УДАЛЕННЫЙ ДОСТУП К ВЫЧИСЛИТЕЛЬНОМУ КЛАСТЕРУ ВЦ ДВО РАН

В. В. Пересветов, А. Ю. Сапронов,
А. Г. Тарасов, Т. С. Шаповалов
Вычислительный центр ДВО РАН, Хабаровск, Россия
e-mail: peresv@as.khb.ru, sapr@as.khb.ru, taleks@as.khb.ru

The remote access to the CC FEB RAS computing cluster resources is described. It consists of web-interfaces for cluster users and administrator, a monitoring system with additional features and a security support system.

Введение

Эффективность использования вычислительного кластера в значительной мере зависит от средств удаленного доступа к его ресурсам. Созданный в Вычислительном центре ДВО РАН вычислительный кластер имеет доступ к сети Интернет и сайт информационной поддержки `http://cluster.as.khb.ru`. На кластере используется свободно распространяемое программное обеспечение (ПО), операционная система — Linux. В работе [1] описаны архитектура, конструкция, ПО и результаты тестирования производительности вычислительного кластера. При создании кластера к программному обеспечению предъявлялись следующие требования: соответствие общепринятым стандартам в области информационных технологий, эффективность взаимодействия пользователей и администратора с вычислительным комплексом, безопасность и надежность.

Для повышения комфортности работы с вычислительным кластером из сети Интернет создан web-интерфейс на языке PHP. Пользователям доступны следующие операции: изменение сведений своей учетной записи, сбор статистики. Администратору кластера предоставлены дополнительные функции. Для эффективной работы с вычислительным кластером необходимы не только стандартные средства мониторинга кластера, но и инструменты управления как информационным содержанием мониторинга, так и вычислительным процессом. Для этого создана система мониторинга расширенной функциональности. Предложена трехуровневая архитектура системы мониторинга, обладающая такими преимуществами, как: имеется возможность расширения существующих систем мониторинга без вмешательства в их работу; допустима гибкая система уведомлений и на ее основе система автоматизированного исправления ошибок; предоставлена возможность сторонним разработчикам расширять функциональность каждого уровня, не перерабатывая исходный код других уровней; допустимость создания иерархической системы монито-

© Институт вычислительных технологий Сибирского отделения Российской академии наук, 2006.

ринга для оптимизации трафика в сети. Необходимой составляющей системы удаленного доступа является обеспечение его безопасности, которая поддерживается средствами web-сервера Apache и системы SSL.

1. Web-интерфейс

На вычислительном кластере используется система диспетчеризации задач OpenPBS, которая предоставляет пользователю средства управления задачами и получения статистической информации, эти команды доступны удаленно по ssh. Однако для обычных пользователей кластера уровень дружественности интерфейса, предоставляемого по данному протоколу, удобство представления статистических и других данных в настоящее время не являются достаточными.

На вычислительном кластере система диспетчеризации OpenPBS сохраняет данные в таблицах свободно распространяемой СУБД MySQL, что позволяет осуществлять гибкие запросы на выборку хранимой информации по задачам. Данная возможность выборки использована при создании web-интерфейса к статистике пользовательских задач. Пользователь кластера, имеющий учетную запись, получает возможность в удобном и привычном виде просматривать следующие статистические данные по его задачам: время старта и завершения расчета задачи, полное время выполнения, используемая виртуальная память, затраченная физическая память, используемые ресурсы процессора, принадлежность определенной очереди задач, имя задачи и некоторые другие. Можно использовать фильтр задач по дате начала и конца выполнения, типу очереди и другим параметрам. Также имеется возможность сортировки выбранных задач и выбора количества отображаемых на странице данных. Статистика представляется в форме таблиц.

Помимо статистических сведений о задачах пользователя в таблицах базы данных хранятся личные данные пользователей. Через web-интерфейс пользователь имеет возможность самостоятельно исправить или добавить личные данные, что упрощает сбор информации о пользователях администратором. Такие данные могут быть использованы, например, для контакта с пользователем. Изначально личную информацию пользователь должен самостоятельно внести при регистрации своей учетной записи. Для этого на кластере создана специальная web-страница регистрации нового пользователя с заполняемыми полями. Предусмотрена проверка корректности заполнения данных с указанием типа ошибки, если таковая обнаружена. В случае корректности данных они заносятся во временную таблицу базы данных и сохраняются там до тех пор, пока администратор не примет решения о создании учетной записи для данного пользователя или отказе в регистрации.

Администратору кластера предоставляется свой интерфейс, в котором помимо управления содержимым web-сайта имеется интерфейс к очереди желающих получить новую учетную запись, а также к статистике по задачам всех пользователей и их личным данным. Очередь кандидатов на получение новой учетной записи представляется администратору в виде таблицы, в которую помимо имени и фамилии выводится краткая информация о пользователе. При регистрации личные данные из временной таблицы базы данных переносятся в постоянную и запускается программа, написанная на интерпретируемом shell-скрипте. Данная программа создает новую учетную запись на кластере, а также генерирует сертификат для доступа к статистике пользователя через web-интерфейс. Подобным образом администратор может удалить пользователя из очереди. Страница статистики

администратора имеет дополнительную возможность выбора пользователя, по которому нужно получить статистику, что удобно для проведения анализа активности пользователей.

2. Система мониторинга кластера расширенной функциональности

Мониторинг вычислительного кластера подразумевает наличие набора функций, предоставляемых для получения детальной и достоверной информации о текущем состоянии вычислительного комплекса, а также о процессах изменения критически важных системных характеристик в течение определенного периода (периода выполнения поставленной задачи, периода работы кластера). Важна также способность системы реагировать на изменения, главным образом негативные, в работе вычислительного комплекса, т. е. необходимо организовать взаимодействие обслуживающего персонала с системой посредством уведомлений об определенных событиях. Основными требованиями, предъявляемыми к системам мониторинга, являются: минимальное потребление ресурсов контролируемых узлов, надежность, переносимость (в смысле платформенной независимости), доступ к текущим данным и данным за период времени, простота развертывания и масштабируемость.

Системы мониторинга разбиты на два уровня функционирования, как и другие существующие системы мониторинга, например Ganglia (`www.ganglia.info`), Nagios (`www.nagios.org`). На программном уровне это реализуется созданием фоновых процессов на контролируемом узле и основного серверного процесса на управляющем узле.

Первый уровень представлен платформенно зависимыми программными модулями (в целях уменьшения потребляемых ресурсов), они устанавливаются непосредственно на узлах и обеспечивают поставку необходимых данных в форме метрик. На втором уровне выполняются сбор информации и представление ее в удобном для пользователя виде, в некоторых системах на этом уровне выполняется анализ данных различной степени сложности. Этот уровень в зависимости от реализации может быть платформенно зависимым (написан на компилируемом языке программирования) и не зависимым от используемой системы (написан на интерпретируемом языке программирования), а также представлен их комбинацией. Такой подход удобен для реализации клиент-серверной архитектуры, в которой сервер занимается в основном хранением данных. Двухуровневая архитектура имеет как преимущества, так и недостатки, среди которых отметим нечеткое разделение задач различных уровней, что затрудняет дальнейшее совершенствование системы мониторинга.

В данной работе предлагается трехуровневая архитектура организации системы мониторинга с использованием на высоких уровнях платформенно независимых компонентов, обеспечивающих сбор, предварительный анализ данных и механизм отклика. Предлагаемая система мониторинга позволяет создавать надстройки над уже существующими системами мониторинга, поскольку может использовать источники данных уже развернутых систем. Расширение функциональности уже установленной на вычислительном комплексе системы мониторинга достигается без ущерба и коренных изменений в ее работе.

На первом (низком уровне) располагаются источники данных (сенсоры). В целях эффективного использования оперативной памяти и процессора сенсоры реализуются на машинно зависимых языках программирования, однако это не исключает при необходимости использование интерпретируемых языков программирования.

На втором (промежуточном уровне) располагается машинно независимый программный модуль, обеспечивающий получение и хранение данных. Соединение посредством локальной вычислительной сети (или к файлу на локальном диске, базе данных и т. д.) реализует доступ к первому уровню. Полученные данные формируют кадры (наборы логически связанных данных), зависящие от используемой системы мониторинга. На данный момент не существует возможности автоматического выбора системой типа данных кадра, поэтому пользователь сам должен указывать формат данных. Для получения кадра клиентское приложение инициирует запрос на получение данных. Соединение передает запрос источнику данных для получения конкретных данных, указанных пользователем. Данные кадра состоят из набора значений нескольких метрик, в общем случае за некоторый период времени. На значения метрик могут быть наложены простые условия (триггеры), выполнение которых влечет за собой запуск определенных процедур (уведомление пользователя, выполнение указанных программ). Триггеры данного уровня являются достаточно простыми операциями, они проводят сравнение: больше, меньше, равно и т. д. Полученные данные сохраняются в одном из предлагаемых хранилищ данных посредством указанного интерфейса кэширования. Употребление данного термина связано с тем, что обычно пользователей интересуют данные за сравнительно небольшой промежуток времени, остальные данные сбрасываются из оперативной памяти в хранилище данных.

Третий уровень (высокий уровень) обеспечивает визуализацию данных, их детальный анализ. Этот уровень может реализовываться как с помощью уже существующего ПО с сохранением данных в общепринятом формате (например, в базе RRD, XML-документах), так и с помощью специально созданного под вышеуказанную архитектуру. Данный уровень является в общем случае клиентским по отношению к вычислительному комплексу. Так как ПО не выполняется непосредственно на вычислительных узлах, время построения различных диаграмм и схем не имеет критического значения, что дает возможность использовать существенно более развитое ПО для представления и анализа данных.

Механизм триггеров позволяет в дальнейшем создать систему уведомления пользователя о неэффективном использовании вычислительных ресурсов (потребляемой мощности узла менее какой-либо величины), указать причины с анализом использования дисковой и сетевой подсистемы и выдать рекомендации. Подобная система сможет также при восстановимом отказе оборудования управлять миграцией задач (если такая возможность поддерживается вычислительным комплексом) на другой узел или корректно завершить задачу пользователя и освободить вычислительные узлы для повторного запуска задачи.

Описанная система мониторинга частично реализована с использованием языка программирования Java на промежуточном и высоком уровнях. Создано приложение Grated, собирающее данные (по умолчанию, посредством подключения к мультикастовому каналу, используемому Ganglia) и проверяющее задаваемые триггеры. Высокому уровню данные предоставляются по протоколу TCP/IP в формате XML. Модульная реализация позволяет настроить приложение на иной формат входных и выходных данных.

Высокоуровневая часть реализована также на языке Java в форме отдельного приложения Grate и аплета Grape для доступа через web-интерфейс (`http://cluster.as.khb.ru//grate`). Она используется для визуализации и хранения данных. В данных приложениях также существует возможность использовать триггеры. Программный модуль grate способен функционировать без использования Grated, обращаясь к высокоуровневым сервисам Ganglia (gmetad) для получения необходимых данных.

Анализ производительности данной системы мониторинга показал, что она по потреблению ресурсов находится на том же уровне, что и система, на базе которой она строилась.

Результаты испытаний системы мониторинга

Выполняемое приложение	Кластер, 8 узлов		Эмуляция, 100 узлов	
	T_o, ч	T_p, %	T_o, ч	T_p, %
grated	768	0.14	50.8	9.19
gmetad	888	0.23	192	10.6

Следует однако отметить большие затраты памяти, связанные с загрузкой в оперативную память помимо собственно программы и ее данных еще и виртуальной машины Java. Тестирование Grated проводилось на сервере локальной вычислительной сети кластера ВЦ ДВО РАН под управлением WhiteBox Linux 3.0 2.4.20-8. В качестве системы для сравнения выступала Ganglia 2.4 (модуль gmetad). Результаты приведены в таблице, где показаны значения общего времени выполнения T_o и относительное время загрузки процессора T_p. Следует учесть, что Grated не сохраняет значения в локальную базу данных, в то время как gmetad сохраняет в RRD. Однако Grated проводит проверку триггеров для показателей температуры центральных процессоров. Использовалась Sun JRE 1.5. В таблице также представлены результаты испытаний при эмуляции на выделенной рабочей станции источников данных для ста вычислительных узлов по методике, примененной создателями Ganglia для тестирования своей системы. Использовалась Sun JRE 1.4.

Приложение, обеспечивающее визуализацию данных, запускалось на программных платформах Linux и Windows98/XP. Использовались несколько дистрибутивов Linux (WhiteBox, Fedora Core 4, Fedora Core 3, SuSE 9.1) с установленными Sun JRE 1.4, Sun JRE 1.5. Во всех случаях использование системы не вызвало программных сбоев.

3. Безопасность удаленного доступа к кластеру

Доступ к ресурсам кластера разрешен пользователям посредством ряда протоколов через всемирную сеть Интернет. При передаче информации через сегменты глобальной сети существует опасность ее перехвата, подлога, взлома и т. п. Более того, на управляющем узле кластера ВЦ ДВО РАН (далее сервера) ежедневно регистрируются попытки взлома. Некоторые локальные пользователи, например студенты, выполняющие лабораторные работы на кластере, также могут быть источником преднамеренных или непреднамеренных вторжений.

При организации коллективного доступа к кластеру необходимо решить ряд вопросов информационной безопасности: разграничение привилегий для категорий пользователей; взаимная идентификация сервера и клиента; шифрование передаваемых данных; умеренное увеличение сложности доступа к защищенным ресурсам и их администрирования.

Управление группами пользователей на кластере осуществляется при помощи стандартных средств операционной системы Linux, которая позволяет организовывать работу с измененным корнем файловой системы — так называемым chroot-окружением. Ограничение различных типов ресурсов для пользователей осуществляется с помощью механизма Pluggable Authorization Modules. Возможность ограничения таких важных ресурсов, как процессорное время, размер различных сегментов оперативной памяти, число процессов, количество открытых файлов и ряд других, позволяет избежать атак типа “отказ в обслуживании”, иначе DOS-атак (Deny of Service). Для уменьшения вероятности успешных атак такого типа используется квотирование дискового пространства. Восстановление системы или отдельных файлов обеспечивается инкрементным резервным копированием системы.

Идентификация объекта, запрашивающего или отдающего определенный ресурс, является ключевой задачей при предоставлении удаленного доступа. Существует три способа для управления и наблюдения за ресурсами сервера: web-интерфейс, java-программа (апплет), работающая по протоколу TCP, и терминальный доступ посредством SSH.

Доступ к web-интерфейсу пользователь осуществляет при помощи браузера, установленного на клиентской машине. Передача данных от web-сервера возможна по двум протоколам — HTTP и HTTPS. В первом случае информация передается в открытом виде, во втором — в шифрованном. Шифрование на стороне сервера осуществляется при помощи криптографической подсистемы OpenSSL, являющейся открытой реализацией стандарта Socket Secure Layer (SSL). Эта система — центральное звено в обеспечении информационной безопасности критически важных компонентов операционной системы Linux: web- и ftp-сервера, SSH, VNC, e-mail и др. Большая часть ПО в операционной системе Linux, использующая криптографические алгоритмы, так или иначе связана с OpenSSL. OpenSSL представляет собой набор библиотек и программ. Использование OpenSSL в качестве системы шифрования с открытым ключом — одна из основных возможностей системы. Применительно к web-серверу шифрование и двухсторонняя клиент-серверная идентификация производятся при помощи сертификатов формата X.509 v.3 и v.4. Такой сертификат — это открытый ключ, который привязан к определенному адресу в нотации имен службы каталогов X.500.

При первом запросе зашифрованной информации пользователю будет предложено принять сертификат с сервера. В случае, если сертификат подписан электронной подписью (распространяемой также в виде сертификатов) доверенного центра сертификации (по умолчанию ряд таких сертификатов от организаций типа VeriSign, AOL, Visa включается в браузеры), то браузер примет это соединение без запроса и установит доверительные отношения с сервером. Сертификация в таких центрах является платной процедурой. Кроме цены в вопросах сертификации (и криптографии) немаловажную роль играют организационный и юридический факторы. Для небольшой группы пользователей оправданным решением будет создание своего собственного центра сертификации (ЦС). В его рамках создается корневой сертификат (при необходимости цепочка доверенных сертификатов), с помощью которого подписываются сертификаты web- и ftp-серверов. Сертификат созданного ЦС пользователь должен получить по безопасному каналу связи (прямое модемное соединение, виртуальная частная сеть, электронная почта с электронной подписью, “из рук в руки” на каком-либо носителе) и экспортировать в браузер. После этого в браузере пользователь может однозначно идентифицировать сервер и установить защищенное соединение. По установленному зашифрованному каналу можно безопасно передавать применяемые в web-технологиях стандартные пароли для идентификации пользователя в целях разграничения ресурсов.

Для идентификации пользователя сервером можно также воспользоваться механизмом сертификатов. Пользователь (или администратор ЦС по запросу пользователя) должен создать запрос на сертификацию в стандартном формате. По этому запросу пользователю будет предоставлен сертификат, подписанный корневым сертификатом ЦС. Экспорт этого сертификата в браузер через безопасный канал позволит web-серверу производить авторизацию пользователя при доступе к строго определенным частям web-сайта.

Использование java-программы (апплета) для доступа к серверу вписывается в показанную выше схему клиент-серверного взаимодействия. С версии Java Development Kit 1.1 (JDK) существует возможность подписывания java-кода. Доступ к серверу осуществляется через web-браузер, но манипуляции с принятыми сертификатами осуществляет java-машина, установленная на компьютере пользователя.

Запрос пользователем сертификата кроме удобства работы с web-сервером позволяет применять этот же сертификат при обмене почтовыми электронными сообщениями с администратором кластера. Это обеспечивает передачу конфиденциальной информации (изменения в адресах, именах и паролях доступа) без применения дополнительных средств шифрования.

Недостатком системы сертификатов является невозможность их использования в терминальных SSH сессиях. Клиент и сервер системы OpenSSH в дистрибутивах операционной системы Linux не поддерживают аутентификацию при помощи сертификатов форматов X.509, хотя и используют аналогичную схему с применением алгоритмов с открытыми ключами. Преодолеть это препятствие можно использованием ряда патчей (модификаций исходного кода независимыми разработчиками), имеющихся в сети Интернет, однако каждый из этих патчей не решает всех вопросов, связанных с авторизацией пользователей. В дальнейшем для "прозрачного" доступа пользователей при помощи одного лишь сертификата ко всем сервисам сервера необходимо внести ряд изменений в исходный код системы.

Список литературы

[1] Пересветов В.В., Сапронов А.Ю., Тарасов А.Г. Вычислительный кластер бездисковых рабочих станций. Хабаровск, 2005 (Препр. РАН. ДВ отд-ние. ВЦ, № 83).

Поступила в редакцию 26 сентября 2006 г.