

СИСТЕМА WBS КАК РАСШИРЕННЫЙ ИНСТРУМЕНТ УПРАВЛЕНИЯ ВЫЧИСЛИТЕЛЬНОЙ СРЕДОЙ

Д.С. Одякова, Г.В. Тарасов, Д.И. Харитонов

Институт автоматики и процессов управления ДВО РАН, Владивосток

Рассмотрены принципы функционирования системы WBS (Web Batch System), разработанной в ИАПУ ДВО РАН. Основной функцией системы WBS является организация прохождения заданий в многопроцессорной вычислительной среде. Благодаря архитектурным особенностям и формальному подходу к формированию заданий, наряду с заданиями пользователей система позволяет выполнять административные задания по тестированию частных характеристик вычислительных узлов, а также по проверке работоспособности вычислительной среды.

Введение

К настоящему времени сформировался образ среды для научных расчётов, удовлетворяющий следующим требованиям: во-первых, это многопроцессорная система, реализованная в большинстве случаев в виде кластера или GRID-среды; во-вторых, это длительный цикл эксплуатации и отказоустойчивость; и в-третьих, предоставление специфического сервиса доступа пользователей к ресурсам и разделения ресурсов, включая запуск задач, требующих специального программного обеспечения. Со временем каждая вычислительная среда для научных расчётов обрастает своими пользователями. Пользователи формируют перечень прикладного программного обеспечения (ПО), которое должно поддерживаться вычислительной средой. Однако существует ряд факторов, препятствующих сохранению работоспособности ПО в вычислительной среде в течении длительного интервала времени. В современном мире согласно закону Мура, как минимум раз в пять лет приходится заниматься обновлением оборудования, что влечёт за собой как минимум обновление низкоуровневых драйверов операционной системы. Новым пользователям требуется установка нового ПО. В последнее время актуальной является установка критических обновлений системы, связанных с безопасностью. Любая из трёх вышеперечисленных причин может привести к необходимости обновления всего программного стека, после которого естественным образом возникает проблема проверки работоспособности как системы в целом, так и всего перечня прикладного программного обеспечения пользователей. К сожалению, невозможно проверить свойства системы, требуемые для различного ПО, исходя из списка установленных файлов. Альтернативным путем является использование множества тестов, проверяющих частные требования. Тестирование в распределённой гетерогенной среде является трудоёмким процессом, поэтому предлагается автоматизировать процесс запуска тестов и сбора информации о результатах тестирования, необходимых для проверки характеристик вычислительной среды.

В рамках системы WBS (Web Batch System), разрабатываемой в ИАПУ ДВО РАН, цикл задач, связанных с обслуживанием вычислительной среды, может быть представлен образом, идентичным вычислительным задачам пользователей. Первичный запуск административной задачи в системе сохраняется в виде шаблона в базе данных, шаблон запуска задачи параметризуется, ему назначаются специфические свойства приоритетности и прав доступа. Готовый шаблон может быть использован в процессе автоматизи-

зированной управления вычислительной средой. Далее рассматривается общая архитектура системы WBS, основной аппарат формирования заданий в системе, а также применение системы для управления различными заданиями в распределённой вычислительной среде.

1. Архитектура WBS

Архитектура WBS состоит из 6 компонент, представленных на рис. 1. Компоненты делятся по зонам доступа: на внешний интерфейс пользователя, распределённая система управления на основе SQL-сервера и NFS-сети, а также внутренний интерфейс доступа к вычислительным ресурсам.

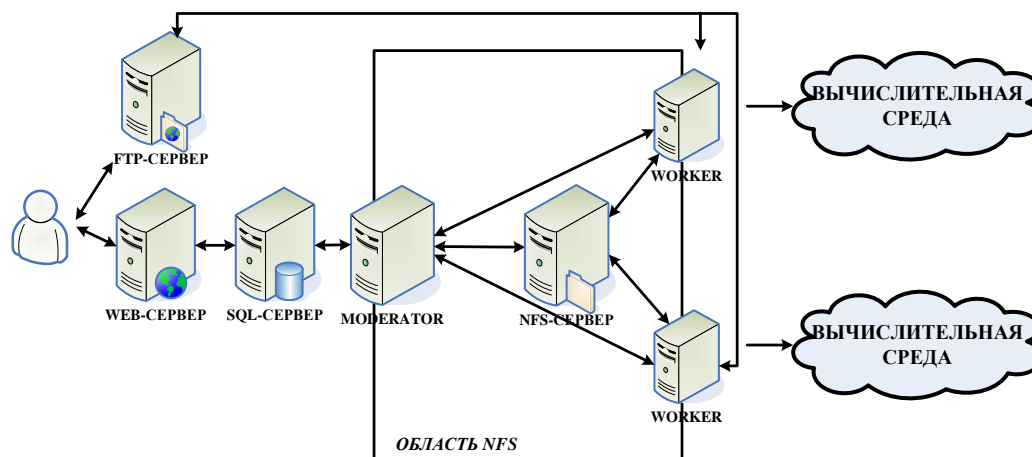


Рис. 1. Архитектура WBS

Web-сервер предоставляет основной интерфейс пользователям к системе WBS через интернет-браузер. Он реализован на основе www-сервера Apache с использованием языков PHP и JavaScript, а также библиотеки jQuery. Позволяет пользователям и администраторам системы управлять заданиями (создавать, удалять, редактировать, проверять статус) через интернет с помощью веб-браузера.

SQL-сервер (EnterpriseDB или PostgreSQL) является ключевым компонентом системы управления, который хранит информацию о пользователях, задачах, тестах, конфигурациях вычислительной системы и формирует файлы заданий. Все задания, выполняемые системой WBS, начинаются и заканчиваются на SQL-сервере, который при необходимости обращается к МОДЕРАТОР'у с требованием обновить список заданий, нуждающихся в выполнении.

МОДЕРАТОР – это часть системы управления, он выполняет связующую функцию между SQL и управляющими узлами вычислительной системы, является программой-демоном, работающей в многопоточном режиме. МОДЕРАТОР выполняет следующие основные функции: забирает задания пользователей из SQL, кладёт файлы заданий в соответствующие директории на NFS-сервере, посылает сообщения, сигнализирующие о поступлении нового задания, соответствующим WORKER'ам, дожидается сообщений о выполнении заданий от WORKER'ов, вносит на SQL-сервер информацию о завершении задач.

WORKER'ы отвечают за взаимодействие с вычислительными ресурсами, являются программами-демонами, работающими в многопоточном режиме. В силу своей специфики компоненты WORKER должны располагаться во внутренней сети вычислительных ресурсов, причем, количество WORKER'ов может варьироваться в зависимости от способа запуска заданий (GRAM, OpenPBS, SLURM и т.д.). После того как WORKER получил задание, всё, что ему необходимо выполнить, описано специальным образом в файле-инструкции. Описанные инструкции WORKER выполняет последовательно, по-

сле того как задание выполнено, отправляет сообщение MODERATOR'у. При необходимости WORKER имеет возможность обращаться к FTP-серверу, чтобы класть или забирать файлы с данными.

NFS-сервер является буфером обмена данными между MODERATOR'ом и WORKER'ами.

FTP-сервер используется в качестве хранилища больших входных и выходных файлов с данными задач пользователей.

2. Генерация скриптов (шаблонов)

Основная идея подхода, использованного при создании WBS, состоит в том, что любые операции пользователей и администраторов вычислительной среды можно представить в виде задачи. Задача проходит определенный жизненный цикл: создание (CREATING), редактирование (EDITING), ожидающая запуска (PENDING), выполнение (RUNNING), окончание (DONE). До начала запуска каждая задача в системе WBS представляется посредством шаблонов и набора параметров к ним. Шаблон описан в виде последовательности параметризуемых действий, необходимых для выполнения задачи. На основе шаблона генерируются файлы, необходимые для выполнения задания на указанном вычислительном ресурсе. Файлы прикладных задач могут быть нескольких типов.

Файл-инструкция для WORKER'а, представляет собой xml-подобный файл, в котором последовательно перечислены инструкции. В настоящее время используется четыре типа инструкций. Во-первых, инструкции обращения к командному интерпретатору (command). Во-вторых, указание WORKER'у временно приостановить выполнение инструкций (wait) и ожидать сигнала продолжения. В-третьих, указание разобрать заданный текстовый файл (parse) и заменить переменные, маркированные специальным образом, на значения, определённые для конкретного WORKER'а. В четвёртых, указание для копирования файлов данных задачи из и на FTP-сервер.

Файл паспорта задачи, относящийся непосредственно к запуску задачи на выполнение, может отличаться в зависимости от типа системы, на которой выполняется задание – OpenPBS, SLURM, GRAM и т.д.

Третий тип – это специфические файлы настроек задачи, представленные в текстовом виде и расположенные в SQL.

Четвёртый тип – это большие файлы задачи (файлы с выходными или входными данными), которые хранятся на FTP-сервере. Входные файлы к моменту запуска задачи пользователь уже должен разместить на FTP-сервере в директорию задачи и указать список файлов в настройках запуска.

Файлы первых трёх типов генерируются на SQL-сервере с использованием следующей универсальной процедуры. Шаблон файлов задания представляется в виде графа, каждый узел которого имеет собственные параметры и множество инцидентных узлов. При заведении нового задания пользователь, редактируя параметры задачи в интерактивных страницах браузера, задаёт структурное дерево параметров файла, каждому узлу которого ставится в соответствие узел шаблонного графа параметров файла. Обход этого дерева происходит сверху вниз (вглубь) и слева на право (в ширину). При заходе и выходе из каждого узла дерева вызываются скрипты формирования файла, определённые администратором для соответствующего узла шаблонного графа параметров файла. Эти скрипты сохраняют в файле задания текстовое представление структуры узла шаблона. Для создания нового шаблона можно либо скопировать, либо наследовать свойства уже существующего (родительского) шаблона. Создание шаблонов задач осуществляется администраторами системы WBS.

3. Область применения WBS

Благодаря тому, что задачи описаны в SQL с помощью шаблонов, а остальные компоненты системы WBS являются исполнителями с чётко определённой функциональностью, область применения WBS может быть достаточно широкой. В настоящее время в суперкомпьютерном центре ИАПУ ДВО РАН WBS используется в качестве 1) системы управления заданиями пользователей на вычислительных ресурсах центра; 2) инструмента тестирования частных характеристик вычислительной среды и её отдельных компонент; 3) расширенного инструмента для администрирования гетерогенной и проверки целостности вычислительной среды.

При управлении заданиями пользователей на SQL-сервере формируется множество шаблонов заданий. Используя эти шаблоны, пользователь имеет возможность создавать, редактировать, удалять, а также просматривать состояние своих задач. Манипулирование шаблонами осуществляется администраторами системы. Пользователи только применяют готовые шаблоны для создания своих задач, задавая значения параметров через веб-браузер. Процесс генерации файлов заданий, копирования их на вычислительный узел, запуска и получения результатов является полностью автоматическим. Пользователь может лишь наблюдать состояния задач и прерывать их исполнение при необходимости.

При тестировании характеристик вычислительной среды на SQL-сервере формируется множество шаблонов тестов заданий. Для совместимости теста с различным аппаратным и программным обеспечением вычислительных узлов исходные тексты тестов располагаются в сетевой директории, доступной с каждого из узлов вычислительной системы. Каждый тест располагается в собственной директории и имеет единственный исполняемый скриптовый файл, совпадающий с названием директории. При запуске тестового задания исходные тексты копируются на тестовый узел. Тест компилируется и запускается. В процессе выполнения теста вывод сохраняется на SQL-сервере, что позволяет администраторам собирать статистическую информацию о системе. В частности можно получить информацию о последнем тестировании, о тестировании на заданное число, график изменения тестируемых характеристик (что является полезным при настройке операционной системы и прикладных библиотек вычислительных узлов).

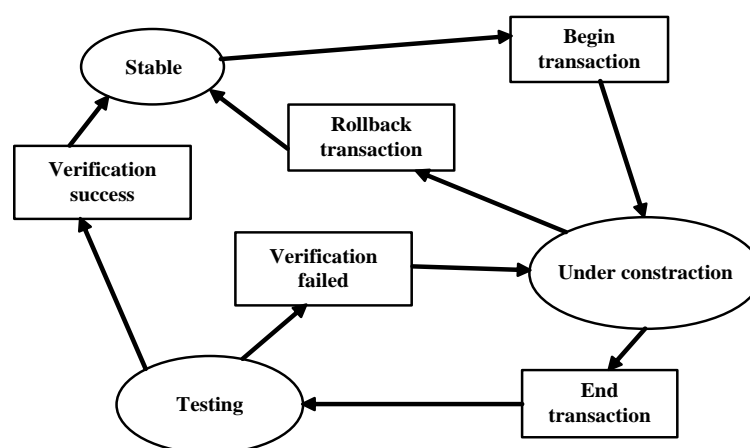


Рис. 2. Диаграмма состояний вычислительных узлов.

При администрировании вычислительной среды необходимо: контролировать установку программного обеспечения на каждом узле программной среды; предотвращать несанкционированную установку программного обеспечения; проверять работоспособность программного обеспечения; хранить историю изменений системы и отоб-

ражать текущее состояние системы. Для выполнения поставленных задач в ИАПУ ДВО РАН разрабатывается программная система OSeye, использующая возможности WBS для запуска задач администрирования и тестирования вычислительных узлов.

При администрировании вычислительной системы каждому узлу назначается одно из трёх состояний (рис. 2): STABLE – в этом состоянии узел может использоваться в обычном штатном цикле, установка нового ПО заблокирована; UNDER CONSTRUCTION – в этом состоянии на узле могут проводиться манипуляции с установленным ПО, а также установка нового ПО; TESTING – в этом состоянии в автоматическом режиме запускаются тесты из предписанного набора. В состоянии UNDER CONSTRUCTION администратор производит изменения в системе, после которых необходимо убедиться, что система находится в работоспособном состоянии, под которым понимается возможность компиляции и запуска распределённых задач пользователей. Основная задача OSeye сводится к переводу из состояния UNDER CONSTRUCTION в STABLE, условием которого является успешная проверка работоспособности системы.

Для обеспечения работоспособности распределённой среды необходимо контролировать большое количество существенно отличных друг от друга параметров, таких как: функционирование Ethernet, Myrinet, Infiniband сетей, наличие и корректность настройки SSH протокола, наличие языков программирования и библиотек, наличие и доступность прикладных программных пакетов и библиотек, корректность функционирования системы управления прохождением заданий. Для контроля работоспособности программного обеспечения используется процесс тестирования, который позволяет абстрагироваться от понятий различных предметных областей в пользу унифицированного интерфейса тестирования системы.

Заключение

В настоящей статье рассмотрена архитектура системы WBS, неформально описан способ генерации файлов заданий на основе шаблонов в SQL, а также описаны примеры использования WBS в качестве расширенного инструмента управления вычислительной средой. Управление вычислительной средой реализовано через выполнение заданий WBS, при помощи которых может контролироваться работоспособность узлов системы и тестирование частных характеристик её компонентов.

Литература

1. Голенков Е.А., Левин В.А., Харитонов Д.И., Шиян Д.С. Организация поддержки циклических расчетов в GRID-сегменте ДВО РАН. - Горный информационно-аналитический бюллетень (научно-технический журнал) Mining informational and analytical bulletin (scientific and technical journal). 2009. Т. 18. № 12. С. 225–229.
2. Бабяк П.В., Тарасов Г.В. Организация потоковой обработки спутниковых данных на основе GRID-технологий // Горный информационно-аналитический бюллетень (научно-технический журнал) Mining informational and analytical bulletin (scientific and technical journal). 2009. Т. 18. № 12. С. 170–175.