

С.В. Минухин

Харьковский национальный экономический университет имени Семена Кузнеця, Харьков

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ОБРАБОТКИ ЗАДАНИЙ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕДАХ

Рассмотрены информационные технологии для решения задач планирования заданий в двухуровневой вычислительной системе, базирующиеся на работе кластерной СУБД и технологии программных агентов.

Для реализации технологий программных агентов, устанавливаемых на узлах локальных ресурсов распределенной системы, выбраны компоненты систем мониторинга, обеспечивающие оперативность и полноту требуемой для принятия решений о выборе стратегий планирования информации.

Ключевые слова: *распределенная вычислительная система, вычислительный кластер, кластерная СУБД, информационная технология, система мониторинга, программные агенты.*

Введение

Важнейшей задачей повышения производительности и эффективности распределенных вычислительных систем (РВС) является своевременное получение и обработка данных о состоянии объектов контроля РВС менеджерами виртуальных организаций (ВО) и системными администраторами вычислительных кластеров для обоснования и принятия решений о выборе стратегий и политик планирования заданий. Обзор информационных технологий и систем, используемых в распределенных средах, приведен в [1], общий подход и информационные технологии планирования пакетов заданий на основе решения задачи о наименьшем покрытии рассмотрены в [2; 3]. Их особенностью является разработка модели предметной области, описывающей модель планирования заданий в двухуровневой РВС [1]. В данной работе рассмотрены особенности реализации этих технологий на основе параллельной СУБД в совокупности с технологией программных агентов.

Разработка информационной технологии планирования в двухуровневой вычислительной системе

Для разработки информационных технологий обработки информации в рассматриваемой модели планирования заданий [1] использованы:

архитектура, обеспечивающая параллельный доступ к данным БД о состоянии ресурсов, заданий, очередях на ресурсы и загрузке БД [4];

СУБД, поддерживающая технологии параллельного доступа к БД;

база данных состояния ресурсов, заданий, очередей заданий на ресурсы;

программное обеспечение, позволяющее удаленно получать информацию о состоянии локальных ресурсов и заданий на узлах вычислительных кластеров;

стандартные коммуникационные протоколы TCP/IP, SNMP, GridFTP;

программное обеспечение медиатора, обеспечивающего взаимодействие между поставщиками и потребителями информации.

В предлагаемой модели планирования поставщиками информации являются ресурсы и задания, потребителями – менеджер ВО (системный администратор локального ресурса РВС). В роли последних также могут выступать пользователи из состава ВО, представляющие свои задания на обработку в распределенную систему.

В качестве базиса архитектуры для информационной технологии получения и обработки информации о состоянии ресурсов и заданий выбрана ParGRES [5], являющаяся кластером БД промежуточного слоя архитектуры грид, которая при обработке запросов использует внутри- и межзапросный параллелизм, а также обеспечивает работу с репликациями. Параллелизм достигается за счет полной репликации БД и адаптивной виртуальной фрагментации (Adaptive Virtual Partitioning, AVP). ParGRES обеспечивает гибкость распределения узлов для обработки запросов: любые запросы могут быть обработаны любым набором узлов кластера. Система AVP обеспечивает динамическое распределение нагрузки между узлами кластера во время обработки запроса. Как и в других кластерных БД, ParGRES управляет параллельным выполнением запросов на основе экземпляров СУБД, установленных на узлах кластера БД [6].

В ParGRES используются глобальные и локальные компоненты, которые формируют двухуровневую организацию реализации предлагаемой модели планирования выполнения заданий:

глобальные компоненты – посредник и обработчик запросов кластера (Cluster Query Processor, CQP), которые выполняют задания, использующие несколько узлов кластера;

локальные компоненты – обработчик запросов узла (Node Query Processor, NQP) и СУБД (Database Management System, DBMS), которые выполняют задания на узле.

Посредник получает запросы от заданий (приложений), передает их обработчику запросов кластера и далее передает результаты обработки запросов кластера приложениям (пользователям).

Поскольку большинство из рассмотренных кластеров используют один доступный для внешних приложений управляющий узел, компонента посредник, как правило, размещается на этом же узле, то есть является централизованной и предоставляет возможность физического распределения обработки запросов кластера для каждого запроса, тем самым повышая общую доступность ресурсов вычислительной среды.

Для параллельной обработки запросов в ParGRES реализовано решение четырех типов задач [7]:

1. Анализ SQL-запросов. Процессор запросов кластера содержит синтаксический анализатор для выполнения команд SQL-клиентского приложения. Это позволяет определить:

множество отношений и атрибутов, на которые ссылается запрос, и которые могут быть использованы для обеспечения параллелизма внутри запроса; информацию, необходимую для выполнения композиции (объединения) запросов;

набор атрибутов, используемых в операциях агрегирования.

2. Обработка запросов с внутрizaпросным и межзапросным параллелизмом.

Внутрizaпросный параллелизм предполагает, что обработчик запросов кластера посылает запрос на узел с наименьшим количеством нерешенных заданий. Параллельно внутри запроса сложные запросы декомпозируются на подзапросы, которые будут выполняться параллельно к различным данным. При межзапросном параллелизме CQP посылает запрос к NQP узла с наименьшим числом нерешенных заданий. На основе внутрizaпросного параллелизма сложные запросы декомпозируются на подзапросы, которые будут выполняться параллельно к различным фрагментам данных. При этом CQP также переписывает исходный запрос в подзапросы.

3. Композиция (агрегация). В ParGRES получение результата – композиции – реализуется на основе двухэтапной агрегации. На первом этапе узлы агрегируются в группы, возвращаемые локальными подзапросами. На втором этапе группы распределяются на соответствующие узлы посредством хэш-функции. После этого каждый узел посылает подмножество его запросов для получения общего результата узлу-координатору, который выполняет их агрегацию.

4. Обновление. ParGRES, как правило, используется только для чтения обрабатываемых запросов,

что типично, например, для технологии OLAP, но в ней могут выполняться и обновления. Поскольку обновления в среде OLAP, как правило, происходят быстро и выполняются в определенное время, в ParGRES реализована активная политика целостности: не разрешается одновременное выполнение обновления и запросов. Для осуществления этой политики ParGRES использует планировщик, который упорядочивает запросы и обновления.

В информационной технологии для реализации модели планирования заданий предлагается использовать два уровня – уровень грид (GParGRES) и уровень узла (на базе СУБД PostgreSQL).

Такой выбор обусловлен архитектурными требованиями современных грид-систем. Сервисное программное обеспечение грид-системы и ее администратор получают возможность работы с абстракцией уровня стандартной реляционной БД, в качестве которой используется PostgreSQL. На этом уровне можно создать запрос к БД, используя стандартную технологию SQL-запросов.

Функциональность предлагаемого решения обеспечивается следующими службами [7]:

службой реестра (Registry Service, RS), которая включает информацию о GParGRES, например, о состоянии каждого фабричного сервиса и экземпляра службы распределенных запросов;

службой «Фабрика» (Fabric Service, FS), предназначенной для создания новых экземпляров службы распределенных запросов. Когда клиентское приложение представляет запросы GParGRES, оно задает сервис, чтобы создать новый экземпляр службы распределенных запросов (DQS). Каждый новый экземпляр получает уникальный идентификатор службы, которая ассоциирует ее с соответствующей фабрикой. Этот идентификатор не используется повторно для новых случаев даже тогда, когда служба завершается;

службой распределенных запросов (Distributed Query Service, DQS) – службой, непосредственно взаимодействующей с приложением клиента. Служба распределенных запросов получает запросы и декомпозирует их на подзапросы для реализации внутрizaпросного параллелизма с помощью подхода, реализованного в ParGRES. Он использует репликацию БД для выполнения виртуальной разметки. Такое разделение создает адаптивные виртуальные области, которые должны быть обработаны параллельно, аналогично обработчику запросов кластера в ParGRES. Эта служба выполняет также окончательную композицию;

службой локального запроса грид (Grid Local Query Service, GLQS) – локальной компонентой, предназначенной для приема подзапросов от службы распределенных запросов и передачи их локальным ParGRES. Эта служба контролирует выполне-

ние подзапроса на ParGRES в соответствии со следующими сценариями: для перераспределения запроса в случае, если узел занят, или для перенаправления подзапроса на другой узел.

Выбор БД нижнего уровня обусловлен требованиями надежности, отказоустойчивости и удобства сопровождения при решении задач планирования заданий в грид-сегменте. Технологичность данного решения достигается за счет использования СУБД и программного обеспечения промежуточного слоя архитектуры грид, что обеспечивается стандартными драйверами, сетевыми интерфейсами и протоколами передачи данных.

Для использования информационной технологии реализации модели планирования рассмотрим технологию обработки запроса, включающую следующие этапы прохождения запроса (множества запросов) к БД состояния ресурсов и заданий, которые формируются администратором локального ресурса РВС для осуществления планирования заданий.

Этап 1. Запрос передается на программную платформу промежуточного уровня GParGRES. На этом этапе запрос анализируется системой и, в зависимости от его структуры и (или) сложности, GParGRES может выполнить запрос на создание временной таблицы для агрегирования результатов, а также принять решение о постановке запроса на определенный(е) узел(ы) ParGRES. Следует отметить, что ParGRES работает не на уровне грид-сегмента, а фактически координирует узлы БД нижнего уровня. В рассматриваемой системе это узлы с установленным на них экземпляром PostgreSQL.

Этап 2. Система ParGRES анализируетхождение запроса для принятия решения о его выполнении на совокупности узлов или конкретном узле кластера БД.

Этап 3. Выполнение запроса осуществляется на конкретном узле (или узлах) PostgreSQL, результаты возвращаются в систему: сначала – в систему ParGRES, а затем агрегированные промежуточные результаты – в систему GParGRES, после чего они отправляются администратору системы.

Таким образом, рассмотренная архитектура решений для создания и обработки запросов в грид-сегменте с использованием средств параллелизма позволяет решить следующие задачи: оперативно обновлять программное обеспечение до актуальной версии без привязки одного слоя промежуточного ПО к другому; выполнять модификацию ПО любого уровня независимо от привязки к определенной реализации, например, БД или решений на уровне кластера в силу использования открытой архитектуры применяемых программных продуктов; обеспечивать на уровне администратора системы создание запроса (запросов) на любом уровне промежуточного ПО и доступ к данным на каждом узле PostgreSQL.

Описанные технологии обработки информации используются для решения: на уровне локального ресурса – задач мониторинга состояния сетевых интерфейсов, оценки загрузки узлов вычислительного кластера, состояния выполняемых заданий, мониторинга состояния запущенных на узлах кластера различных сервисов (программных агентов), а также контроля над загрузкой оперативной памяти, дисковых систем, БД и т. д.; на уровне грид-сегмента – обеспечения контроля над потоками заданий на ресурсы (кластеры, узлы кластеров), статусом выполняемых заданий, временем выполнения задания и т. д. с использованием компонент системы Nagios [8] или систем с аналогичным функционалом (Icinga, Zabbix). В случае масштабирования грид-инфраструктуры для распределенного мониторинга следует использовать систему корпоративного класса с открытыми исходными кодами Zabbix, характеризующуюся поддержкой параллельной СУБД PostgreSQL и наличием механизма обработки событий.

Выбор функциональности системы мониторинга, компоненты которой будут использоваться медиатором [2; 3] в предлагаемой модели планирования заданий, обосновывается наличием у нее следующих составляющих:

- возможности удаленного запуска скриптов – программных агентов;
- поддержки работы с БД PostgreSQL;
- механизма анализа событий.

При этом для решения задач мониторинга объектов контроля на уровне грид-сегмента РВС требуется не только осуществлять контроль над ресурсами, но и формировать потоки сообщений о происходящих на этих ресурсах событиях – о постановке заданий на выполнение и ход их выполнения.

В качестве удовлетворяющего требованиям состава компонент для функционирования медиатора грид-сегмента предлагается использовать открытый пакет Nagios или его ответвление – проект Icinga. Система Nagios [8] позволяет не только отслеживать параметры состояния узлов и сетевого оборудования РВС, но и формировать сообщения о ситуациях, возникающих в системе.

Технологически медиатор может получать данные о состоянии вычислительных кластеров (узлов) и коммуникационного оборудования по протоколу SNMP. Для отслеживания состояния служб, например, базы данных PostgreSQL, на узле должен быть установлен программный агент, который по журналам работы службы будет получать показатели результатов работы системы. При этом возможен сценарий, когда для получения данных о состоянии узлов и выполняемых заданий медиатор удаленно запускает скрипт на узле – программу (программное расширение), результатом выполнения которой

является информация, получаемая из лог-файлов локальных планировщиков и ЛСУР.

Универсальность предлагаемого решения на основе использования системы Nagios для контроля над состоянием грид-сегмента PBC обеспечивается расширением этого продукта средствами пакета NDOUtils. Это расширение позволяет узлу передавать данные о выполняемых работах во внешнюю базу данных (например, MySQL для Nagios). Аналогичное решение DB IDO (Database Icinga Data Output) как модуля для Icinga позволяет этой системе экспортировать конфигурацию и свое состояние в PostgreSQL в двухуровневой архитектуре грид-сегмента. Затем используя внешнее программное обеспечение, например, промежуточного слоя грид-сегмента, можно обратиться к этой БД и получить информацию о системе, например, о состоянии ее ресурсов и выполняемых заданий в ней заданий (workload).

Особенностью системы Nagios/Icinga является наличие конфигурационного файла, который содержит информацию обо всех контрольных точках. На уровне вычислительного узла использование компонент Icinga позволяет выполнять проверки состояния на основе стандартного протокола SNMP. Для мониторинга состояния узла на основе информации, получаемой от ЛСУР Torque, например, для анализа информации файла журнала (лог-файла), можно использовать существующий модуль или же разработанный скрипт. Поскольку Icinga позволяет запускать скрипт удаленно, ее можно использовать для оценки состояния выполняемых заданий на узлах вычислительных кластеров, использующих ПО Maui и PBS Torque.

Выводы

Таким образом, предложенная технология формирования и обработки информации о состоянии ресурсов и заданий в системе позволяет:

организовать процессы получения данных о состоянии объектов контроля грид-сегмента за счет использования унифицированной программной платформы, построенной на стандартных компонентах существующих систем мониторинга;

обеспечить планировщику грид получение оперативных данных о состоянии ресурсов и выполняемых на них заданиях.

Получение всеобъемлющей и оперативной информации позволяет формировать и обосновывать стратегии планирования менеджерами ВО и системными администраторами локальных ресурсов PBC.

Список литературы

1. Минухин С.В. Модели и методы решения задач планирования в распределенных вычислительных системах: монография / С.В. Минухин. – Х.: Изд-во ООО «Щедрая усадьба плюс», 2014. – 324 с.
2. Минухин С.В. Информационные технологии реализации двухуровневой модели планирования пакетов заданий в распределенной вычислительной системе на основе решения задачи о наименьшем покрытии / С.В. Минухин // Системи управління, навігації та зв'язку. – 2015. – Вип. 1(33). – С. 111-115.
3. Минухин С.В. Информационная технология для планирования заданий на вычислительных кластерах распределенной системы на основе интеграции сервисов удаленного доступа / С.В. Минухин // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 12(137). – С. 134-139.
4. Kotowski, A.A.B. Lima, E. Pacitti et al. // Concurrency and Computation: Practice and Experience. – 2008. – Vol. 20, Issue 17. – P. 2039-2048.
5. ParGRES: a middleware for executing OLAP queries in parallel. Technical Report ES-690 / Marta Mattoso, Geraldo Zimbrão, Alexandre A. B. Lima et al. – 2005. – 8 p.
6. Yue-sheng Tan. Study on Query Processing Mechanism of OGSA-DQP / Tan Yue-sheng, Wu Zhi, Wang Jing-yu // Communications in Information Science and Management Engineering. – 2011. – Vol. 1, No.1. – P. 22-25.
7. Akal F. OLAP Query Evaluation in a Database Cluster: a Performance Study on Intra-Query Parallelism / F. Akal, K. Böhm, Hans-Jörg Schek // Advances in Databases and Information Systems Lecture Notes in Computer Science. – 2002. – Vol. 2435. – P. 218-231.
8. Nagios – The Industry Standard in IT Infrastructure Monitoring [Электронный ресурс]. – Режим доступа: <http://www.nagios.org>.

Поступила в редколлегию 7.12.2017

Рецензент: д-р техн. наук проф. В.О. Алексеев, Харьковский национальный экономический университет им. С. Кузнеця, Харьков.

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ОБРОБЛЕННЯ ЗАВДАНЬ В РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ СЕРЕДОВИЩАХ

С. В. Мінухін

Розглянуто інформаційні технології для вирішення завдань планування завдань в дворівневої обчислювальної системі, що базуються на роботі кластерної СУБД і технології програмних агентів. Для реалізації технологій програмних агентів, що встановлюються на вузлах локальних ресурсів розподіленої системи, обрані компоненти систем моніторингу, що забезпечують оперативність і повноту необхідної для прийняття рішень про вибір стратегій планування інформації.

Ключові слова: розподілена обчислювальна система, обчислювальний кластер, кластерна СУБД, інформаційна технологія, система моніторингу, програмні агенти.

INFORMATION TECHNOLOGIES FOR PROCESSING JOBS IN DISTRIBUTED COMPUTING ENVIRONMENTS

S.V. Minukhin

An information technologies to solve scheduling problems in the two-level computing system based on the work of the cluster database and software agent technology are consider. To implement mounted on the nodes local resources of a distributed system software agent technology, selected components of the monitoring system to ensure the efficiency and completeness required for the choice of scheduling strategy decisions.

Keywords: distributed computing system, the computing cluster, the cluster DBMS, information technology, monitoring systems, software agents.