор группы	unsigned long long	Id
Тип группы	short	Type
Время начала	t_time	BeginTime
Время конца	t_time	EndTime
Количество событий	int	Nevents
Номер головного события	int	Head
Номер первого события данной группы в get файле	long	FirstEv
Другие		

Описание *.get файла

В get файле в виде последовательности записей хранится информация о каждом событии в каждой группе. События внутри группы идут подряд, а группы следуют в порядке возрастания поля Id. Информация о событии в группе представлена в виде следующей структуры:

```

class GEvent {
...
long Group;      // идентификатор группы, которой принадлежит событие
long Event;     // TBD
t_time Time;    // время возникновения события
short Comp;     // компонент, в котором произошло событие
short Type;     // тип события
...
}

```

Таблица 11. Описание трассы группы событий *.get.

Описание параметров события в Трассе групп событий *.get		
Поле	Тип данных	Перемен.
Идентификатор группы, которой принадлежит событие	long	Group
TBD	long	Event
Время возникновения события	t_time	Time
Компонент, в котором произошло событие	short	Comp

Тип события	short	Type
Другие		

2.2.3 Трасса состояний

Состояние компонента характеризуется:

- типом
- временем начала
- продолжительностью

Трасса состояний состоит из файла с расширением `sta` и используется для отображения в визуализаторе временной диаграммы состояний, в котором находятся ЧМ в течение эксперимента. Описание файла приведено в таблице 12.

В библиотеке `tracedb` для описания состояния был введен класс `State`, основные поля которого приведены ниже:

```

Class State {
    ...
    Short StType;      // тип состояния
    Short Comp;       // компонент
    T_time StTime;    // время начала состояния
    T_time StDur;     // продолжительность состояния
    Long Attr;        // атрибут состояния
    ...
}

```

Трасса состояний представляет собой последовательность объектов `State` для каждого интервала модельного времени ненулевой продолжительности, в течение которого некоторый компонент находился в одном и том же состоянии, причем порядок записи соответствует закрытию состояний.

Таблица 12. Описание трассы состояний.

Описание параметров события в Трассе состояний *.sta		
Поле	Тип данных	Перемен.
Тип состояния	short	StType
Компонент	short	Comp
Время начала состояния	t_time	StTime
Продолжительность состояния	t_time	StDur
Атрибут состояния	long	Attr
Другие		

3 Формат трассы в проекте V-Ray

3.1 Цели сбора и анализа трасс в проекте V-Ray

Основной целью данного проекта является развитие математической технологии и программных средств, обеспечивающих достижение двух основных свойств параллельных приложений: эффективности и переносимости.

Это включает в себя:

- разработку системы V-Ray - средства анализа и преобразования программ;
- разработку новых технологий, обеспечивающих развитие переносимых параллельных приложений;
- разработку математических технологий оптимизации программ для широкого спектра компьютеров.

3.2 Описание формата трассы событий

Все трассы событий используемые в системе можно разделить на:

- трассы с путями вызовов,
- трассы без путей вызовов,
- индексные файлы событий.

3.2.1 Трассы с путями вызовов

Трасса без путей вызовов содержит информацию только о факте возникновения некоторого события в конкретном месте прикладной программы, но не информацию о контексте возникновения этого события.

Трасса без путей вызовов - это текстовый файл состоящий из последовательности записей о событиях [34]. Запись о событии состоит из двух текстовых строк. Первая строка содержит индекс события. Вторая строка имеет формат:

<TYPE> <SEP> <TIME> [(<SEP> <PARAM>)]

Таблица 13. Описание параметров трассы с путями вызовов.

Формат трассы с путями вызовов	
Параметр	Описание параметра
<TYPE>	- тип события. Система поддерживает 3 основных типа событий: 0 - контрольная точка; 1 - начало операции; 2 - окончание операции;
<SEP>	
<TIME>	- время возникновения события. Это поле содержит действительное значение в формате IEEE936. время может быть представлено в любой линейной шкале.

<code><PARAM></code>	- параметры, семантика которых может меняться в зависимости от семантики события (см. 2)
----------------------------	--

Пример трассы без путей вызовов:

```
111-
1 .3369670D+00
122
2 .3903979D+00
133- 438-
0 .3905109D+00
```

3.2.2 Трассы без путей вызовов

Трасса с путями вызовов, помимо информации о факте возникновения некоторого события, содержит так же информацию о контексте возникновения этого события.

Трасса с путями вызовов - это текстовый файл состоящий из последовательности записей о событиях. Запись о событии состоит из двух текстовых строк. Первая строка содержит информацию о контексте возникновения события в формате:

`<ID> [(<SEP> <ID>)`

где `<ID>` - это индекс события (см. 1.3).

Вторая строка имеет формат:

`<TYPE> '-' <TIME> [('-' <PARAM>)`

Пример трассы с путями вызовов:

```
148- 136- 131- 130- 436-
1 .3369670D+00
148- 136- 137- 139- 231- 143- 441-
2 .3903979D+00
133- 438-
0 .3905109D+00
148- 136- 137- 139- 231- 143- 441-
0 .3906009D+00
```

3.2.3 Индексные файлы событий

Индексный файл событий содержит информацию о месте возникновения событий в прикладной программе. Это текстовый файл, состоящий из последовательности записей о возможных местах возникновения событий. Запись о возможном месте возникновения события располагается на отдельной строке и имеет формат:

**<ID><SEP><NAME>[['.<OPER>']('<CALL>')]
[<SEP><FILE><SEP><STARTPOS><SEP><ENDPOS>]**

Таблица 14. Описание параметров индексного файла.

Формат индексного файла событий	
Параметр	Описание параметра
<ID>	индекс события
<NAME>	имя модуля где возникло событие
<OPER>	номер оператора в модуле
<CALL>	имя вызываемой функции
<FILE>	имя исходного файла
<STARTPOS>	стартовая позиция оператора в исходном файле
<ENDPOS>	конечная позиция оператора в исходном файле

Пример индексного файла событий:

```
1 TRFD
2 INTGRL
3 OLDA
4 DFINDX
5 TRFD.75(MPI_SEND) TRFD.F 3265 3368
6 TRFD.86(MPI_RECV) TRFD.F 3641 3756
7 OLDA.73(MPI_SEND) TRFD.F 9494 9624
8 OLDA.107(MPI_RECV) TRFD.F 10462 10584
```

3.3 Форматы проектных файлов VD-Ray

В системе используются три основных формата проектных файлов, описанных в таблице 15 [34].

Таблица 15. Проектные файлы.

Проектные файлы VD-Ray	
Название	Расширение
Проектные файлы для трасс с путями вызовов	*.tv
Проектные файлы для трасс без путей вызовов	*.tvp
Проектные файлы для сравнения трасс	*.tvc

Все проектные файлы имеют общую структуру, и различие в формате трасс определяется только по расширению.

Общая структура проектного файла:

index

trace1

trace2

trace3

...

traceN

param1=val1

param2=val2

...

paramM=valM

Где index - относительный путь индексного файла, traceK - относительный путь K-го файла трассы. paramJ - имя J-го параметра, valJ - значение J-го параметра. Значением параметра может быть либо скаляр, либо массив. Скалярные значения указываются без дополнительного форматирования (как строки так и числовые значения). Массивы разделяются запятыми и заключаются в фигурные скобки.

Параметры в проектных файлах:

NR_prefix - массив префиксов функций, которые считаются накладными расходами;

nTypes - число различных типов функций (для выделения цветом);

TypeK - массив имен функций K-го типа;

GlobFunc - массив имен функций, которые считаются глобальными;

Sends - массив имен функций, которые считаются функциями передачи сообщений;

Recvs - массив имен функций, которые считаются функциями приема сообщений;

toProcessIndex - индекс параметра содержащего номер процесса-опонента в операциях взаимодействия;

TagIndex - индекс параметра содержащего тег в операциях взаимодействия;

AsyncIDIndex - индекс параметра содержащего флаг асинхронной передачи данных;

Пример проектного файла (trace.tvp):

name.lst

trace4.0

trace4.1

trace4.2

trace4.3

NR_prefix={MPI*}

nTypes=4

Type1={MPI_SEND,MPI_ISEND}

Type2={MPI_RECV}

Type3={MPI_BARRIER}

Type4={MPI_PROBE}

GlobFunc={MPI_BARRIER,MPI_ALLREDUCE}

Sends={MPI_SEND,MPI_ISEND}

Recvs={MPI_RECV}

toProcessIndex=0

TagIndex=-1

AsyncIDIndex=-1

4 Формат трассы Rajé

4.1 Цели сбора и анализа трасс в проекте Rajé

Цель проекта: разработка Rajé trace file format и интерактивного и масштабируемого инструмента Rajé для визуализации этого формата.

Цель сбора и анализа трасс:

- мониторинг производительности параллельных приложений
- мониторинг работы процессоров в огромных масштабируемых PC кластерах
- отображение поведения распределенных приложений (Java)

4.2 Описание формата трассы

Визуализация Rajé формируется из объектов, организованных в соответствии с иерархией древовидного типа, в которой вершины называются контейнерами, а листья объектами (или сущностями) [35].

The Rajé data format – самоопределяемый, хотя он не полностью SDDF формат, используемый Pablo.

Существует «мета-формат»:

- Формат инструкций, определяющий контейнеры и объекты.
- Формат событий, записанных во время выполнения параллельных программ.

Эти определения обычно вставляются в файл трассы. Они могут быть даже вставлены в исходные файлы наблюдаемой программы, что обеспечивает использование записей событий этой программы, собираемых новым определением событий, определенных пользователем.

Используя эти определения, это дает возможность определять иерархию контейнеров и объектов, адаптированных для данной модели программирования и языка. Определения типов иерархий так же как и инструкций и форматов событий составляют конкретизацию понятия “обобщенный” инструмент визуализации Rajé: это может быть использовано для визуализации распределенных приложений, написанных на Java или для системного мониторинга огромных кластеров.

Событие состоит из полей, а каждое поле включает три составляющих:

<имя поля> <тип поля> <значение поля>

пример:

Field Name	Field Type	Field Value
EventName	string	SendMessage
Time	timestamp	3.233222
ProcessId	integer	5
Receiver	integer	3
Size	integer	320

Field Name	Field Type	Field Value
EventName	string	UnblockProcess
Time	timestamp	5.123002
ProcessId	integer	5
FileName	string	sync.c
LineNumber	integer	98

```
%EventDef SendMessage 21      %EventDef UnblockProcess 17
%   Time          date        %   Time          date
%   ProcessId     int         %   ProcessId     int
%   Receiver      int         %   LineNumber   int
%   Size          int         %   FileName     string
%EndEventDef                %EndEventDef
```

Figure 3.2: Examples of event definitions

```
21 3.233222 5 3 320
17 5.123002 5 98 sync.c
```

Figure 3.3: Examples of events

Рис. 21. Пример определений событий и событий в трассе[35]

Типичный файл трассы содержит тысячи событий десяти различных типов. Файл состоит из двух частей: определение событий в начале файла и далее сами события.

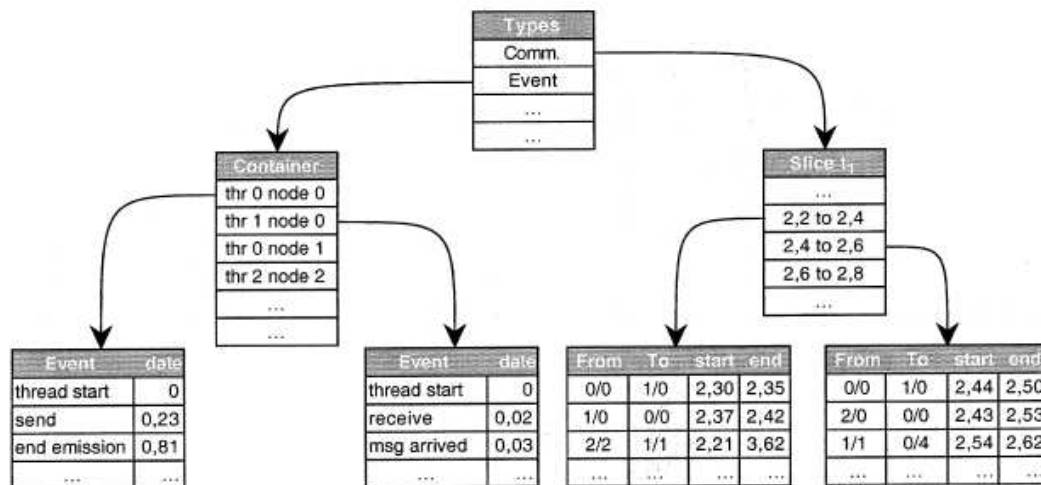


Fig. 9. Structure of an observation window. Types and containers are stored in hash tables. Instantaneous objects are sorted by date. Non-instantaneous objects are grouped in time slices, depending on their initial date. Within each time slice, they are sorted by termination date.

Рис. 22. Структура окна наблюдения [35].

5 Формат трассы EPILOG

EPILOG – бинарный формат трасс событий и библиотека времени (этапа) выполнения для генерации трассы событий MPI и OpenMP приложений.

EARL (Event Analysis and Recognition Library – Библиотека анализа и распознавания событий) [38].

EARL – универсальный интерфейс высокого уровня для чтения трасс событий в формате EPILOG. EARL обеспечивает произвольный доступ к отдельным событиям и вычисляет состояние выполнения на момент данного события, а также predefined отношения между парами соответствующих событий. EARL может быть использован для большого числа различных задач анализа трасс. Реализован на C++.

5.1 Цели сбора и анализа трасс

Цель сбора и анализа трасс: анализ производительности MPI и OpenMP приложений.

5.2 Описание формата трассы

Формат трасс Epilog был разработан Forschungszentrum Jülich GmbH, Germany и University of Tennessee, Knoxville, USA. Он является частью проекта Kojak и доступен как открытый под лицензией BSD license [31].

Формат EpiLog использует бинарное кодирование. Он поддерживает все обычные типы определений и записей событий. Существуют специальные типы записей для отслеживания событий, связанных с OpenMP многопоточностью. Он поддерживает все обычные определения и типы записи по событию. Эпилог также использует фиксированные спецификации местом для физического и логического размещения событий с набором из 4 элементов (машина, узел, процесс, поток).

Другой особенностью является поддержка Эпилога для явной синхронизации времени. Если вычислительные узлы или процессы используют различные локальные таймеры, то для синхронизации записей можно задать локально-глобальное преобразование времени в определенных точках во время трассировки. На основании такой информации можно осуществлять глобальное регулирование времени как шаг пост-обработки.

Эпилог имеет специальные типы записей, чтобы отметить определенные события, порождающие самотрассировку, например, деактивировать и повторно активации трассировки или случайного ввода / вывода, вызванных трассировкой подсистемы.

Файл трассы в формате EPILOG состоит из заголовка, за которым следуют последовательность записей.



Рис 23. Структура файла трассы в формате EPILOG [38]

Заголовок состоит строки " EPILOG ", а затем два байта, содержащие наименьшую и наивысшую версии EPILOG, затем другой байт, указывающий на то, какие арифметические типы будут использоваться.

Структура каждой записи имеет вид:

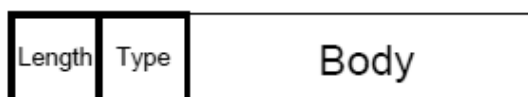


Рис 24. Структура записи в формате трассы EPILOG [38].

Запись события:



Рис 25. Структура записи события в формате трассы EPILOG [38].

6 Формат трассы *LOG

6.1 Цели сбора и анализа трасс

Цель сбора и анализа трасс: SLOG - Масштабируемый формат лог-файлов для трассировки данных параллельных программ [39].

Jumpshot trace formats (ALOG, BLOG, CLOG, SLOG)

Существует ряд форматов трасс ALOG, BLOG, CLOG, SLOG, связанных с инструментами визуализации из семейства Jumpshot. В отличие от всех других инструментов, Jumpshot использует терминологию ведения журнала событий (event logging) вместо трассировки событий (event tracing) [31].

Различные форматы трасс развивались вместе со средствами визуализации. Все они используют бинарное кодирование и состоят из одного файла. Во-первых, был формат ALOG и его преемник BLOG. Они используют кодирование фиксированных записей, состоящих из шести целых значений плюс текстовая строка из 12 символов. В следующем формате CLOG устранено кодирование фиксированной длины для того чтобы позволить более гибкие расширения и дополнения типов записей. Эти три формата – типичные форматы, основанные на событиях в Jumpshot терминологии, то есть они используют события трассы, как наименьшие сущности (объекты) данных.

В формате SLOG-1 отмечается переход к так называемому формату, основанному на состояниях. Теперь, наименьшими элементами данных являются состояния, которые имеют начало и конец, соответствующие определенным временным меткам.

Состояние соответствует двум входному и выходному событиям в формате, основанном на событиях, с двумя временными отметками. Состояния организованы в так называемых кадрах (фреймах), которые содержат все состояния последовательных временных интервалов, которые не пересекаются. Такие кадры могут быть загружены отдельно, в целях поддержки загрузки по требованию. В частности, эта схема требует дублирования пограничных состояний, которые разделены между соседними кадрами на шаге пост-обработки. Это замечательная особенность для формата трассы: это уже не более или менее пассивной контейнер для чистых данных измерений. Вместо этого, она становится активной частью инструмента визуализации, а также модифицирует содержимое для этой цели.

Последний SLOG-2 формат идет еще дальше в направлении SLOG-1. Это так называемый формат, основанный на холсте (drawable-based format), что означает, что теперь графические объекты используются как наименьшие элементы. Вместо прежней структуры кадра, записи (графические объекты) организованы в иерархии ограничивающих

прямоугольников (фактически ограничивающие интервалы) за время измерения. Это не требует больше дублирования границ элементов. Кроме того, они включают в себя краткую информацию в верхней части иерархии, с тем, чтобы обеспечить быстрый просмотр.

6.2 Описание формата трассы

SLOG-2 позволяет собирать данные в логические группы, а не формирование их как поток записей. Структура SLOG-файла представлена на рис. 26.

Приведенный рисунок описывает файл SLOG2 [41]. Обратите внимание, что SLOG2 не определяет формат файла, это блок-схема показывает одну из возможных организаций, чтобы проиллюстрировать, что такое содержимое файла и структура позволяют создавать SLOG2 файл в один проход.

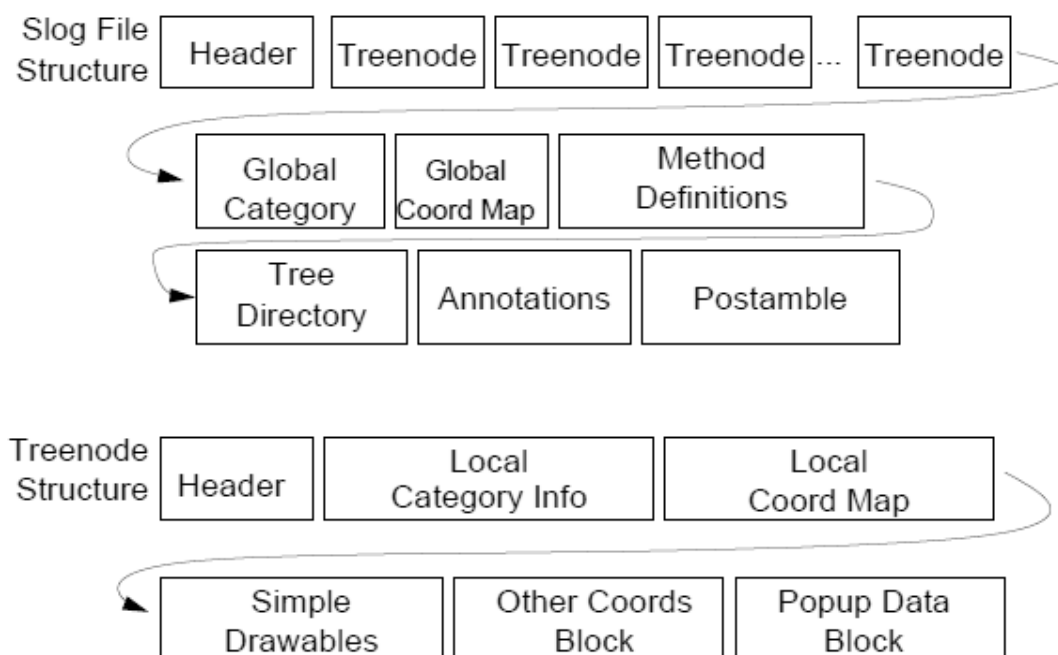


Рис 26. Структура файла трассы в формате SLOG [41].

Структура Slog файла

Header - содержит информацию о версии SLOG, имя программы и пользователя и другую информацию о файле.

Treenode – блок данных, соответствующий R_j (узел).

global category info – информация об определениях категорий, используемых в одной или более вершин дерева(Treenode).

global coord map – информация об отображении координат для координат в одной или более вершин(Treenode).

method definition - информация о методах, которые должны быть использованы для всплывающих данных. Это предназначено для будущего усовершенствования интерфейса SLOG; методы (или, точнее, методы, отличные от стандартных методов) не определенные в данной версии API.

tree directory – блок смещений в начале каждой вершины, наряду с началом и окончанием каждого узла дерева. Смещения – это 8-байтовые целые числа в байтах относительно начала файла.

annotations - Блок текста, записей для просмотра истории и предоставленные пользователем аннотации о SLOG2 файл. Этот блок в конце файла (за исключением postamble), так что аннотации могут добавляться в большие файлы без перезаписи файла.

postamble - содержит пять 4-байтовых целых чисел, указывающих расположение (по отношению к концу файла) из global category info, global coord map, method definition, treedirectory и аннотации блоков. 4 целых достаточно здесь, потому что эти объекты являются относительно небольшими. Тем не менее, мы могли бы использовать 8-байтовые целые числа для согласования с treedirectory смещениями.

7 Формат трассы CCG

CCG – использует альтернативный линейному способу хранения данных – способ, основанный на полном дереве вызовов функций. Полное дерево вызовов (CCT) содержит всеобъемлющую историю вызовов функций в процессе [40]. Более того, не только функции вызывают события, но могут содержаться и все другие виды событий такие как события отправки/получения сообщения, события ввода/вывода или события от аппаратных счетчиков производительности. Однако иерархия вызовов функций определяет структуру CCT.

7.1 Описание формата ССГ

Пример ССТ на рис.27 слева внизу. Один процесс P0 вызывает функцию main(). Функция main() вызывает один раз функцию foo(), а функция foo() в свою очередь дважды вызывает функцию bar(). Функции bar() вызывают по три раза функцию MPI_Send().

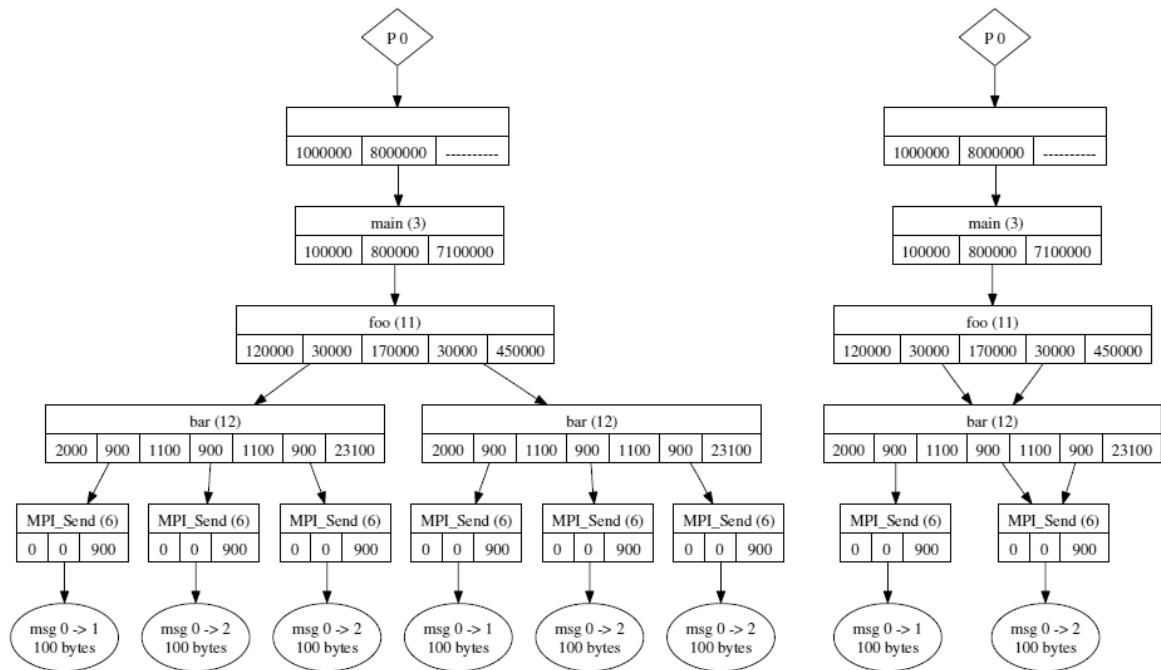


Рис 27. Преобразование ССТ в ССГ [31].

Можно произвести сжатие дерева справа в дерево слева. Левое дерево после этого преобразования и называется ССГ. На примере из дерева с 17 вершинами получили дерево с 8 вершинами. В клетках указывается время выполнения данной функции.

Операции:

- произвольный доступ к позициям в ССГ.
- линейный обход по позициям.
- поиск по временной метке.

Операции для диапазонов позиций:

- статистические сводки.
- представление хронологии событий.
- приемо-передающие соответствия.

Достоинства:

- возможность интерактивного анализа производительности и визуализации
- возможность регулировать степень сжатия трасс (уменьшение количества вершин, снижение расхода памяти и вычислительных ресурсов)
- есть алгоритмы для всех общих задач анализа
- доступ к сжатым данным не требует декомпрессии.
- Возможность анализировать большие трассы

Недостатки:

- нужно целиком реализовывать самостоятельно в соответствии с описанием, приведенным в [31] и алгоритмами работы с ним.

8 Формат трасс STF

Формат STF (The Structured Trace Format) был разработан Pallas GmbH совместно с Center for High Performance Computing (ZHR), TU Dresden в 2001 году. После того как корпорация Intel приобрела Pallas GmbH, доступность формата STF была ограничена платформами Intel. Текущая версия формата STF является частью набора инструментов Intel Trace Collector(ITC) и Intel Trace Analyzer(ITA), соответственно его описание и API являются закрытыми. Поэтому в данном обзоре формат STF подробно рассматриваться не будет[31].

8.1 Описание формата трассы

Трасса включает в себя набор файлов следующих типов:

- Глобальный индексный файл, ссылающийся на все другие файлы.
- Файл глобальных определений, содержащий определения записей.
- Так называемый файл кадров, который содержит общую информацию и специальную краткую информацию.
- Файл статистики со статистикой обо всех типах записей событий.
- n файлов с данными о событиях, содержащих фактические записи событий.
- n закрепленных файлов с индексом и информацией об истории файлов с данными о событиях.
- один файл для всех записей сообщений типа “точка-точка”.
- один файл для записей коллективной связи.

9 Формат трассы OTF

Открытый формат трасс (OTF) – это новый формат трассы для использования в крупномасштабных параллельных платформах. OTF преследует три цели:

- открытость (доступен под лицензией BSD open source license),
- гибкость,
- производительность.

Особенности формата:

1. Быстрый и эффективный последовательный и параллельный доступ
2. Независимость от платформы
3. Селективный доступ к:
 - Процессам
 - Временным интервалам
4. API / Интерфейсы
 - Считывание / запись полной трассы с поддержкой нескольких файлов
 - Поддержка фильтрации и параллельного ввода / вывода
 - Низкий уровень интерфейса для библиотек трасс

9.1 Цели сбора и анализа трасс

Цель проекта: Данный формат OTF создавался специально для поддержки разработки масштабируемых средств трассировки производительности для IBM BG / L машины [36]. Формат направлен на приложения, написанные на Fortran77, Fortran (90/95/etc.), C, and C++.

Цель сбора и анализа трасс: анализ производительности приложений на IBM BG / L.

9.2 Описание формата трассы событий OTF

OTF использует специальные ASCII представления данных для кодирования элементов данных.

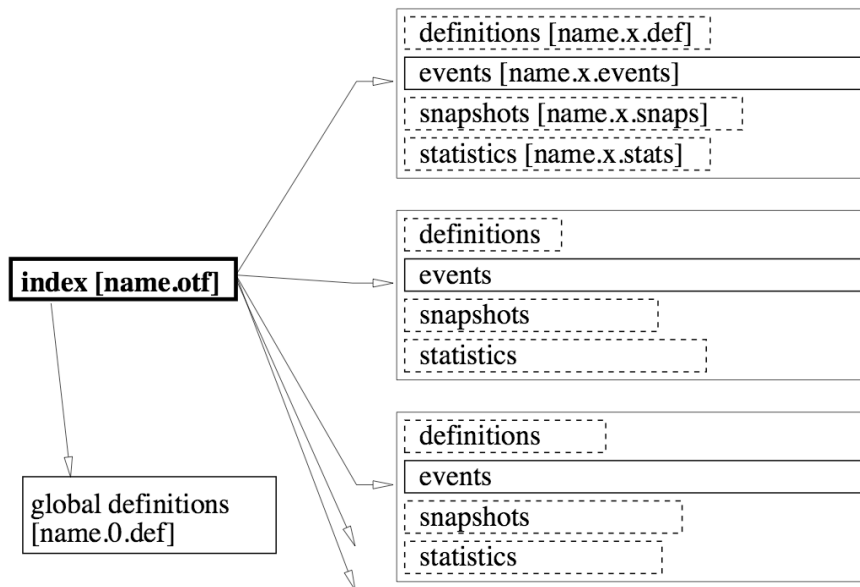


Рис 28. Структура трассы формата OTF [37]

Содержание OTF[37]:

1. Определение записей
 - Карта идентификаторов событий, интервалы (начало / конец) и имена событий
 - Символы атомных событий
 - Группы процессов
2. Производительность событий
 - Датируемые события для входных и выходных состояний
 - Датируемый счетчик событий (монотонно возрастающий или нет)
3. Глобальный основной файл
 - Отображение процессов в потоки
4. Статистика
 - Обзор по всем интервалам времени
5. Краткая характеристика
 - Стека вызовов, списка отложенных сообщений и т.д. в некоторый момент времени.

В целях поддержки быстрого и селективного доступа к большим объемам данных трассы, OTF основана на потоковой модели, т. е. единственные отдельные блоки, представляющие сегменты общих данных. OTF потоки могут содержать несколько независимых процессов в то время как процесс принадлежит исключительно единственному потоку.

OTF организует запись трассы в несколько потоков, т.е. отдельные фрагменты могут быть доступны независимо (см рис.28), таким образом достигается селективный доступ и параллельный ввод/вывод. Поток содержит события одного или более процессов в одном файле во временном порядке до тех пор, пока всякий процесс трассы отображается на одном потоке исключительно [31].

Каждый поток представляет собой несколько файлов, которые хранят определения записей, события о производительности, информацию о состоянии, и резюме событий в отдельности. Единый глобальный мастер-файл содержит информацию, необходимую для отображения процесса на поток.

Имена частей файлов трассировки следуют строгому правилу именования. Каждое имя файла начинается с произвольного общего префикса, который может быть определен пользователем. За ним следуют маркер идентификатор, который используется для внутренних целей (процесс отображения) и суффикс в соответствии с типом файла. OTF файлы не предназначены для доступа напрямую, а только через API библиотеки OTF's. Это строгое требование для того, чтобы гарантировать дальнейшую совместимость.

Таблица 16. Описание файлов составляющих трассу формата OTF.

Файлы, составляющие трассы OTF			
№	Значимость	Описание	Структура имени
1	Обязательны	Мастер-файл	<name>.otf
2		Глобальный файл определений	<name >.0.def
3		Файл событий	<name>.x.events
4	Опциональны	Файл локальных определений	<name>.x.defs
5		Файл снимков (snapshots)	<name>.x.snaps
6		Файл статистики	<name>.x.stats

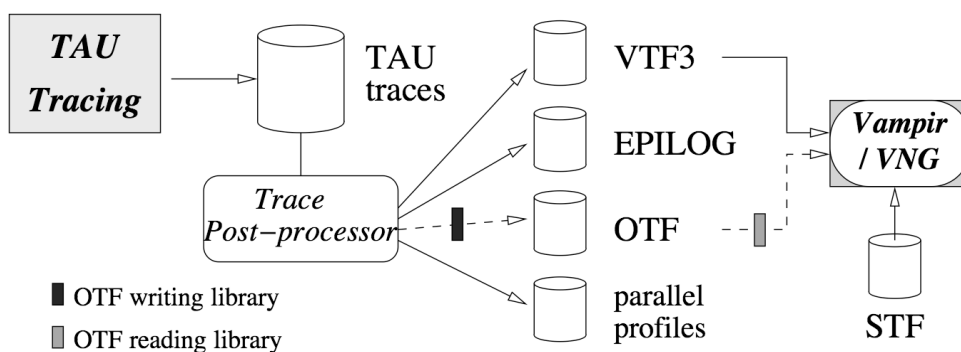


Рис 29. Средства генерации и анализа трасс OTF формата [36].

Файл формата OTF может быть сгенерирован с помощью библиотеки VampirTrace в автоматическом режиме. Чтение и запись в трассу формата OTF осуществляется через API чтения и записи [36].

Г.4 Средства анализа трасс событий

«Визуализатор временной диаграммы»

Внешний вид представлен на рис. 29.

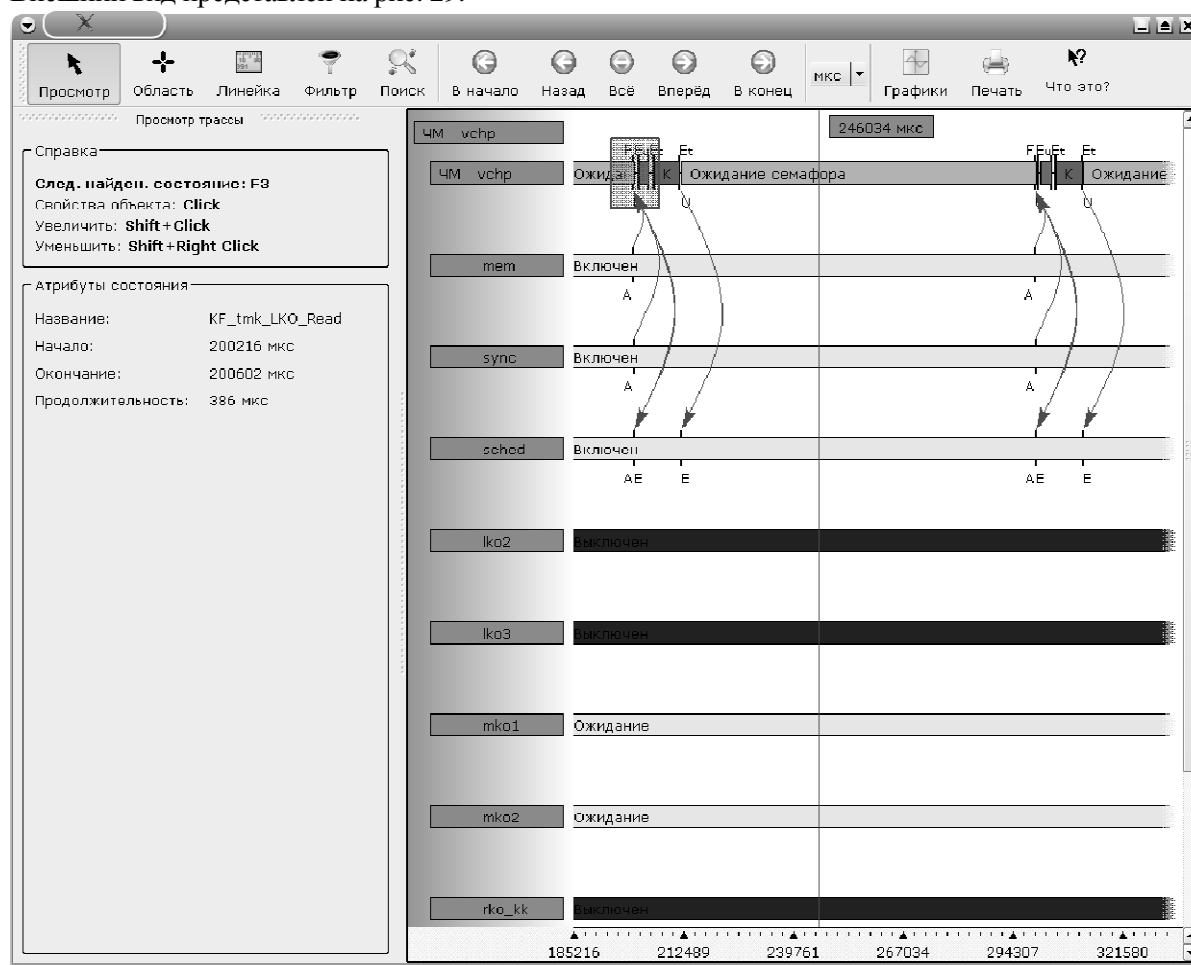


Рис 29. Окно временной диаграммы.

Назначение визуализатора: Он предназначен для отображения информации о функционировании частных моделей и их взаимодействии в виде временной диаграммы. На диаграмме отображаются: состояния частных моделей и каналов бортовых интерфейсов; происходящие в частных моделях события; взаимосвязи между событиями.

Визуализатор временной диаграммы позволяет:

- Просматривать трассу (сведения о трассе: точки начала и конца диапазона времени эксперимента, количество отображаемых на временной диаграмме (ВД) и

неотображаемых (скрытых) компонентов стенда, количество отображаемых типов событий).

- Отображение области по значениям модельного времени.
- Получить информацию о состоянии выделенного компонента (имя состояния, время начала и длительность).
- Просмотреть список всех событий, находящихся на некоторой ЛЖ в некоторый момент времени и в окрестности этого момента времени.
- Выбирать типы отображаемых событий для уменьшения объема информации, представленной на ВД. Выбирать отображение компонентов и состояний.
- В отдельном окне отображать графики зависимости параметров моделей от времени.
- Изменять масштаб временной диаграммы.
- Просматривать атрибуты событий и состояний.
- Осуществлять поиск событий и состояний.
- Выводить на печать ВД.

Средство Rajé

Инструмент визуализации Rajé позволяет программисту визуализировать выполнения параллельных программ, использующих потенциально большое число взаимодействующих потоков (нитей, легковесных процессов), появляющихся динамически. Визуализация выполнения – это важный инструмент, помогающий настраивать приложения, использующие модели параллельного программирования.

Визуализация большого числа потоков увеличивает количество проблем с копированием в связи с нехваткой рабочего пространства на экране для того, чтобы визуализировать их и понять. Графические дисплеи большинства существующих инструментов визуализации для параллельных программ показывают активность фиксированного числа узлов и межузловых коммуникаций. Это только возможность представлять активность отдельного потока управления на каждом из узлов. Это непременно есть возможность использовать эти системы для визуализации активности многопоточных узлов, представляя каждый поток как узел. В этом случае количество потоков должно быть четко ограничено и не должно быть различным во время выполнения программы. Эти инструменты визуализации не адаптированы к визуализации потоков, количество которых постоянно варьируется и время их жизни зачастую очень короткое. В итоге, эти инструменты не поддерживают визуализацию синхронизацию локальных потоков, использующих мьютексы и семафоры.

Некоторые инструменты были созданы для отображения многопоточных программ [14, 27]. Однако они поддерживали модель программирования, затрагивающую единственный уровень параллелизма без узлов, эти узлы были с общей разделяемой памятью (и с несколькими центральными процессорами). Наши программы выполняются на нескольких узлах: внутри таких узлов потоки взаимодействуют, используя примитивы синхронизации, однако потоки, выполняющиеся на разных узлах, взаимодействуют посредством обмена сообщениями. Более того, по сравнению с этими системами, Rajé следует представлять гораздо большее число объектов.

Самой инновационной чертой Rajé является комбинирование характеристик согласованности действий и масштабируемости с расширяемостью. По сравнению с инструментами пассивной визуализации, где объекты параллельных программ – коммуникации, изменения в состояниях процессоров и др. - отображаются как только они порождены и не могут быть запрашиваемыми, т.е. это возможность рассмотреть все объекты изображенные на текущем экране и переместиться назад во времени, отображая прошлые объекты снова. Масштабируемость – свойство справляться с огромным числом потоков. Расширяемость – важная характеристика инструмента визуализации справляться с эволюцией интерфейсов параллельного программирования и методами визуализации. Расширяемость дает возможность расширить среду с новой функциональностью: обработка новых типов трасс, добавление новых графических экранов, визуализация новых моделей программирования и т.д.

Характеристики интерактивности и масштабируемости Rajé были описаны в предыдущих статьях. Эта статья фокусируется на характеристиках расширяемости: модульное проектирование с легким добавлением новых модулей, семантическая независимость модулей, которая позволяет им быть использованными в большом количестве различных контекстов и особенно универсальность имитатора компонентов Rajé, которые дают приложениям возможность определять, что они хотят визуализировать и как это должно быть сделано.

Функциональность Rajé:

1. ATHAPASCAN: Модель параллельного программирования, основанная на потоках
2. Трассировка параллельных программ
3. Визуализация потоков
4. Взаимодействие
5. Масштабируемость: фильтрация информации и возможности изменения масштаба

Vampir, VampirTrace

Vampir – набор инструментов для анализа производительности высокопроизводительных параллельных приложений. В набор входят:

1. VampirTrace – инструмент для измерения
2. Средства визуализации Vampir
3. VampirServer

VampirTrace изначально порожден из библиотеки KOJAK для работы с трассами и был доступен как свободное программное обеспечение под лицензией BSD license.

VampirTrace – средство для сбора трасс в открытом OTF формате.

URL	http://www.pallas.de/pages/vampir.htm
Место разработки	Коммерческий продукт, разработка компании <u>Pallas</u> (Германия).
Версии	VAMPIR 4.0 (X Window), VAMPIRtrace 4.0
Тип	Тип А (трассировка + визуализация). VampirTrace - система генерации трасс (A1), Vampir - система визуализации (A2).
Языки/библиотеки	Языки - Fortran, C; передача сообщений в рамках MPI.
Платформы	<ul style="list-style-type: none">• <u>Cray T3D/T3E</u>• DEC Alpha (OSF/1)• Fujitsu VP 300/700• Hitachi SR2201• <u>HP 9000</u>• <u>IBM RS/6000, SP</u>• Intel Paragon• NEC SX-4• <u>SGI Origin, PowerChallenge (IRIX 6)</u>• Sun SPARC• Intel x86 (Solaris 2.5)
Функциональность трассировки.	Сбор трасс. Линковка с VampirTrace - прослойкой между MPI и пользовательской программой. Уровни детализации. Слабые возможности настройки уровня детализации - только по подпрограммам. Возможна установка точек начала/конца трассировки. Тип трассировки. Только события (статистика собирается на этапе анализа трасс).
Визуализация	Процессы - параллельные линии, события - точки на них. Взаимодействия. Связь линий процессов, матрицы объемов и количества пересылок Другие объекты. Круговые диаграммы и статистические гистограммы. Поддерживается связь с исходным кодом.
Статистика	Суммарное время по замеряемым инструкциям или типам инструкций и количеству срабатываний; отображается на круговых диаграммах и гистограммах.

AIMS - Automated Instrumentation and Monitoring System

Место разработки:	Некоммерческий продукт, разрабатывается в NASA Ames Research Center в рамках программы High Performance Computing and Communication Program.
Тип	Тип А (трассировка + визуализация)
Языки/Библиотеки	Fortran 77, HPF, C. Библиотеки передачи сообщений: MPI,PVM,NX.
Платформы	IBM RS/6000 SP, рабочие станции Sun и SGI, Cray T3D/T3E.
Функциональность трассировки	<p>Сбор трасс. Автоматизированное изменение исходного кода программы путем вставки специальных вызовов. Параллельно со сбором трассы создается файл со статической информацией.</p> <p>Уровни детализации. Подпрограммы, вызовы процедур, процедуры различного типа (процедуры ввода-вывода, MPI процедуры т.п.)</p> <p>Формат трасс. Формат описан в[7]. Ориентирован на передачу сообщений.</p> <p>Тип трассировки. События, статистика (может собираться без полной трассы).</p>
Визуализация	<p>Процессы - параллельные линии. События изображаются точками на этих линиях. Особым образом изображаются накладные расходы: времена ожидания, блокировка. Есть возможность "проигрывания" трасс.</p> <p>Время - реальное (астрономическое)</p> <p>Связь линий процессов линиями, обозначающими взаимодействия (передача сообщений, глобальные операции).</p> <p>Диаграммы взаимодействия процессов, временные срезы, история вызовов и трассируемых блоков.</p> <p>Поддерживается связь с исходным кодом.</p>
Статистика	Суммарное время по замеряемым инструкциям или типам инструкций и количество срабатываний.

Jumpshot

URL	http://www-unix.mcs.anl.gov/mpi/www/www1/Jumpshot.html
Где разрабатывается?	Некоммерческое средство, разработано в Аргоннской национальной лаборатории. Распространяется вместе с пакетом <u>MPICH</u> .
Версия	Jumpshot 1.0 (требуется Java 1.1 или выше)
Тип	A2 (визуализация трасс)
Языки/библиотеки	Передача сообщений: MPI.
Платформа	Сбор трасс - любые платформы, где работает MPICH. Визуализация - Java.
Функциональность трассировки	Сбор трасс. Для получения трассы программу необходимо откомпилировать с профилировочной версией библиотеки MPICH. Формат трасс. *LOG. Тип трасс. События
Визуализация	Процессы - параллельные линии, цветом изображается тип функции. Взаимодействия. Связь линий процессов. Другие объекты. Объемы пересылок по времени, гистограммы накладных расходов по времени.
Статистика	Суммарные времена работы различных типов процедур.
Разное	jumpshot входит в состав MPICH начиная с <u>версии 1.1.1</u> и заменяет собой Tcl/Tk-программы upshot/nupshot, входившие в состав MPICH более ранних версий.

Pablo Performance Analysis Toolkit Software

Пакет состоит из набора средств:

- SvPablo - визуализатор статистической информации (X Window).
- SDDF - библиотека для записи трасс и набор средств для работы с SDDF файлами
- Trace Library and Extensions - библиотека для трассировки
- I/O Analysis - статистика операций ввода-вывода
- MPI I/O Analysis - статистика MPI I/O
- HDF (Hierarchical Data Format) Analysis - анализ использования HDF операций
- Analysis GUI - библиотека средств для просмотра SDDF трасс
- IO Benchmarks - сбор трасс операций ввода-вывода

URL	http://vibes.cs.uiuc.edu/Software/Pablo/pablo.htm
Где разрабатывается?	Некоммерческий пакет, разработан в университете шт. Иллинойс.
Языки/библиотеки	ANSI C, Fortran 77, Fortran 90 (с ограничениями), HPF (Portland Group).
Платформы	<ul style="list-style-type: none"> • SvPablo - SunOS 5.6, SGI Irix 6.5 • Trace Library and Extensions - Sun SunOS, Sun Solaris, RS6000, SP2, Intel Paragon, Convex Exemplar, SGI IRIX • I/O Analysis - Sun Solaris, SGI IRIX • MPI I/O Analysis - Sun SunOS, SGI IRIX • HDF Analysis - Sun Solaris, SGI IRIX • Analysis GUI - Sun Solaris (X11R5+Motif) • IO Benchmarks - Sun Solaris, SGI IRIX, Intel Paragon
Функциональность трассировки.	<p>Уровни детализации. На уровне интерфейсов, можно делать ручную разметку с использованием svPablo.</p> <p>Формат трасс - SDDF Тип трасс. Статистика, события.</p>
Визуализация	<p>SvPablo. Основа визуализации - связь с исходным кодом. Представляет цветом число вызовов и общее время фрагмента.</p> <p>Analysis GUI. Библиотека подпрограмм для визуализации трасс в формате SDDF</p>
Статистика	<p>Развернутые средства статистики, в виде набора пакетов.</p> <ul style="list-style-type: none"> • I/O Analysis: анализ операций ввода-вывода • MPI I/O Analysis: анализ ввода-вывода MPI функций • HDF Analysis: анализ операций HDF.
Совместимость	Есть конверторы из разных форматов в SDDF – IBM VT Trace, AIMS.
Развитие	Поддержка HPF, Fortran 90. Поддержка MPI 2.0.

Paradyn

URL	http://www.cs.wisc.edu/paradyn
Где разрабатывается?	Некоммерческое средство, разрабатывается в University of Wisconsin,

Версия	4.0
Тип	В (онлайн-анализ)
Языки/библиотеки	Fortran, Fortran 90, C, C++: MPI, PVM; HPF
Платформы	<ul style="list-style-type: none"> • Sun SPARC (только PVM) • Windows NT на x86 • IBM RS/6000 (AIX 4.1 или старше)
Функциональность трассировки	Динамическая настраиваемая инструментовка программ во время выполнения. В код программы во время ее выполнения динамической вставляются и убираются вызовы трассирующих процедур. Все делается автоматически, в результате значительно уменьшаются накладные расходы. Начинает с крупных блоков, затем постепенно детализирует узкие места (для этого программа должна достаточно долго работать)
Визуализация	<p>В основе визуализации лежат два вектора</p> <ul style="list-style-type: none"> • измеряемые параметры производительности: процессорное время, различные накладные расходы, ожидания, времена пересылок и ввода-вывода и т.д. • компоненты программы/вычислительной системы, к которым относятся параметры: процедуры, процессоры, диски, каналы передачи сообщений, барьеры и т.д. <p>На этих векторах образуется матрица: ее элементы либо скаляр (значение, среднее, минимум, максимум и т.д.), либо временная диаграмма (история изменения характеристики).</p> <p>Все характеристики отображаются во время исполнения программы.</p>
Проблемы	Есть проблемы с масштабируемостью. На программе при малом числе процессоров (меньше 12) все выглядело нормально, а на большем числе процессоров - более чем 80% увеличение времени. Так же сейчас самой системой занимается очень много памяти.
Развитие	Устранение проблем масштабируемости, уменьшение требуемой памяти, поддержка других платформ.

CXperf

URL	HP Performance Analysis Tools - http://www.hp.com/esy/lang/tools/Performance/CXperf User's Guide
Где разрабатывается?	Коммерческое средство, разработка <u>Hewlett-Packard</u> .
Тип	A (трассировка + визуализация)
Языки/библиотеки	HP ANSI C (с89), ANSI C++ (aCC), Fortran 90 (f90), HP Parallel 32-bit Fortran 77
Платформы	Сервера HP на базе PA-RISC
Функциональность трассировки	Сбор и настройка трасс осуществляется с помощью указания специальных профилировочных опций компилятора.
Визуализация	3D-визуализация, связь с кодом программы, масштабирование, сопоставительный анализ, графы вызовов.

Intel Trace Analyzer and Collector(ИТАС)

Коммерческий продукт

Intel Trace Analyzer and Collector(ИТАС) - средство для поиска ошибок и анализа эффективности MPI приложений на основе построения и последующего анализа, визуализации трассы событий (MPI вызовов, пересылок сообщений). ИТАС состоит из двух основных частей. Trace Collector и Trace Analyzera, также ряда вспомогательных утилит.

Intel Trace Analyzer - средство графической визуализации и анализа трасс событий MPI приложений, собранные частью называемой Intel Trace Collector.

Представление данных в виде различных диаграмм, отображающих один и тот же участок трассы с различных качественных представлений. В совокупности с гибкой системой настройки визуализации выводимой информации помогает лучше проанализировать взаимодействия и события, происходящие при работе приложения. Все это помогает выявлять узкие места, строить сравнительные графики и находить возникающие ошибки.

С помощью пакета можно собирать трассы исполняемых файлов, получать статистическую информацию о выполненном приложении или только о его части. Есть возможность осуществлять профилировку функций, а также работать с информацией "дебага".

Компоненты ИТАС

ИТАС состоит из двух основных частей. Часть, отвечающая за сбор трассы и опции работы с приложением, называется Intel Trace Collector. Она устанавливается на кластерную часть системы: Linux или Windows Cluster/Server. Отвечает за сбор трассы, формирование файла трассы, формат и работу рассматриваемого приложения.

Intel Trace Analyzer - визуализирующая часть продукта. Она работает уже с файлами трасс, представляет информацию в них в наглядном виде, понятном пользователю. Имеет

много гибких настроек для вывода диаграмм, графов вызовов, и сбора статистики. Важной отличительной чертой Intel Trace Analyzer является то что статистика автоматически пересчитывается при изменении масштаба рассматриваемого участка.

Также в пакет ITAC входит ряд вспомогательных утилит которые облегчают процесс сборки и обработки собранных трасс, таких как stftool, инструмент верификации, itcrin.

Ограничения на визуализацию данных на ввод ITAC

Intel Trace Analyzer спроектирован для обработки файлов трасс очень больших размеров. Потому может анализировать трассы размером в гигабайты. Каждая диаграмма отображает информацию с учетом возможного разрешения и не отображает мелкие детали для лучшей читаемости. Однако можно добиться любого уровня детализации при приближении масштаба. Такой вариант позволяет эффективно отображать очень объемные файлы и потому не требует много времени на обработку и пересчет статистики. Каждая диаграмма может быть распечатана также с максимально возможным уровнем детализации.

VITE - Visual Trace Explorer

URL: <http://vite.gforge.inria.fr/>

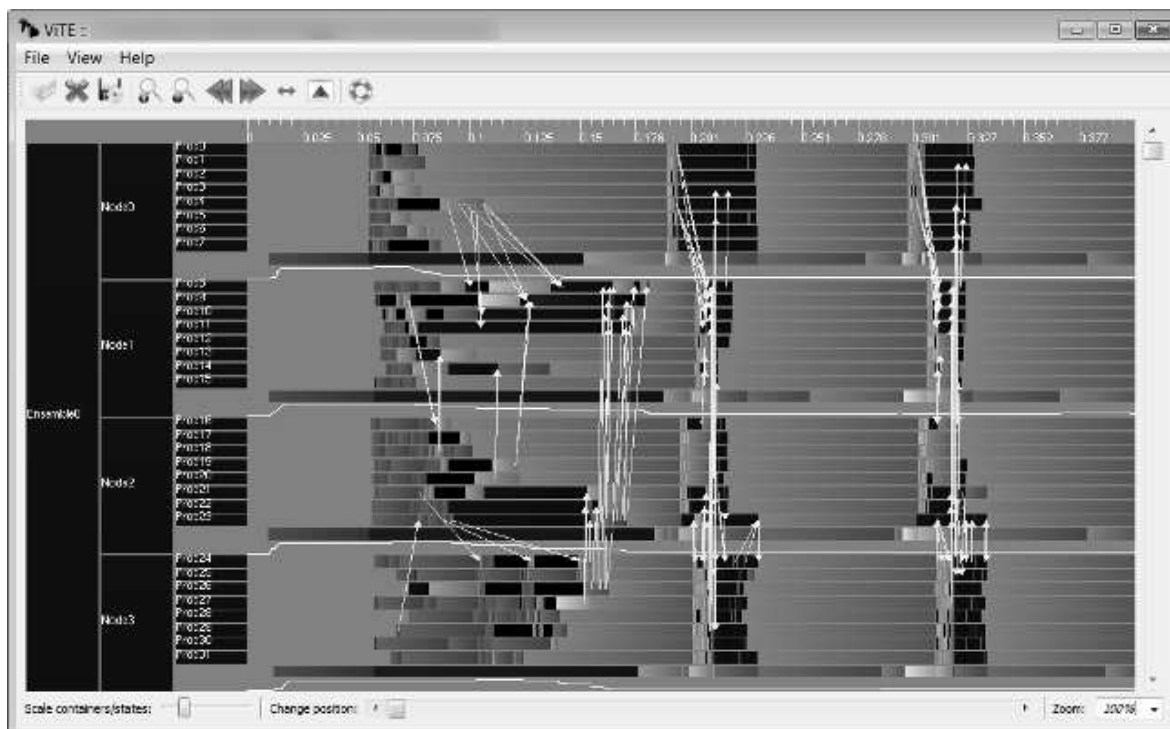
Поддерживаемые форматы: Paje, TAU, OTF.

Где разрабатывается:

Лицензия: open source software licenced under CeCILL-A.

Поддерживаемые системы: GNU/Linux, MacOS X, Windows

Тип: Средство визуализации



Для генерации трасс можно использовать Generic Trace Generator (GTG) (в том числе для генерации трасс Page).

Г.5 Охранные документы

Рег. номер 2009610697 (29.01.2009)

Авторы Ясенюк А.В., Минеев М.И., Калинин Ю.И., Зенин В.В., Чугаев В.В., Болин В.П., Лыготчиков С.Я., Павленко Ю.М., Сапегин К.В., Сапарина Т.П., Дрожжина А.Ю.

Правообладатель Федеральное государственное унитарное предприятие "Лётно-исследовательский институт имени М.М. Громова"

Программа Программно-математическое обеспечение лётно-моделирующего комплекса для проведения исследований и сопровождения лётных испытаний пилотажно-навигационного оборудования летательных аппаратов

Аннотация Программно-математическое обеспечение лётно-моделирующего комплекса для проведения исследований и сопровождения лётных испытаний пилотажно-навигационного оборудования летательных аппаратов применяется при полунатурном сопровождающем моделировании в процессе лётных испытаний самолетов. Программно-математическое обеспечение предназначено для моделирования динамики полета самолетов и элементов пилотажно-навигационного оборудования в локальной сети стенда и выполняет следующие функции: исследование математических моделей самолетов с целью обеспечения их

сходимости с характеристиками реального самолета, автоматизация управления на режимах общего и специального самолетовождения, определение облика информационно-управляющего поля кабины для перспективных самолетов, обеспечение автоматизированной обработки результатов исследований.

Тип ЭВМ IBM PC-совмест. ПК

Язык C++

ОС Microsoft XP

Объем программы 120 Кб

Рег. номер 2010613286 (19.05.2010)

Авторы Барабаненков Ю.А., Гурвиц А.В., Девятов С.А., Макарова И.В.

Правообладатель Федеральное государственное унитарное предприятие "Государственный научно-исследовательский институт авиационных систем" (ФГУП "ГосНИИАС")

Программа "Динамическое моделирование полетной обстановки" (SAMVME)

Аннотация Программа предназначена для имитации полетной обстановки при полунатурном моделировании (ПНМ) совместно с макетом кабины, ее органами управления и пилотом, находящимся в макете кабины, и обеспечения динамического моделирования процесса движения самолета совместно с другими разнотипными летательными аппаратами в реальном масштабе времени. Программа обеспечивает реализацию информационного обмена между вычислительным центром (ВЦ), имитаторами бортовых систем и внешней сигнальной обстановки, средствами моделирования и бортовыми системами в соответствии с протоколами информационного взаимодействия и временными диаграммами, а также управление процессом регистрации и анализа результатов ПНМ.

Программа может быть использована в составе ВЦ на стендах Главных конструкторов, комплексах полунатурного моделирования для отработки и интеграции бортового радиоэлектронного оборудования современных и перспективных авиационных комплексов

методами ПНМ.

Тип ЭВМ Промышленный компьютер с процессором уровня не ниже Intel Pentium 233MHz; ОЗУ не менее 16MB; видеоадаптер с разрешением не менее 1024x768,256 цветов; сетевой контроллер NE2000-совместимый

Язык C++ в интегрированной среде разработки Borand C++ версии 5.02

ОС Windows 95/98

Объем программы 905 Кб

Рег. номер 2010613287 (19.05.2010)

Авторы Барабаненков Ю.А., Гурвиц А.В., Девятов С.А., Макарова И.В.

Правообладатель Федеральное государственное унитарное предприятие "Государственный научно-исследовательский институт авиационных систем" (ФГУП "ГосНИИАС")

Программа "Задание условий работы комплекса полунатурного моделирования" (Startset)

Аннотация Программа предназначена для задания в удобной для восприятия оператора форме начальных условий (НУ) полетной обстановки, режимов работы комплекса полунатурного моделирования (КПМ), средств моделирования, а также для выбора варианта моделирования из рабочих каталогов. Программа работает в интерактивном режиме с использованием удобного пользовательского интерфейса последовательно раскрывающихся диалоговых окон, в которых оператор формирует файл НУ, а также просматривает заданную информацию в справочных окнах. В программе используется метод задания положения разнотипных летательных аппаратов (ЛА) в полярной системе координат относительно самолета с последующим пересчетом в абсолютную декартовую систему координат, что предоставляет оператору достаточно простое средство для формирования различных начальных условий, отличающихся лишь ориентаций по сторонам света самолета и других ЛА. Программа может быть использована в составе вычислительных центров на стендах Главных конструкторов, КПМ для отработки и интеграции бортового радиоэлектронного оборудования современных и перспективных авиационных комплексов методами полунатурного моделирования.

Тип ЭВМ Промышленный компьютер с процессором не ниже Intel Pentium 233MHz, ОЗУ не менее 16МВ, видеоадаптер с разрешением не менее 1024x768,256 цветов

Язык С++ с использованием библиотеки классов Object Windows в интегрированной среде разработки Borland С++ версии 5.02

ОС Windows 95/98

Объем программы 135 Кб

Рег. номер 2010613288 (19.05.2010)

Авторы Барабаненков Ю.А., Гурвиц А.В., Девятов С.А., Макарова И.В.

Правообладатель Федеральное государственное унитарное предприятие "Государственный научно-исследовательский институт авиационных систем" (ФГУП "ГосНИИАС")

Программа "Тестирование информационных каналов последовательного кода" (RTM)

Аннотация Программа предназначена для регистрации и отображения на экране монитора в реальном масштабе времени и после сеанса моделирования в удобной для оператора графической форме информации, передаваемой по каналам последовательного кода РТМ-

1495. Программа может быть использована в составе вычислительных центров на стендах Главных конструкторов, комплексах полунатурного моделирования для отработки и интеграции бортового радиоэлектронного оборудования современных и перспективных авиационных комплексов методами полунатурного моделирования.

Тип ЭВМ Промышленный компьютер с процессором уровня не ниже Intel Pentium 233MHz; ОЗУ не менее 16МВ; видеоадаптер с разрешением не менее 1024x768,256 цветов; сетевой контроллер NE2000-совместимый

Язык Visual Basic 4.0

ОС Windows 95/98

Объем программы 255 Кб

Рег. номер 2010613289 (19.05.2010)

Авторы Барабаненков Ю.А., Гурвиц А.В., Девятков С.А., Макарова И.В.

Правообладатель Федеральное государственное унитарное предприятие "Государственный научно-исследовательский институт авиационных систем" (ФГУП "ГосНИИАС")

Программа "Диспетчер работы вычислительного центра" (DISPATCH)

Аннотация Программа DISPATCH предназначена для централизованного управления процессом полунатурного моделирования и отображения состояния программ вычислительного центра (ВЦ) на различных компьютерах ВЦ. Программа DISPATCH обеспечивает синхронный запуск, завершение, приостановку и возобновление работы программного обеспечения при полунатурном моделировании (ПНМ). Программа может быть использована в составе ВЦ на стендах Главных конструкторов, комплексах полунатурного моделирования для отработки и интеграции бортового радиоэлектронного оборудования современных и перспективных авиационных комплексов методами ПНМ.

Тип ЭВМ Промышленный компьютер с процессором уровня, не ниже Intel Pentium 233MHz, ОЗУ, не менее 16МВ, видеоадаптер с разрешением, не менее 1024x768,256 цветов, сетевой контроллер NE2000-совместимый

Язык С++ среде разработки Borland С++ версии 5.02

ОС Windows 95/98

Объем программы 26 Кб

Рег. номер 2010613290 (19.05.2010)

Авторы Барабаненков Ю.А., Гурвиц А.В., Девятков С.А., Макарова И.В.

Правообладатель Федеральное государственное унитарное предприятие "Государственный научно-исследовательский институт авиационных систем" (ФГУП "ГосНИИАС")

Программа "Отображение тактической обстановки" (ТО)

Аннотация Программа предназначена для визуализации моделируемой тактической обстановки в реальном масштабе времени при полунатурном моделировании (ПНМ). Обеспечивает наглядное представление о процессе движения летательных аппаратов с помощью вывода на монитор графической информации об их взаимном расположении и информации о фазовых координатах в виде числовых формуляторов. Программа может быть использована в составе вычислительных центров на стендах Главных конструкторов, комплексах полунатурного моделирования для отработки и интеграции бортового радиоэлектронного оборудования современных и перспективных авиационных комплексов методами ПНМ.

Тип ЭВМ Промышленный компьютер с процессором уровня не ниже Intel Pentium 233MHz; ОЗУ не менее 16MB; видеоадаптер с разрешением, не менее 1024x768,256 цветов; сетевой контроллер NE2000-совместимый

Язык C++ с использованием библиотеки классов Object Windows в интегрированной среде разработки Borland C++ версии 5.02

ОС Windows 95/98

Объем программы 43 Кб

Рег. номер 2010613291 (19.05.2010)

Авторы Барабаненков Ю.А., Гурвиц А.В., Девятов С.А., Макарова И.В.

Правообладатель Федеральное государственное унитарное предприятие "Государственный научно-исследовательский институт авиационных систем" (ФГУП "ГосНИИАС")

Программа "Программа управления имитатором подвесных авиационных средств" (SSS)

Аннотация Программа предназначена для отработки бортовых систем управления подвесными авиационными средствами (ПАС) в реальном масштабе времени с помощью имитации многовариантности ПАС, способов их применения и оперативной смены вариантов загрузки, а также для мониторинга информационного обмена имитатора ПАС с бортовыми системами. Программа может быть использована на стендах Главных конструкторов, а также на комплексах полунатурного моделирования.

Тип ЭВМ Промышленный компьютер с процессором уровня не ниже Intel Pentium 233MHz; ОЗУ не менее 16MB; видеоадаптер с разрешением не менее 1024x768,256 цветов; сетевой контроллер NE2000-совместимый

Язык Visual Basic 4.0

ОС Windows 95/98

Объем программы 1500 Кб

Рег. номер 2010613553 (28.05.2010)

Авторы Толмачева М.В.

Правообладатель Открытое акционерное общество "Концерн "Центральный научно-исследовательский институт "Электроприбор"

Программа Экспертная система постанализа вычислительных процессов в приборах навигационного комплекса

Аннотация Программа предназначена для анализа трасс событий вычислительного процесса при отладке и диагностировании приборов навигационного комплекса. Программа позволяет обнаруживать отклонение параметров от номинальных значений, которые могут быть следствием отказов аппаратуры и ошибок проектирования, среди которых: ошибки прикладного программного обеспечения; ошибки в программном обеспечении операционной системы реального времени; ошибки проектирования и, в частности, ошибки планирования вычислительного процесса.

Тип ЭВМ IBM PC-совмест. ПК

Язык C++ Builder

ОС Windows XP

Объем программы 1286 Кб

Рег. номер 2003611671 (29.05.2003)

Авторы Не указаны.

Правообладатель Общество с ограниченной ответственностью «Редлаб ЛТД»

Программа Среда моделирования ДИАНА

Аннотация Среда моделирования ДИАНА предназначена для системного проектирования и анализа распределенных вычислительных систем. Она обеспечивает решение следующих задач: описание структуры и алгоритмов функционирования аппаратуры и программного обеспечения исследуемой системы; воспроизведение функционирования исследования системы методом имитационного моделирования; анализ логики работы системы и определение количественных характеристик её производительности.

Тип ЭВМ PC Pentium II/Celeron/Athlon не ниже 300 МГц, RAM не менее 128 Мб, жёсткий диск не менее 4Гб, цветной монитор SVGA не менее 15”.

Язык Си, C++

ОС Linux Debian версии не ниже 2.0.

Объем программы 20507 Кб

Рег. номер 2010612302 (28.01.2010)

Авторы Не указаны.

Правообладатель Общество с ограниченной ответственностью «Редлаб ЛТД»

Программа Средства функционального тестирования комплексов бортового оборудования

Аннотация Программный модуль «Программное средство разработки и отладки тестовых сценариев» применяется в составе специального программного комплекса «Комплекс программных средств разработки и выполнения функциональных тестов ИУС, средств планирования информационного обмена в ИУС» для поддержки разработки частных моделей и подготовки имитационного эксперимента.

Тип ЭВМ программа предназначена для эксплуатации на ЭВМ, обладающих следующими характеристикам: объём оперативной памяти – не менее 512 Мбайт; частота процессора – не менее 1,6 ГГц; объём жёсткого диска – не менее 120 Гбайт; дисковод DVD/CD-RW; монитор – не менее 17"; сетевая карта Ethernet; адаптеры МКИО, ДПК, FC-AE-ASM (для инструментальных машин, выполняющих обмен по каналам этих типов); клавиатура; мышь..

Язык Си, С++

ОС Debian Linux версии не ниже 4.0 с расширениями реального времени (real-time preemption patch).

Объем программы 7,36 Мб

Рег. номер 2010612303 (26.03.2010)

Авторы Не указаны.

Правообладатель Общество с ограниченной ответственностью «Редлаб ЛТД»

Программа Анализатор МКИО

Аннотация ПО «Анализатор МКИО» предназначено для обеспечения регистрации информационного обмена по каналам МКИО ГОСТ Р 52070-2003, отображения и поддержки анализа пользователем результатов регистрации, а также выдачи в канал МКИО заданной пользователем информации.

Тип ЭВМ программа предназначена для эксплуатации на ЭВМ, обладающих следующими характеристикам: объём оперативной памяти – не менее 512 Мбайт; частота процессора – не менее 1,6 ГГц; объём жёсткого диска – не менее 120 Гбайт; дисковод DVD/CD-RW; монитор – не менее 17"; сетевая карта Ethernet; адаптер МКИО; клавиатура; мышь.

Язык Си, С++

ОС Windows XP.

Объем программы 276 Кб

Рег. номер 2010612304 (26.03.2010)

Авторы Не указаны.

Правообладатель Общество с ограниченной ответственностью «Редлаб ЛТД»

Программа Среда моделирования «ДИАНА» версия 7.2

Аннотация ПО ДИАНА 7.2 предназначено для поддержки выполнения полунатурного моделирования и испытаний БВК и ИУС, в том числе поддержки распределённого выполнения моделей в реальном времени на нескольких компьютерах, а также обмена данными по бортовым каналам информационного обмена.

Тип ЭВМ ПО ДИАНА 7.2 для эксплуатации на ЭВМ, обладающих следующими характеристикам: объём оперативной памяти – не менее 512 Мбайт; частота процессора – не менее 1,6 ГГц; объём жёсткого диска – не менее 120 Гбайт; дисковод DVD/CD-RW; монитор – не менее 17"; сетевая карта Ethernet; клавиатура; мышь.

Язык Си, C++

ОС Debian Linux версии не ниже 4.0 с расширениями реального времени (real-time preemption patch).

Объем программы 610 Кб

Рег. номер 2010612305 (26.03.2010)

Авторы Не указаны.

Правообладатель Общество с ограниченной ответственностью «Редлаб ЛТД»

Программа Анализатор ДПК

Аннотация ПО «Анализатор ДПК» предназначено для обеспечения регистрации информационного обмена по линиям ДПК ГОСТ 18977-79, РТМ 1495-75, отображения и поддержки анализа пользователем результатов регистрации, а также выдачи на линию ДПК заданной пользователем информации.

Тип ЭВМ ПО «Анализатор ДПК» предназначен для эксплуатации на ЭВМ, обладающих следующими характеристикам: объём оперативной памяти – не менее 512 Мбайт; частота процессора – не менее 1,6 ГГц; объём жёсткого диска – не менее 120 Гбайт; дисковод DVD/CD-RW; монитор – не менее 17"; сетевая карта Ethernet; адаптер ДПК; клавиатура; мышь.

Язык Си, C++

ОС Debian Linux версии не ниже 4.0 с расширениями реального времени (real-time preemption patch).

Объем программы 241 Кб