

На правах рукописи



УЖИНСКИЙ
Александр Владимирович

МЕТОДЫ И СРЕДСТВА МОНИТОРИНГА СЕРВИСОВ
ПЕРЕДАЧИ ДАННЫХ В ГЛОБАЛЬНЫХ РАСПРЕДЕЛЕННЫХ
ИНФРАСТРУКТУРАХ

Специальность: 05.13.01 — системный анализ, управление
и обработка информации (отрасль: информационные,
телекоммуникационные и инновационные технологии)

Автореферат диссертации на соискание ученой степени
кандидата технических наук

29 АПР 2010

Дубна 2010

Работа выполнена в Международном университете природы, общества и человека «Дубна» на кафедре системного анализа и управления.

Научный руководитель:

кандидат физико-математических наук Кореньков Владимир Васильевич

Официальные оппоненты:

доктор физико-математических наук Ильин Вячеслав Анатольевич

доктор технических наук Сахаров Юрий Серафимович

Ведущая организация:

Институт прикладной математики им. М. В. Келдыша

Защита состоится «14» мая 2010 года в 16 часов в аудитории 1-300 на заседании диссертационного совета Д800.017.02 при Международном университете природы, общества и человека «Дубна» по адресу: 141981 Московская обл., г. Дубна, ул. Университетская, д. 19.

С диссертацией можно ознакомиться в библиотеке Международного университета природы, общества и человека «Дубна».

Автореферат разослан: «5» апреля 2010

Ученый секретарь
диссертационного совета Д800.017.02
кандидат физико-математических наук
Токарева Надежда Александровна



ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы

Уже более 20 лет распределенные компьютерные инфраструктуры используются для решения широкого класса задач. Одним из наиболее интересных направлений в данной области является грид – технология создания географически распределенных программно-аппаратных компьютерных инфраструктур, предназначенных для объединения разнородных ресурсов с целью повышения эффективности их использования и экономии затрат. Физика, химия, астрономия, медицина, науки о земле - это далеко не полный список областей, в которых используются грид-инфраструктуры. Одной из важнейших функций грид является распределение, хранение и обработка огромных массивов данных, за что отвечают специализированные сервисы.

В своей работе сервисы передачи данных взаимодействуют с множеством компонентов. Грид-инфраструктуры могут состоять из сотен сайтов (региональных компьютерных центров), на которых установлены различные системы хранения файлов. Для взаимодействия с системами хранения используются различные сервисы, а за передачу данных отвечают различные протоколы. Сервисы передачи данных организуют работу всех элементов и сервисов управления данными, а также обрабатывают информацию об ошибках на любом этапе передачи данных. Ежедневно подобными сервисами выполняются тысячи транспортных задач, пересылаются десятки тысяч файлов, общие объемы передаваемой информации составляют десятки терабайт в день! От надежности и корректности работы подобных сервисов напрямую зависит успех использующих их проектов.

Программная составляющая грид – middleware, промежуточное программное обеспечение (ППО). Среди наиболее известных ППО можно выделить gLite и Globus. Сервис передачи файлов в gLite называется FTS (File Transfer Service), его аналог в Globus – RFT (Reliable File Transfer Service). К концу 2006-го года FTS и RFT находились на этапе становления и набор средств их мониторинга был весьма ограничен. В основном он состоял из скриптов, визуализирующих информацию, предоставляемую самими сервисами посредством интерфейса командной строки, или комплексов мониторинга широкого профиля, отражающими общую информацию по передачам данных. Ни одна из систем не предоставляла полную информацию о состоянии сервисов, истории их функционирования, а главное об ошибках, возникающих в распределенных системах передачи данных. Подобная ситуация

неблагоприятно сказывалась на надежности функционирования сервисов и требовала оперативного решения.

Актуальной задачей является и решение проблемы адаптивности грид. Из-за сложности и разнородности грид-инфраструктур существует множество трудностей в их поддержании и управлении. Для их установки, конфигурирования и контроля требуется множество опытных специалистов, которых зачастую очень сложно найти, а тем более обучить. Современные грид-инфраструктуры не способны автоматически подстраиваться под изменения программно-аппаратной среды (отсутствует адаптивность), вследствие чего реакция на сбои и изменения в инфраструктуре происходит с большим запозданием. Автоматизация работы как отдельных, так и групп сервисов способно самым положительным образом сказаться на адаптивности, а значит и на эффективности работы грид-инфраструктур.

Цель работы:

Разработка новых и совершенствование существующих методов и средств анализа и обработки информации о функционировании сервисов передачи файлов в глобальных распределенных инфраструктурах.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Исследование структуры и принципов функционирования сервисов передачи файлов в распределенных инфраструктурах;
2. Разработка классификации ошибок передачи данных, возникающих в грид-инфраструктурах.
3. Разработка подходов к проектированию систем мониторинга сервисов передачи файлов в грид;
4. Разработка системы мониторинга сервиса передачи данных и исследование полученной информации о сбоях.
5. Исследование возможностей адаптивности и автоматизации сервисов передачи файлов.

Методы исследования

Для решения поставленных задач использовались понятия и методы системного анализа, математической статистики и проектирования программных средств.

Научная новизна

1. Предложена классификация ошибок, возникающих в распределенных системах передачи данных, основанная на четкой иерархии описания сбоев и использовании механизма паттернов.

2. Предложен и реализован подход к проектированию систем мониторинга сервисов передачи файлов в крупных распределенных инфраструктурах.
3. Предложен новый сервис – сервис управления грид (Grid Management Service, GMS), способный решить проблему адаптивности глобальных распределенных систем.

Практическая ценность работы

1. Создана система мониторинга сервиса передачи файлов, предоставляющая широкие возможности для анализа и обработки информации о его функционировании.
2. Проведено исследование ошибок, возникающих при передаче файлов в крупных распределенных грид-инфраструктурах. Исследование позволило определить самые распространенные ошибки, причины их возникновения, основные области возникновения ошибок, выявить программные и логические ошибки в приложениях.
3. С целью автоматизации работы сервиса передачи файлов и апробации идей GMS создан прототип модуля экспертной системы, способный значительно облегчить работу администраторов систем передачи данных.

Основные положения, выносимые на защиту

1. Архитектурные особенности сервиса передачи файлов таковы, что для решения задач мониторинга сервиса требуется использовать модифицированные методы мониторинга из различных областей.
2. Предложенная классификация ошибок позволяет решить вопросы разделения, объединения и детализации сбоев, возникающих в распределенных системах передачи данных.
3. Разработанный подход к проектированию систем мониторинга сервисов передачи данных в глобальных распределенных инфраструктурах стандартизирует разработку приложений мониторинга.
4. Разработанная система мониторинга предоставляет широкие возможности для анализа и обработки информации о функционировании сервиса передачи файлов.
5. Предложенный сервис управления грид (Grid Management Service, GMS) способен решить проблему адаптивности глобальных распределенных систем.

Апробация работы

Результаты работы неоднократно докладывались на семинарах ЛИТ ОИЯИ, Дубна, рабочих совещаниях и семинарах ITGS, ЦЕРН, WLCG Service Reliability Workshop, ЦЕРН (26.11.2007), а так же на конференциях

«Distributed computing and Grid technologies in science and education», GRID-2008, Дубна (30.5-4.06.3008) , СЕР (Computing in High Energy and nuclear Physics) 2009, Прага (20-27.03.2009) и «Молодежь и XXI век», Курск (26.5-29.5.2009). Работа была награждена первой премией молодых ученых и специалистов в номинации – «научно-технические прикладные работы», на 13-ой зимней конференции ОМУС-2009 (Объединение молодых учёных и специалистов ОИЯИ) (16.02-21.02.2009) и признана лучшей на 16-й научной конференции студентов, аспирантов и молодых специалистов университета «Дубна» (23.3-3.4.2009), а так же конференции «Информационные системы и технологии 2009», Обнинск (15.05.2009).

Публикации и личный вклад автора

По результатам диссертации опубликовано 7 работ из них три в списке изданий, рекомендуемых ВАК для опубликования основных научных результатов диссертации [1, 2, 4].

Автор принимал непосредственное участие в создании классификации ошибок и подхода к проектированию средств мониторинга сервисов передачи данных. Разработка систем мониторинга сервисов передачи данных и определение основных принципов функционирования, архитектурных особенностей и задач сервиса управления грид проводилась лично автором.

Объем и структура работы

Диссертация состоит из четырех глав, введения, заключения и приложения, содержит 118 страниц, включает 33 рисунка, 4 таблицы и библиографию из 92 наименований.

Содержание работы

Во введении обоснована актуальность выбранной темы, определены цель и задачи, решаемые в работе. Перечислены научные результаты, полученные в диссертации. Отражена практическая ценность полученных результатов и приведены сведения об апробации результатов диссертационной работы.

В первой главе представлены результаты исследования внутреннего устройства и методов взаимодействия FTS (File Transfer Service), как типового сервиса передачи данных. FTS создавался на основе RFT, но на 2007 год являлся более развитым – предоставлял большие функциональные возможности, в связи с чем было решено исследовать именно его.

Типичная грид-инфраструктура выглядит как набор территориально распределенных, связанных между собой ресурсов, принадлежащих

различным организациям, но использующихся совместно. Ресурсы могут использоваться различными пользователями и организациями в зависимости от локальных и глобальных политик безопасности и договоренностей. В региональных центрах (сайтах) используются различные системы хранения данных. В gLite поддерживаются такие системы хранения данных, как Castor, dCache и DPM. Для взаимодействия с системами хранения (предоставления унифицированного интерфейса) существует специальный сервис – SRM (Storage Resource Manager). За перемещение данных на физическом уровне отвечает GridFTP – основной протокол передачи файлов в грид. Для обеспечения надежного и эффективного механизма передачи файлов используется сервис передачи файлов – FTS (File Transfer Service). FTS позволяет:

- ✓ задавать правила использования ресурсов сайта, в соответствии с политиками сайтов и организаций;
- ✓ предотвращать перегрузку сетей и хранилищ данных;
- ✓ отслеживать общую ситуацию и получать комплексную информацию об ошибках, возникающих во время работы сервиса;
- ✓ обеспечить надежный механизм передачи файлов типа точка-точка;
- ✓ предоставить удобный способ распределения ресурсов между экспериментами;
- ✓ предоставить возможности мониторинга и контроля для администраторов сайтов;
- ✓ дать возможность контролировать запросы, поступающие со стороны пользователей, устанавливать последовательность их исполнения, а также ранжировать их по приоритетности для менеджеров организаций.

В задачи FTS входит и обработка сообщений об ошибках от всех систем, сервисов и протоколов, участвующих в передаче данных в глобальных распределенных инфраструктурах. Это является весьма не тривиальной задачей, поскольку сообщения о сбоях имеют различные структуру и содержание не только в различных компонентах но и в различных версиях и модификациях одного и того же компонента.

FTS состоит из ряда отдельных элементов, которые взаимодействуют только с центральной базой данных на платформе Oracle. Каждый элемент сервиса выполняет свою отдельную задачу. Например, агенты виртуальных организаций (VO) отвечают за управление запросами на передачу данных от определенных организаций. Агенты каналов отвечают за инициализацию, контроль передачи данных и распределение ресурсов канала между виртуальными организациями и т.д. Благодаря тому, что отдельные компоненты взаимодействуют только с базой данных, значительно повышается надежность системы, т.к. при выходе из строя

любого из элементов (кроме баз данных) сервис продолжит функционировать, хотя и не в полном объеме.

На рис. 1 представлена схема взаимодействия объектов FTS. Любая задача задается и принадлежит одной из виртуальных организаций (ВО) и состоит из набора файлов. Агенты отвечают за передачу файлов, используя каналы. Канал соединяет два сайта, на которых установлен сервис управления элементами хранения (SRM) и непосредственно элементы хранения (SE) на которых хранятся файлы. В процессе передачи может произойти ошибка, которая ее прерывает. Ошибка может произойти из-за проблем с системами хранения, сервисами управления элементами хранения, каналами, авторизацией, квотами на ресурсах самим сервисом передачи файлов и т.д.



Рис. 1. Схема взаимодействия объектов FTS.

Большое количество разнородных объектов, с которыми приходится взаимодействовать сервису в своей работе, а так же интенсивность потоков данных предъявляют к системам мониторинга сервиса весьма серьезные требования.

В главе так же изложены результаты исследований технологий мониторинга информационных систем с 60-х годов 20-го века и до наших дней, проводившегося с целью определения наиболее перспективных методов, которые при должной модификации могли бы быть использованы для решения задач мониторинга сервиса передачи файлов.

Уже в 70-х годах появились специализированные методологии создания систем мониторинга, скрупулезно рассматривались вопросы анализа данных мониторинга и возможности применения различных статистических методов: структурного, корреляционного, регрессионного

или факторного анализа, цепей Маркова и т.д. В качестве методов представления результатов предлагались текстовые отчеты, таблицы, а также различные диаграммы и графики. В конце 70-х годов активное развитие баз данных инициировало новую волну исследований, в результате которых появились технологии сессий, транзакций, а так же механизм оповещений. Суть механизма оповещения состояла в том, что пользователь мог определить набор различных правил, при нарушении которых ему выдавалось бы сообщение. Немаловажное значение сыграл и механизм паттернов, основная идея которого состояла в том, что сообщение об ошибке характеризовалось ее определенными неизменными частями. Персонализированная информация, например имена пользователей, исключалась из рассмотрения, что позволяло объединять данные об ошибках и работать с ними на более высоком уровне. С появлением распределенных систем, возникли и новые задачи для мониторинга, реализация которых гораздо сложнее, нежели в централизованных системах. Одним из основных требований к системам мониторинга распределенных систем является их расширяемость, как физическая, так и функциональная. В основном, для этого применялся подход, в котором задачи сбора данных отделялись от задач их обработки и предоставления пользователям. В развитии компьютерных сетей важнейшую роль сыграли стандарты RFC (Request for Comments). В аспекте мониторинга, стандарты описывали свойства, состояния, методы, а так же протоколы взаимодействия между объектами сети. Основным плюсом подобного подхода явилась независимость от аппаратного и программного обеспечения при использовании стандартных команд и запросов.

В грид-мониторинге, в доработанном и модифицированном виде используются подходы, разработанные для распределенных систем и компьютерных сетей. Разработчики ГПО создают специальные сервисы, отвечающие за сбор и распространение данных об элементах сети: Globus – MDS (monitoring and discovery service), European DataGrid – R-GMA (Relational-Grid Monitoring Architecture), Condor – Hawkeye, а за анализ и представление информации пользователям обычно отвечают другие средства, разрабатываемые отдельными группами в рамках одного или нескольких экспериментов. Существуют примеры создания комплексных средств мониторинга, основанных на технологиях Web-сервисов. Подобные комплексы получают информацию из различных источников, что позволяет расширять список отслеживаемых параметров работы грид: MonALISA, GridICE, BOSS, MapCenter, Nagios и т.д. Таким образом, задачи сбора и обработки информации отделяются от ее анализа и представления.

Для мониторинга сервисов передачи данных полнофункциональной системы не было разработано, существовал набор скриптов, который собирал информацию и визуализировал некоторые аспекты функционирования сервисов передачи файлов. Необходимо было, опираясь на перспективные методы и подходы (механизм паттернов, механизм оповещений, корреляционный анализ и т.д.), разработать архитектуру системы мониторинга сервисов передачи данных, которая бы позволяла:

- ✓ Получать полную информацию о состоянии сервисов, их производительности, надежности и истории функционирования.
- ✓ Точно определять причины сбоев при передаче данных (ошибки дисковых серверов, систем управления данными, локальных сетей, каналов передачи данных, компонент промежуточного программного обеспечения, приложений и т.д.).
- ✓ Анализировать информацию об ошибках: полнота списка ошибок, частота возникновения определенных ошибок, история развития проблемных ситуаций, причины возникновения ошибок и т.д.

Результатом главы является вывод о том, что для повышения надежности функционирования сервиса передачи файлов, набор средств мониторинга необходимо расширять, причем архитектурные особенности сервиса таковы, что для решения задач мониторинга сервиса требуется использовать модифицированные методы мониторинга из различных областей.

Вторая глава посвящена методам и средствам, разработанным для хранения, обработки и визуализации информации о сбоях, возникающих при передаче данных в грид-инфраструктурах.

FTS отвечает за взаимодействие различных элементов хранения и передачи данных, их различных версий и модификаций, поэтому спектр возникающих ошибок, приводящих к неудачному завершению передачи, довольно велик. Если проанализировать текст различных ошибок, то часто в нем содержится уникальная персонифицированная информация – время, дата, название файла, атрибуты пользователя и т.д.:

```
Failed to complete PrepareToPut request [id] on remote SRM [srm]:  
[SRM_INVALID_REQUEST]
```

В приведенном примере [id] – уникальный идентификатор запроса, [srm] – название SRM, к которому происходит запрос, и [SRM_INVALID_REQUEST] – внутренний идентификатор запроса SRM, могут быть различными, однако, тип ошибки при этом не меняется.

Необходимо было разработать удобный механизм хранения, добавления и изменения информации об ошибках. Автором было

предложено использовать механизм паттернов – устойчивых составных частей, однозначно характеризующих тип ошибки. Так в приведенном примере части «*Failed to complete PrepareToPut request*» и «*on remote SRM*» не меняются и могут быть использованы для определения типа сбоя. Это позволит объединять данные о подобных ошибках и работать с ними на более высоком уровне. При использовании подобного механизма следует определить минимальное достаточное число составных частей, описывающих сбой. В результате изучения всех имеющихся сообщений об ошибках, было определено, что для однозначного определения типа сбоя достаточно трех его составных частей.

Одним из источников информации о работе сервиса являются файлы отчетов - log-файлы, в которых протоколируется его работа. У сервиса передачи файлов в log-файлах содержится подробная информация о передачах данных. Требовалось обрабатывать эти файлы, сохранять полученные данные, а так же предоставлять доступный интерфейс для работы с ними. Соответственно, для прототипа системы мониторинга автором была предложена трехзвенная архитектура: извлечение данных – набор скриптов, ответственных за перемещение и обработку log-файлов (реализация – Perl, shell), хранение данных – схема данных системы, реализованная под управлением СУБД MySQL, представление данных – информация из базы данных доступная через web-интерфейс (реализация – PHP, XHTML).

Web-интерфейс системы имел несколько частей, используемых для получения различной информации Пользователи могли использовать различные фильтры, определять временные интервалы (последняя информация, информация за последние 24 часа и т.д.) и методы отображения результата (таблицы, диаграммы, графики). В системе была предусмотрена возможность создания типовых отчетов, рейтингов и получения различных статистических выкладок.

Автором было выделено более 350 паттернов ошибок, для удобства работы с ними было решено ввести объединяющие классы сбоев. В результате изучения имеющихся паттернов, было выделено 16 базовых классов ошибок. Например, “user” – ошибки, возникающие по вине пользователей, “tcp” – ошибки подключения к сети, “t_dcache” – ошибки системы хранения данных dCache сайта, на который производится запись и т.д. Развитие данного направления исследований привело к введению в FTS версии 2.0 новой классификации ошибок, которая будет описана позже.

Практическим результатом данного этапа явилась разработка прототипа системы мониторинга, который в течение полутора лет использовался в ЦЕРН. Этот прототип предоставил надежный инструмент

для поддержания работоспособности каналов и значительно расширил функциональные возможности анализа сбоев на каналах связи.

Другим практическим результатом явилось исследование сбоев, возникающих в распределенных системах передачи данных. Исходными данными для исследования послужила информация о работе 28-и каналов связи, пересылку по которым осуществлял FTS сервер, установленный в ЦЕРН, за 10 месяцев (24.03.2007 - 10.12.2007). Данные были получены из прототипа системы мониторинга, описанной выше. Автором было выделено более 350 шаблонов ошибок и следовало определить самые распространенные из них.

В рамках исследования были составлены рейтинги самых распространенных ошибок для FTS версии 1.5 и 2.0. Ошибки классифицировались и изучались причины их возникновения. Например, для FTS 2.0 в рейтинге присутствовали 6 ошибок сервиса SRM. Две ошибки SRM, занимающие первое и второе места в рейтинге, характеризовались превышением времени ожидания ответа, одна сигнализировала недоступность сервиса и три являлись программными ошибками SRM. Число GridFTP ошибок составляло 4, и они занимали 3-е, 4-е, 7-е и 14-е места; все четыре были связаны с превышением времени ожидания ответа, либо невозможностью установить соединение. По три ошибки было связано с Castor и dCache, их причины были связаны с архитектурной особенностью приложений. Ошибки пользователей занимали 8-е, 17-е и 18-е места. Таким образом, были определены приложения, при взаимодействии с которыми возникало наибольшее количество ошибок, а так же общие источники возникновения ошибок – превышение времени ожидания ответа, программные ошибки, специфические ошибки приложений и ошибки пользователей. Выяснение причин возникновения сбоев – довольно сложная задача. Во время исследования подтвердилось предположение, что определение коэффициента корреляции между новыми и хорошо изученными ошибками способно значительно сократить время на определение причин возникновения новых ошибок. Например, если у неизученной ошибки высокий коэффициент корреляции с ошибками, возникающими при работе с SRM, то, вероятно, и новая ошибка возникает по причине сбоев SRM.

Результаты проведенного исследования возникающих ошибок были изучены разработчиками сервисов и приложений, администраторами грид-сайтов и менеджерами виртуальных организаций. Легче всего искоренить программные ошибки, т.к. существуют отработанные методологии исправления критических ошибок в приложениях. Однако стоит заметить, что приложения постоянно обновляются, следовательно, работы по выявлению программных ошибок должны проводиться постоянно.

Специфические ошибки приложений – довольно сложная область и, скорее всего, подобные ошибки будут присутствовать всегда, т.к. являются одним из механизмов контроля работоспособности приложений. Единственное правильное направление деятельности в данной области, это уменьшение числа таких ошибок путем своевременного реагирования на их появление. Для исправления ситуации с ошибками пользователей, очевидным представляется путь повышения квалификации пользователей, а так же увеличения качества и количества документации на приложения. Проблема превышения времени ожидания ответа, по-видимому, самая сложная из четырех. С одной стороны, логичным решением представляется увеличение размера времени ожидания ответа, но с другой - подобный шаг может самым негативным образом сказаться на состоянии сервисов. Основными задачами, решаемыми введением времени ожидания ответа, являются выявление неработоспособных элементов и предотвращение возникновения чрезмерных очередей, поэтому определение оптимальных временных интервалов, при которых система будет сохранять свою функциональность и при этом не терять производительность - сложная и кропотливая работа, которой приходится заниматься администраторам, как конкретных узлов, так и глобальных сервисов. Пожалуй, единственный способ достичь положительных результатов в данном направлении, это постоянные эксперименты и своевременное реагирования на изменения в глобальной инфраструктуре.

Благодаря исследованию были определены и описаны самые распространенные ошибки при передаче данных, изучены причины их возникновения и способы предотвращения. Еще одним результатом исследования стала апробация метода определения причины возникновения неизученных ошибок путем определения степени их связанности с уже известными ошибками. В результате в новой системе мониторинга, которая будет описана позднее, были реализованы интегрированные средства определения степени корреляции между ошибками.

В третьей главе представлена разработанная при непосредственном участии автора классификация ошибок, предоставляющая различные уровни детализации. Использование паттернов позволило стандартизировать информацию об ошибках, достичь взаимопонимания между разработчиками программных приложений и пользователями FTS, а также установить связи между различными ошибками. За время эксплуатации прототипа системы мониторинга сервиса передачи данных, в связи с развитием многих компонентов, обеспечивающих этот сервис, было выделено более 400 паттернов ошибок. Для облегчения работы с ними было определено 16 базовых объединяющих классов ошибок. В FTS

использовалось только одно поле для описания ошибки, и обработка данных о них требовала много ресурсов. Изучение всех выделенных паттернов и классов позволило совместно с коллективом разработчиков применить в FTS версии 2.0 новую классификацию ошибок, в соответствии с которой сообщение о сбое состоит из четырех частей:

Scope – источник возникновения ошибки (сайт-источник, сайт-назначения, транспортные протоколы и проблемы связи).

Category – общий класс ошибки. Было определено более 30 классов ошибок (нехватка места на диске, превышение времени ожидания и т.д.)

Phase – этап в жизненном цикле передачи, на котором произошла ошибка (подготовка к передаче, выделение ресурсов, передача и т.д.)

Message – детальное описание текста сообщения об ошибке с применением паттернов.

Четкая иерархия описания сбоев позволяет достаточно легко решать вопросы разделения, объединения и детализации. Существует возможность сгруппировать сбои по источникам их возникновения, определить их класс для получения общего представления о ситуации. При необходимости можно получить детальное описание ошибок с целью их исправления. Использование различных полей для хранения составных частей ошибок значительно уменьшает время обработки данных.

Из вышеизложенного следует второе защищаемое положение: предложенная классификация ошибок, представляет стандартизированное решение вопросов разделения, объединения и детализации сбоев, возникающих в распределенных системах передачи данных.

В главе 3 так же представлен общий подход к проектированию систем мониторинга для сервисов передачи файлов. Подход ориентирован на удовлетворение общих требований к функциональности системы, сформулированным в первой главе, направлен на минимизацию времени вычислений, использование единых стандартов и автоматизированных механизмов обработки информации. Основные положения разработанного при непосредственном участии автора подхода следующие:

1. *Все компоненты систем мониторинга должны разрабатываться в соответствии с единой классификацией ошибок.* Это позволит избежать недопонимания между пользователями и администраторами, упростить процесс разработки приложений и снизить нагрузку на БД при выполнении большинства запросов по обработке исходных данных.
2. *Агрегация и визуализация информации должны предоставлять возможность точного определения причин и мест возникновения*

проблем. У пользователей должна существовать возможность получения как общих отчетов – количество удачных/неудачных передач для всего сервиса или канала, так и максимально детализированных, вплоть до количества определенных ошибок на конкретном хосте (элементе хранения) для любой из организаций. Должна существовать возможность работы с различными уровнями детализации и получения отчетов в наиболее удобных для пользователей формах.

3. *Большинство вычислений должны производиться в базе данных.* Кроме удобного механизма добавления изменений, это позволит не зависеть от дополнительных программных и аппаратных ресурсов, и значительно снизить время при изменении и добавлении процедур.
4. *Агрегированная для различных отчетов информация должна храниться в БД.* Большинство нужных пользователям отчетов, это различные «срезы» одних и тех же исходных данных. Поэтому, следует еще на этапе проектирования определить основные объекты, информация о которых может заинтересовать пользователей, и разработать механизмы объединения исходных данных для большинства типов отчетов. Кроме того, необходимо определить временные интервалы, по истечению которых данные могут быть агрегированы.
5. *Все средства мониторинга должны быть оснащены механизмами оповещений.* Объемы информации об ошибках настолько велики, что анализ требует очень много времени. Механизмы оповещений позволяют администраторам задавать наборы правил, при срабатывании которых им будут отправлены сообщения, требующие срочного вмешательства.
6. *В системах мониторинга должна существовать возможность анализа взаимозависимостей между ошибками.* Определение причин возникновения ошибок – очень сложная задача. Изучение взаимозависимостей между новыми и хорошо изученными ошибками способно значительно сократить время на определение причин возникновения новых ошибок.

Представленный подход детально обсуждался и уточнялся на рабочих совещаниях группы IT GS в ЦЕРНе, и был принят как основа для разработки системы мониторинга сервиса передачи данных в 2008 году.

Подход призван упростить и стандартизировать разработку приложений, предназначенных для мониторинга сервисов передачи файлов в крупных распределенных инфраструктурах.

Практическим результатом этих исследований явилась разработка полнофункциональной системы мониторинга сервиса передачи данных FTS.

В процессе проектирования системы были определены четыре основные категории пользователей – менеджеры виртуальных организаций, высший менеджмент, администраторы FTS сервиса и администраторы грид-сайтов. Каждая из представленных групп заинтересована в различных данных, собранных за различные временные интервалы и представленных в различном виде. Менеджеров виртуальных организаций интересует информация об общих параметрах сервиса передачи данных и конкретная информация о грид-сайтах за определенный период времени. Администраторов грид-сайтов интересует информация о настройках и текущем состоянии каналов передачи данных, информация об ошибках на стороне их сайта и хостов. Более востребованы последние данные и информация за прошедшие сутки. Администраторов грид-сайтов интересуют как категории ошибок для определения проблем, так и конкретные описания ошибок для их устранения. Высший менеджмент интересуют самые общие отчеты, представленные в презентабельном виде за достаточно большие промежутки времени. Администраторов FTS сервиса интересует оперативная информация об ошибках, загрузке, работе сайтов и виртуальных организаций, а также степени связанности различных ошибок. Для всех категорий пользователей приветствуются любые отчеты, позволяющие в кратчайшие сроки с наименьшими затратами найти и определить причину ошибок, возникающих при передаче данных. Таким образом, при проектировании системы мониторинга существует большой спектр отчетов для разных категорий пользователей, которые необходимо стандартизировать и реализовать в системе.

После детального анализа базы данных сервиса передачи файлов автором была спроектирована модель данных системы мониторинга, предоставляющая удобную базу для создания различных отчетов. Ядро модели составляют информационные и системные таблицы, а также таблицы, содержащие обработанную и необработанную информацию. Данные в указанные таблицы поступают непосредственно из базы данных сервиса передачи файлов либо задаются пользователем. Существуют также таблицы отдельных модулей системы: «система оповещения» и «экспертная система». Общая модель данных системы мониторинга представлена на рис.2.

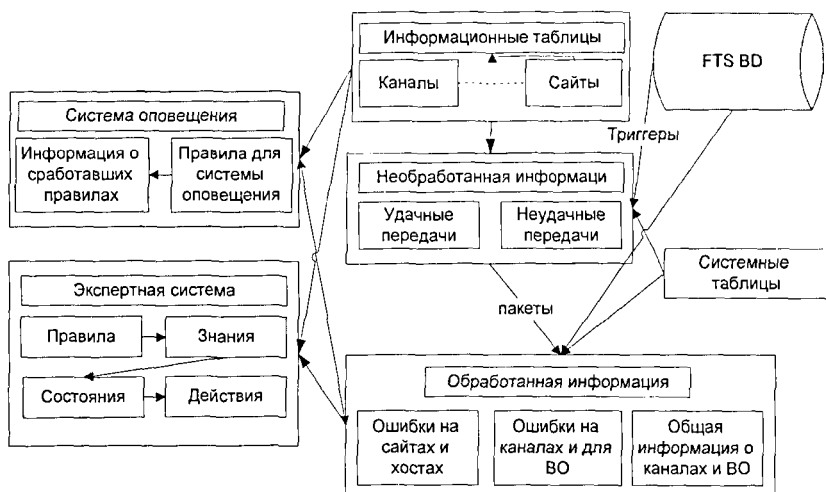


Рис. 2 Модель данных системы мониторинга FTS.

Интерфейс системы состоит из 9 модулей. У пользователей есть возможность начать свою работу с системой непосредственно из интересующего его модуля, либо с главной страницы, на которой представлены общие отчеты, позволяющие определить состояние сервиса и возможные источники проблем. Система предоставляет возможности получения широкого спектра отчетов, рейтингов, статистических выкладок и определения коэффициента корреляции для пары ошибок. Практически все отчеты системы мониторинга сервиса передачи данных снабжены перекрестными ссылками, предоставляющими удобный способ детализации результатов.

В системе реализован механизм оповещения при сбоях, позволяющий администратору сервиса создать свои собственные наборы правил (триггеры), при срабатывании которых будут выполнены определенные действия (отправлены сообщения посредством web-интерфейса, электронной почты, sms и т.д.). Триггеры можно создавать для каналов передачи, грид-сайтов, хостов и виртуальных организаций.

Существует три типа триггеров:

1. При превышении числа ошибок определенного уровня;
2. При изменении уровня ошибок более, чем на заданную величину.
3. При превышении процента неудачных передач определенного уровня.

Если пользователь работает с триггерами типа 1 и 2, он может указать идентификационный номер определенной ошибки, чтобы отслеживать только ее развитие. При работе с каналами, грид-сайтами и хостами, пользователь может указать виртуальную организацию для получения необходимых параметров. Информация о срабатывании

триггера может быть получена из специальной таблицы, что существенно упрощает использование механизма оповещений. Благодаря механизму оповещений значительно упрощается работа администраторов сервиса.

Разработанная система предоставляет удобный инструмент для получения подробной информации о работе сервиса, поддержания его работоспособности, оперативного реагирования на изменения параметров при огромных потоках передаваемых данных (до 100 терабайт в день и более).

Из изложенного следует третье и четвертое защищаемые положения:

Разработанный подход к проектированию систем мониторинга сервисов передачи данных в глобальных распределенных инфраструктурах стандартизирует разработку приложений мониторинга.

Разработанная система мониторинга предоставляет широкие возможности для анализа и обработки информации о функционировании сервиса.

В четвертой главе проводится исследование в области автоматизации и адаптивности грид. Хотя грид и является одной из наиболее интересных и перспективных технологий в сфере ИТ, существует множество трудностей в поддержке и управлении грид-инфраструктурами. Для их установки, конфигурирования и контроля требуется множество опытных специалистов, которых зачастую очень сложно найти, а тем более обучить. Грид-инфраструктуры не могут автоматически реагировать на изменения в окружающей программно-аппаратной среде, т.е. отсутствует адаптивность. Одним из наиболее перспективных направлений решения данных проблем может стать интеграция в грид идей автономного компьютеринга (*autonomic computing*). Подобный подход позволил бы усилить грид дополнительными функциями, предоставляя возможность автоматически защищать себя от возможных атак и неполадок, восстанавливаться после сбоев, оптимизировать производительность и перераспределять ресурсы в зависимости от всевозможных изменений и требований пользователей. В главе кратко рассматриваются архитектурные особенности наиболее популярных ППО, основные концепции автономного компьютеринга и текущее положение дел в автоматизации грид. Кроме того в ней выдвигается предложение о создании, определяются основные принципы функционирования, архитектурные особенности и задачи сервиса управления грид - *Grid Management Service (GMS)*. Этот сервис может взять на себя решение ряда важнейших задач:

Отключение сервиса. Если *GMS* обнаружит, что один из доверенных ему сервисов недоступен, он инициализирует перезапуск, перенастройку или переустановку. Он также может перенастроить окружение, чтобы избежать

попыток использования недоступного сервиса, либо попытаться установить новую версию сервиса на одном из свободных узлов. Если ни один из вариантов действий не принес желаемого результата, он может обратиться за помощью к вышестоящему сервису либо администратору.

Обнаружение ошибок. Сервис может обрабатывать различные проблемы, связанные с безопасностью, вычислительными или транспортными задачами. В качестве первого шага GMS будет определять «симптомы» и сравнивать их с уже известными паттернами ошибок. Если ошибка будет идентифицирована, то в соответствии с планом действий, будут выполнены соответствующие процедуры. Сложной задачей является определение всего спектра возможных ошибок и разработка действий при их возникновении. В данном направлении могут быть очень полезны функции обучения.

Обучение. В случае обнаружения неполадок GMS использует один из стандартных алгоритмов для их устранения. Могут быть использованы и более сложные механизмы, например выбор алгоритма, опираясь на степень связанности сбоев. Естественно, при довольно сложных сбоях он будет обращаться за помощью к администратору. Как только у GMS имеется решение, он будет распространять свои знания о симптоматике и методах исправления проблемы между другими GMS, либо направлять их в базу знаний, чтобы они были доступны всем сервисам.

Оптимизация. GMS может выбирать различные сочетания настроек для повышения производительности с целью оптимизации и поиска наилучшей конфигурации грид-инфраструктуры. Другой путь оптимизации – использование свободных узлов для уменьшения нагрузки или увеличения производительности.

Установка и обновление. GMS весьма полезен при включении в инфраструктуру новых узлов или подключении нового сегмента, а также вычислении оптимальной конфигурации сегмента для решения определенных задач. Другая обязанность GMS - контроль версий его ресурсов и осуществление обновления на всех узлах инфраструктуры. Это решит проблему существования различных версий одного и того же сервиса.

Глобальный контроль. Использование GMS позволит получать полную информацию о состоянии, производительности, конфигурации и настройках грид-инфраструктуры, что создает возможность управления отдельными компонентами и всей инфраструктурой в целом.

GMS основан на сервис ориентированной архитектуре, т.е. имеет ту же природу, что и сервисы, которыми он призван управлять. Полная интеграция в ППО позволит GMS воздействовать на другие сервисы

просто и эффективно, что способствует решению проблемы адаптивности глобальных распределенных систем.

Вряд ли стоит ожидать появления подобного сервиса в ближайшее время, возможно, потребуется несколько лет, чтобы приблизиться к его реализации. Создание GMS в рамках gLite или Globus – это в первую очередь административная задача, которая потребует серьезных организационных решений и скоординированных действий разработчиков всех грид-сервисов. Однако автоматизация отдельных сервисов, а так же отработка различных методов и средств может происходить и локально. В качестве первого шага на пути автоматизации работы операторов сервиса передачи файлов и апробации идей GMS автором был разработан прототип экспертной системы, интегрированный в систему мониторинга. Прототип призван продемонстрировать возможности автоматизации сервиса передачи данных. Была спроектирована база знаний, которая включает более 70 базовых правил, сформулированных экспертами. Прототип позволяет интерпретировать состояния объектов системы в понятных пользователям терминах, объяснять, почему объекты находятся в данном состоянии, и предоставлять рекомендации по исправлению сложившейся ситуации.

Работа экспертов по заполнению базы знаний сводится к трем этапам:

1. Задание правил для всевозможных объектов. Базовые правила задаются для определенного класса объектов (каналы, виртуальные организации, сайты, хосты), в виде отношений (больше, меньше, равно и т.д.) к определенному значению.
2. Определение состояний, которые характеризуются тем или иным набором правил.
3. Определение действий при различных состояниях объектов.

Подобный механизм, позволяет перекрывать множество различных объектов и их характеристик, а также использовать доступные для понимания описания состояний объектов и действий, необходимых для их изменения.

Результатом работы системы является список объектов, их состояний и рекомендуемых действий, а так же таблица истории изменений состояния объектов. Благодаря такому подходу можно без труда не только определить текущие неисправности, но и выполнить определенные шаги для их устранения.

Дальнейшие исследования в данной области и развитие прототипа способны привести к частичной, либо полной автоматизации работы системы передачи данных.

В заключении сформулированы основные результаты диссертационной работы.

1. Выделены основные объекты, характеристики, состояния, схемы взаимодействия, принципы функционирования и архитектурные особенности сервиса передачи файлов, так же определены наиболее перспективные методы и подходы в аспекте мониторинга сервисов передачи файлов.
2. Предложена классификация ошибок, возникающих в распределенных системах передачи данных, основанная на четкой иерархии описания сбоев и использовании механизма паттернов.
3. Предложен и реализован подход к проектированию систем мониторинга сервисов передачи файлов в крупных распределенных грид-инфраструктурах;
4. Создана система мониторинга сервиса передачи файлов, предоставляющая широкие возможности для анализа и обработки информации о функционировании сервиса.
5. Проведено исследование ошибок, возникающих при передаче файлов в крупных распределенных грид-инфраструктурах. Исследование позволило определить самые распространенные ошибки, причины их возникновения, основные области возникновения ошибок, выявить программные и логические ошибки в приложениях.
6. Предложен новый сервис – сервис управления грид (Grid Management Service, GMS), который решает проблему адаптивности глобальных распределенных систем. Определены основные принципы функционирования, архитектурные особенности и задачи этого сервиса. С целью автоматизации работы сервиса передачи файлов и апробации идей GMS создан прототип экспертной системы, интегрированный в систему мониторинга.

Основные результаты диссертации опубликованы в работах:

1. Кореньков В.В., Ужинский А.В. На пути к адаптивному grid // Открытые системы. - 2009.- N9 – С.18-19.
2. Кореньков В.В., Ужинский А.В. Система мониторинга сервиса передачи данных (FTS) проекта EGEE/WLCG // Вычислительные методы и программирование. - 2009.- Т.10 – С.96-100.
3. Кореньков В.В., Ужинский А.В. История развития технологии мониторинга информационных систем // Системный анализ в науке и образовании. – Дубна, 2009.- Вып. 1.
4. Кореньков В.В., Ужинский А.В. Архитектура сервиса передачи данных в grid // Открытые системы. - 2008.- N2 – С.52-56.

5. Кореньков В.В, Ужинский А.В. Monitoring system of the file transfer service for EGEE/WLCG // Материалы международной конференции "Distributed computing and Grid technologies in science and education, GRID-2008". – Дубна, 2008.- ISBN 5-9530-0183-5. С.197-201.
6. Кореньков В.В., Ужинский А.В. Система мониторинга сервиса передачи данных (FTS) проекта EGEE/WLCG // Сообщения ОИЯИ. – Дубна, 2008.- P11-2008-80 .
7. Кореньков В.В, Ужинский А.В. Статистический анализ сбоев, возникающих при пересылке данных в глобальной грид-инфраструктуре EGEE/WLCG // Сообщения ОИЯИ. – Дубна, 2008.- P11-2008-82.

Получено 30 марта 2010 г.

Отпечатано методом прямого репродуцирования
с оригинала, предоставленного автором.

Подписано в печать 02.04.2010.

Формат 60 × 90/16. Бумага офсетная. Печать офсетная.

Усл. печ. л. 1,43. Уч.-изд. л. 1,65. Тираж 100 экз. Заказ № 56950.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.

E-mail: publish@jinr.ru

www.jinr.ru/publish/