

УДК 004.6

С.В. Минухин

Харьковский национальный экономический университет имени Семена Кузнеця, Харьков

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ОБРАБОТКИ ЗАДАНИЙ В ДВУХУРОВНЕВОЙ РАСПРЕДЕЛЕННОЙ СИСТЕМЕ С ИСПОЛЬЗОВАНИЕМ СУБД POSTGRESQL

Рассмотрена информационная технология обработки и планирования заданий в двухуровневой распределенной системе. Для получения информации о состоянии ресурсов, заданий и очередей заданий предложена двухуровневая архитектура обработки информации. Технология обработки информации базируется на двухуровневой архитектуре – GParGRES и кластерной СУБД PostgreSQL, обеспечивающих оперативность обработки данных за счет параллельного выполнения запросов к БД, а также использования компонент систем мониторинга распределенных вычислительных сред.

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

Введение

Одним из подходов и методов разработки информационных технологий обработки заданий в распределенных вычислительных системах (РВС) является двухуровневая архитектура ParGRES [1], позволяющая:

- обеспечить параллельный доступ к данным БД о состоянии ресурсов, заданий, очередях на ресурсы и загрузке БД [2];

- использовать СУБД, поддерживающую технологию параллельного доступа к БД;

- разрабатывать БД состояния ресурсов, заданий, очередей заданий на ресурсы;

- использовать программное обеспечение (ПО), обеспечивающее удаленное получение информации о состоянии локальных ресурсов и заданий, выполняемых на узлах вычислительных кластеров грид-сегментов;

- использовать стандартные коммуникационные протоколы TCP/IP, SNMP, GridFTP;

- использовать программное обеспечение медиатора (посредника), обеспечивающего взаимодействие между поставщиками и потребителями информации [3–5].

В данном исследовании в качестве модели обработки и планирования заданий в РВС использована модель, исследованная в [3], в которой поставщиками информации являются ресурсы и задания, потребителями – менеджер виртуальной организации (системный администратор локального ресурса – вычислительного кластера)РВС. В роли последних также могут выступать пользователи, входящие в состав виртуальной организации (ВО), представляющие свои задания на обработку в распределенную систему. В качестве базиса архитектуры для реализации информационной технологии получения

и обработки данных о состоянии ресурсов и заданий предлагается использовать ParGRES [1], являющуюся кластером БД промежуточного слоя архитектуры грид-сегмента, использующая при обработке запросов внутри- и межзапросный параллелизм и обеспечивающая работу с репликациями. Параллелизм достигается за счет полной репликации БД и адаптивной виртуальной фрагментации (Adaptive Virtual Partitioning, AVP). ParGRES позволяет гибко распределять узлы для обработки запросов: любые запросы могут быть обработаны любым набором узлов кластера. При этом система AVP обеспечивает динамическое распределение нагрузки между узлами кластера во время обработки запроса. Как и в большинстве кластерных БД, ParGRES управляет параллельным выполнением запросов на основе экземпляров СУБД, установленных на узлах кластера БД [6, 7].

В ParGRES используются глобальные и локальные компоненты, формирующие двухуровневую организацию [1, 2] реализации модели обработки заданий [3]:

- глобальные компоненты – посредник и обработчик запросов кластера (Cluster Query Processor, CQP), выполняющие задания на нескольких узлах кластера;

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

Поскольку большинство из кластеров используют один доступный для внешних приложений управляющий узел, компонента «посредник», как правило, размещается на этом же узле и тем самым определяет централизованную архитектуру управления вычислениями и предоставляет возможность физического распределения обработки запросов на

кластере для каждого запроса, таким образом, повышая общую доступность ресурсов распределенной вычислительной среды.

Целью данной работы является разработка информационной технологии для модели обработки и планирования заданий в двухуровневой вычислительной системе [3] с использованием архитектуры ParGRES– GParGRES на уровне грид-сегмента и кластерной СУБД PostgreSQL на уровне узла.

Модель обработки заданий

Согласно двухуровневой модели обработки и планирования заданий на уровне грид-сегмента и ресурса РВС [3, 4], из заданий входной очереди формируется пул, организованный в виде пакета заданий, ранжированных по приоритету (например, директивному сроку), являющийся временным хранилищем заданий для их последующего планирования на доступные и свободные кластеры РВС. Управляемыми параметрами данной схемы обработки и планирования заданий являются: величина пула, интенсивность входного потока реальной системы, количество доступных и свободных на момент планирования ресурсов системы. При этом задания из пула выгружаются на выполнение через

интервалы времени, определяемые *периодом планирования*, определяемого на основе величины пула, то есть от количества заданий пакета, количества доступных и свободных на момент планирования вычислительных ресурсов (кластеров) РВС и назначаются на них на основе решения задачи о наименьшем покрытии [3].

Технология обработки заданий в двухуровневой распределенной системе

В информационной технологии для реализации расстраиваемой модели обработки и планирования заданий [3] предлагается использовать два уровня – уровень грид-сегмента (GParGRES) и уровень узла (на базе СУБД PostgreSQL) [8] (рис. 1). Такой выбор обусловлен архитектурными требованиями, предъявляемыми к грид-системам.

Сервисное программное обеспечение грид-системы и ее администратор (менеджер виртуальной организации) обеспечивают возможность работы с абстракцией уровня стандартной реляционной БД, в качестве которой используется PostgreSQL. На этом уровне можно создавать запросы к БД, используя стандартную технологию SQL-запросов.

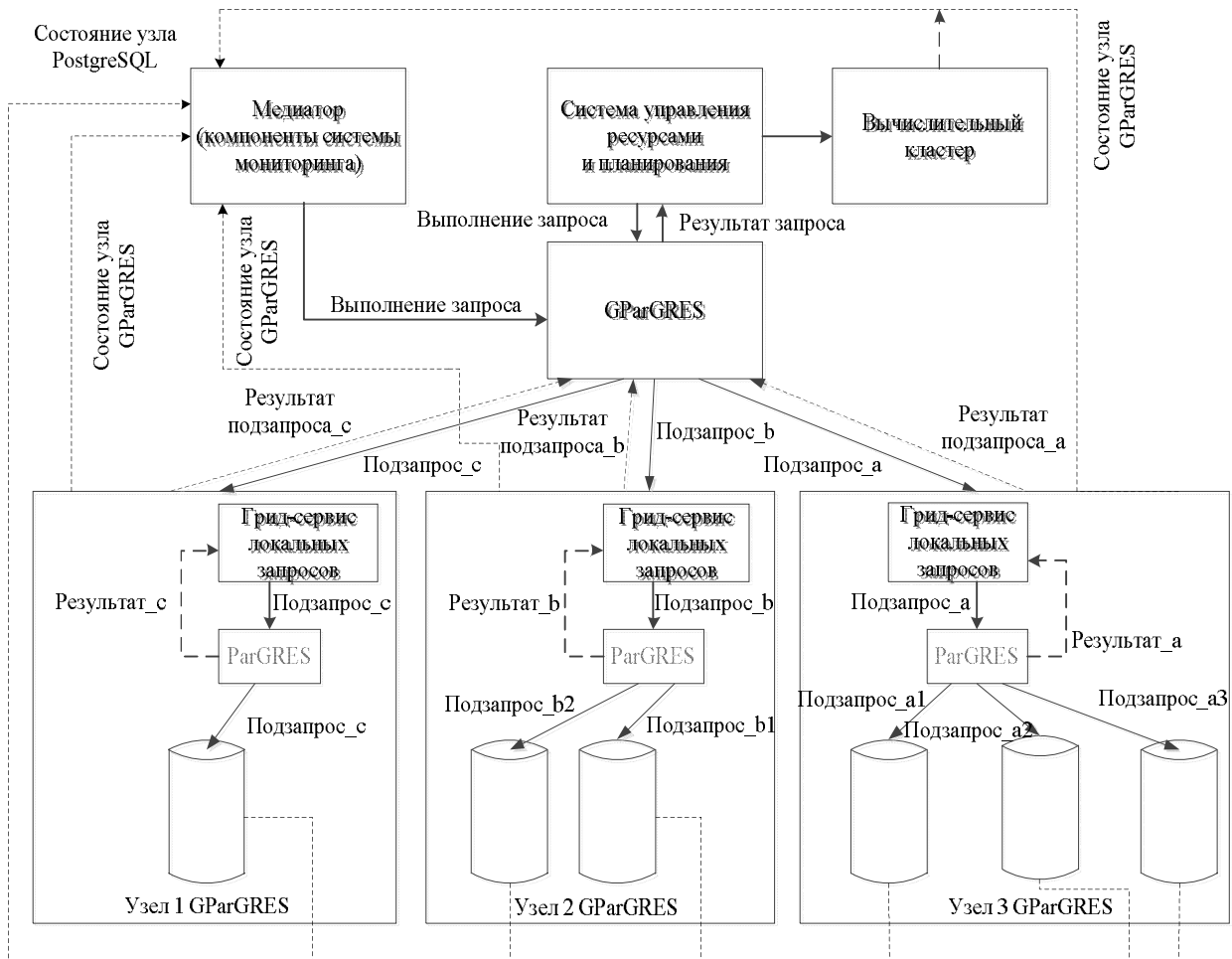


Рис. 1. Технология обработки заданий в двухуровневой распределенной системе

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 QueryService, 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 (GridLocal Query Service, GLQS) – локальной компонентой, предназначенной для приема подзапросов от службы распределенных запросов и передачи их локальным ParGRES. Эта служба контролирует выполнение 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 (узла) 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-сегмента, что обеспечивается стандартными драйверами, сетевыми интерфейсами и протоколами передачи данных [3 – 5].

Для реализации информационной технологии обработки и планирования заданий в РВС рассмотрим технологию обработки запроса, включающую следующие этапы прохождения запроса (множества запросов) к БД состояния ресурсов и заданий, которые формируются администратором локального ресурса (кластера) РВС для 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 работу узлов БД нижнего уровня системы. В рассматриваемой системе – это узлы с установленными на них экземплярами СУБД PostgreSQL.

Этап 2. Система ParGRES анализируетхождение запроса для принятия решения о его выполнении на множестве 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 отправляются администратору системы.

Таким образом, рассмотренная архитектура решений для создания и обработки запросов в грид-сегменте РВС с использованием технологий их параллельной обработки позволяет:

оперативно обновлять программное обеспечение до актуальной версии без привязки одного слоя промежуточного ПО РВС к другому;

выполнять модификацию ПО любого уровня независимо от привязки к его определенной реализации, например, БД или решений на уровне кластера, посредством 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.

Предлагаемые технологии обработки информации в двухуровневой РВС используются для решения следующих задач:

на уровне локального ресурса (узла) – определения состояния сетевых интерфейсов, оценки загрузки узлов вычислительного кластера, состояния выполняемых заданий, мониторинга состояния запущенных на узлах кластера различных сервисов на основе использования удаленных программных агентов, а также контроля над загруженностью оперативной памяти, дисковых систем, БД и т. п.;

на уровне грид-сегмента – обеспечения контроля над потоками (загрузками) заданий на локальные ресурсы (кластеры, узлы кластеров), статуса выполняемых заданий, временем выполнения заданий.

Выбор системы мониторинга, для реализации которой предлагается использовать медиатор [4, 5] в модели обработки заданий, обосновывается наличием у нее следующих составляющих:

удаленным запуском скриптов – программных агентов;

поддержкой работы с БД PostgreSQL;
механизмом анализа событий.

Одной из распространенных систем мониторинга грид-систем, компоненты которой предлагается использовать для реализации информационной технологии модели обработки и планирования заданий, является система Ganglia [9, 10], использующая для хранения данных технологию базы данных на базе циклической системы (Round Robin). Такой подход позволяет, используя фиксированный размер хранилища, выполнять обработку множества данных о параметрах, регистрируемых для обеспечения контроля над различными объектами РВС.

Технология Ganglia позволяет решать задачи мониторинга на нижнем (кластерном) уровне грид-сегмента. Однако для решения задач мониторинга на уровне грид-сегмента требуется не только осуществлять контроль над ресурсами, но и формировать потоки сообщений о происходящих на этих ресурсах событиях – о постановке заданий на выполнение, их выполнение и анализ полученных результатов выполнения заданий.

В качестве удовлетворяющего существующим требованиям состава компонент для функционирования медиатора грид-сегмента [4, 5] предлагается использовать открытый пакет Nagios (проект Icinga) [11]. Система Nagios позволяет не только отслеживать параметры состояния узлов и сетевого оборудования РВС, но и формировать сообщения о ситуациях, возникающих при этом в системе.

Технологически медиатор может получать данные о состоянии сетевых 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 формировать и передавать данные о результатах работы системы. При этом возможен сценарий, когда для получения данных о состоянии узлов и выполняемых на них заданиях медиатор удаленно запускает скрипт на узле – программу (программное расширение), результатом выполнения которой является информация, получаемая из лог-файлов локальных планировщиков и ЛСУР [4, 5].

Универсальность предлагаемого решения на основе использования системы мониторинга Nagios для контроля над состоянием объектов уровня грид-сегмента обеспечивается расширением этого продукта средствами пакета NDOUtils [12]. Это

расширение позволяет узлу передавать данные о выполняемых заданиях во внешнюю базу данных (например, MySQL для Nagios). Аналогичное решение DB IDO (Database Icinga Data Output) в качестве модуля системы Icinga позволяет экспортировать конфигурацию и свое состояние в PostgreSQL в двухуровневой архитектуре грид-сегмента. Затем внешнее программное обеспечение, например, планировщик грид-сегмента, может обратиться к этой БД и получить текущую информацию о системе, например, о состоянии ее ресурсов, выполняемых заданиях, загрузке БД и т. д.

На уровне вычислительного узла грид-сегмента РВС использование компонент Icinga позволяет выполнять проверки состояния на основе стандартного протокола SNMP. Для мониторинга состояния узла на основе информации, например, получаемой от ЛСУР Torque при анализе данных файла журнала (лог-файла), можно использовать существующий модуль или же разработанный скрипт [4, 5].

Для удаленного запуска скриптов системой Icinga/Nagios на сервере медиатора устанавливается программное расширение Nagios Remote Plugin Executor (NRPE), а на удаленном узле – клиентская часть системы (программный агент), поддерживающая NRPE. Использование компонент системы Icinga позволяет выполнять проверки состояния узлов – как по протоколу SNMP, так и по стеку протоколов TCP/IP. Кроме этого, система позволяет разработать новый плагин, например, плагин для взаимодействия с СПО PBS Torque. Для работы с плагином системный администратор должен установить следующие пороговые значения параметров:

«OK» (код возврата 0) – сервис работает нормально;

«WARNING» (код возврата 1) – сигнал предупреждения о проблеме (событие 1);

«CRITICAL» (код возврата 2) – критическое состояние сервиса (событие 2);

«UNKNOWN» (код возврата 3) – неизвестное состояние сервиса (событие 3).

В результате проверки плагин будет генерировать состояния системы, формируя для проведения контроля множество событий, которые являются результатом работы плагинов [4, 5, 10, 11]: Схема алгоритма обработки событий приведена на рис. 2. На рис. 2 приведены процедуры обработки данных о состоянии узлов грид-сегмента, состоянии узлов кластера – на основе данных лог-файлов ЛСУР Torque и данных лог-файлов локального планировщика Maui – и возникающих при этом событий.

Для автоматизации процесса запуска скриптов предлагается обобщить технологию удаленного использования скриптов, рассмотренную в работе [5], на кластерах грид-сегмента РВС. С этой целью в структуру грид следует ввести хранилище GridFTP

[13], к которому будут иметь доступ узлы вычислительного кластера. Соответственно, на узле системы с помощью команды `check_npre-N` с `update_npre`

можно вызвать скрипт, загружающий необходимые файлы обновления для обеспечения оперативной обработки данных.

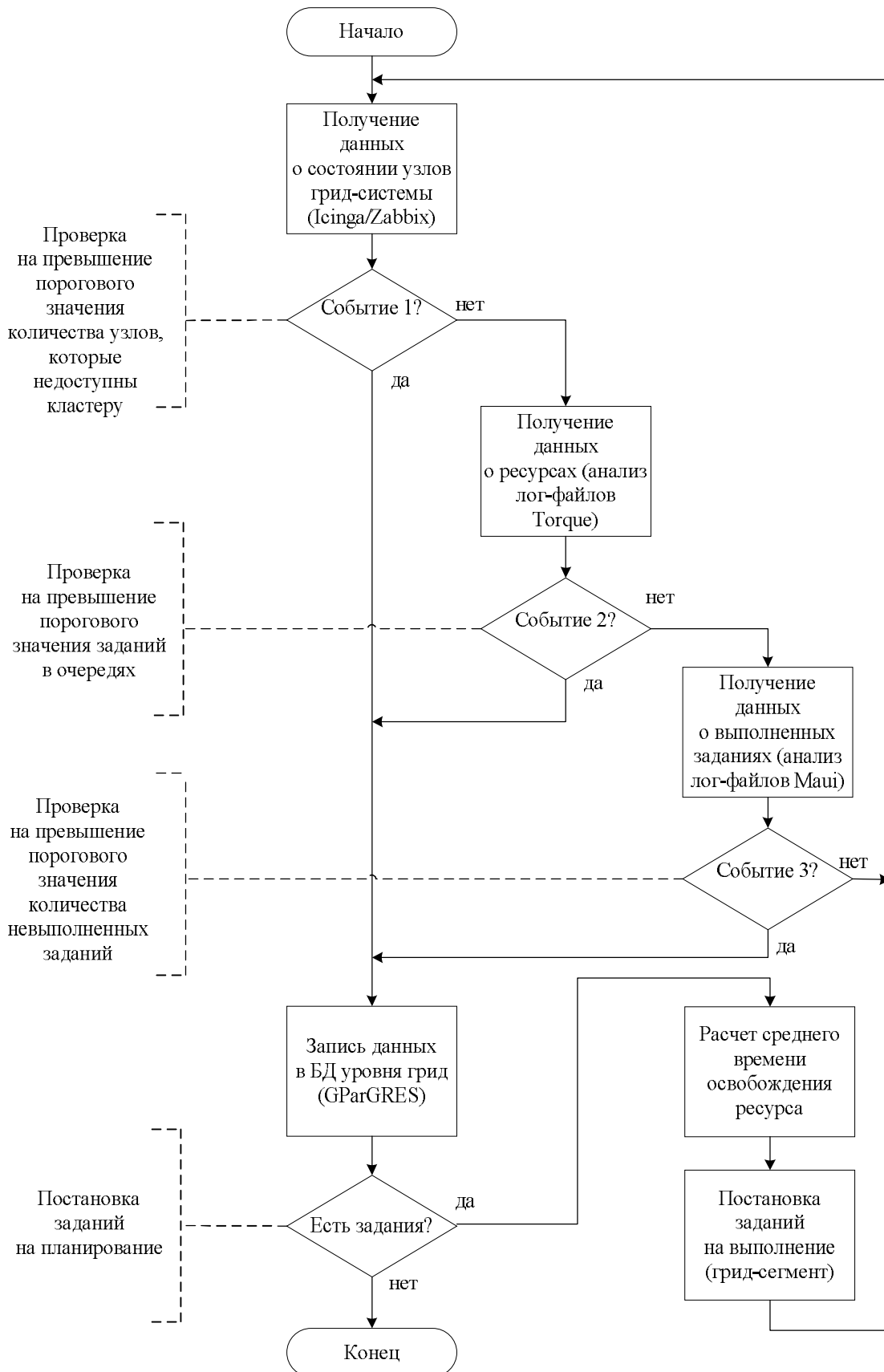


Рис. 2. Схема алгоритма интегрированной обработки событий на основе данных о состоянии ресурсов и заданий

Выводы

Рассмотрена информационная технология обработки и планирования пакетов заданий на уровне грид-сегмента PBC, базирующаяся на двухуровневой архитектуре ПО промежуточного слоя ParGRES. Данная архитектура позволяет повысить эффективность работы с транзакциями при выполнении большого количества запросов к БД, сформированной на основе данных о состоянии узлов и выполняемых на них заданиях путем опроса лог-файлов локальных систем управления ресурсами и локальных планировщиков программными агентами, обеспечивающими текущий контроль над объектами управления грид-сегмента PBC, на основе использования параллельных методов доступа к БД. Для повышения эффективности принятия решений системными администраторами вычислительных кластеров грид-сегментов предлагается использовать компоненты и коммуникационные технологии систем мониторинга, в частности, технологию удаленно работающих программных агентов, позволяющую на основе стандартных коммуникационных протоколов автоматизировать процессы обработки данных о состоянии объектов управления.

Список литературы

1. High-performance Query Processing of a Real-world OLAP Database with ParGRES [Электронный ресурс] / Melissa Paes, Alexandre A.B. Lima, Patrick Valduriez et al. – Режим доступа : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.463.5455&rep=rep1&type=pdf>.
2. Parallel query processing for OLAP in grids / N. Kotowski, A.A.B. Lima, E. Pacitti et al. // *Concurrency and Computation: Practice and Experience*. – 2008. – Vol. 20, Issue 17. – P. 2039-2048.
3. 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.
4. 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 // *Системы управління, навігації та зв'язку*. – Полтава : ПНТУ, 2015. – Вип. 1(33). – С. 111-115.
5. Минухин С.В. Информационная технология для планирования заданий на вычислительных кластерах распределенной системы на основе интеграции сервисов удаленного доступа / С.В. Минухин // *Системы обработки информации*. – Х.: ХУПС, 2015. – Вип. 12 (137). – С. 134-139.
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. PostgreSQL [Electronic resource]. – Access mode : <http://www.postgresql.org/>.
9. Ganglia Monitoring System [Electronic resource]. – Access mode : ganglia.sourceforge.net.
10. Ganglia u Nagios: Часть 1. Мониторинг коммерческих кластеров с помощью Ganglia [Electronic resource]. – Access mode : <https://www.ibm.com/developerworks/ru/library/l-ganglia-nagios-1/>.
11. Nagios – The Industry Standard in IT Infrastructure Monitoring [Электронный ресурс]. – Режим доступа : <http://www.nagios.org>.
12. NDOUtils 2.1.1 [Электронный ресурс]. – Режим доступа : <https://www.nagios.org/news/2016/09/ndoutils-2-1-1-released/>.
13. A Tutorial on Configuring and Deploying GridFTP for Managing Data Movement in Grid/HPC Environments [Электронный ресурс]. – Режим доступа : <http://www.mcs.anl.gov/kettimut/tutorials/SC07GridFTPTutorialSlides.pdf>.

Надійшла до редколегії 23.03.2017

Рецензент: д-р техн. наук, проф. В.О. Алексієв, Харківський національний економічний університет імені Симеона Кузнеця, Харків.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОБРОБЛЕННЯ ЗАВДАНЬ В ДВОРІВНЕВІЙ РОЗПОДІЛЕНІЙ СИСТЕМІ З ВИКОРИСТАННЯМ СУБД POSTGRESQL

С.В. Мінухін

Розглянута інформаційна технологія оброблення і планування завдань у дворівневій розподіленій системі. Для отримання інформації про стан ресурсів, завдань і черг завдань запропонована дворівнева архітектура оброблення інформації. Технологія оброблення інформації базується на дворівневій архітектурі – GParGRES і кластерній СУБД PostgreSQL, які забезпечують оперативність оброблення даних за рахунок паралельного виконання запитів до БД, а також використання компонент систем моніторингу розподілених обчислювальних середовищ.

Ключові слова: розподілена обчислювальна система, обчислювальний кластер, база даних, кластерна СУБД, інформаційна технологія, система моніторингу.

INFORMATION TECHNOLOGY OF JOB PROCESSING IN A TWO-LEVEL DISTRIBUTED SYSTEM USING DBMS POSTGRESQL

S.V. Minukhin

The information technology of processing and scheduling tasks in a two-level distributed system is considered. To obtain information about the status of resources, tasks and job queues, a two-level information processing architecture is proposed. Information processing technology is based on the two-tier –GParGRES architecture and clustered DBMS PostgreSQL, which ensure the speed of data processing due to parallel execution of database queries, as well as the use of components of monitoring systems for distributed computing environments.

Keywords: distributed computing system, computational cluster, database, cluster DBMS, information technology, monitoring system.