УДК 004.75

С.В. Минухин

Харьковский национальный экономический университет имени С. Кузнеца, Харьков

ИНФОРМАЦИОНАЯ ТЕХНОЛОГИЯ ДЛЯ ПЛАНИРОВАНИЯ ЗАДАНИЙ НА ВЫЧИСЛИТЕЛЬНЫХ КЛАСТЕРАХ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ НА ОСНОВЕ ИНТЕГРАЦИИ СЕРВИСОВ УДАЛЕННОГО ДОСТУПА

Предложена информационная технология для планирования заданий на вычислительных кластерах распределенных систем, использующая сервисы удаленного доступа систем мониторинга и кластерную СУБД для работы с данными о состоянии узлов и выполняемых заданий кластера. Рассмотрена процедура выбора сервисов и разработана модель мультисервисного удаленного доступа, полученная на основе интеграции сервисов. Построены отображения этапов работы модели планирования на множество сервисов удаленного доступа, оптимизирующие количество используемых сервисных модулей.

Ключевые слова: *распределенная вычислительная система, вычислительный кластер, база данных, кластерная СУБД, сервис, удаленный доступ, программный агент, информационная технология.*

Введение

Функционирование современных распределенных вычислительных систем (РВС) поддерживается информационными службами – сервисами, обеспечивающими сбор и передачу данных, и системами хранения данных, использующими различные протоколы доступа и модели данных.

Распространенным инструментом, включающим целые комплексы разнообразных сервисов для сбора данных о состоянии ресурсов, являются системы мониторинга [1 – 3]. В настоящее время они развиваются по следующим основным направлениям: разработка систем метамониторинга РВС [1] и разработка систем мониторинга вычислительных кластеров [2, 3]. Актуальным является разработка средств и технологий взаимодействия локальной системы управления вычислительным кластером и мониторинга узлов с планировщиком вычислительного кластера [3].

Целью данного исследования является разработка информационной технологии для планирования и выполнения заданий на вычислительных кластерах РВС в рамках модели пакетного планирования заданий, рассмотренной в работах [4, 5], на базе информационных технологий, предложенных в работе [6].

Базовая технология организации работы сервисов удаленного доступа

В предложенных в работе [6] информационных технологиях для двухуровневой модели пакетного планирования заданий в РВС использованы: технологии обработки информации на вычислительных кластерах – поставщиках информации, информационные технологии потребителей информации – пользователей и администраторов грид-сегментов и медиатора.

Для информационной технологии управления заданиями на вычислительном кластере, базирующейся на сборе данных о состоянии заданий и узлов, предлагается использовать сервисы удаленного доступа – программные расширения (агенты) системы Nagios [7]. Данная технология реализуется путем инициализации запуска удаленных скриптов, получения информации о состоянии заданий и ресурсов и ее записи в БД РВС.

Управление сервисами осуществляется медиатором – управляющим узлом, выполняющим функции координатора процессов сбора и записи информации в БД РВС и использующего распределенное программное обеспечение NPRE [7], с одной стороны, и системными администраторами грид-сегментов и пользователями – с другой.

Для оценки состояния служб на сервере кластера и на его узлах устанавливаются агенты NPRE, обеспечивающие получение данных о состоянии узлов и заданий из журналов работы соответствующих сервисов. Для идентификации анализируемых объектов используется конфигурационный файл системы Nagios, в который записывается информация об именах и состоянии узлов: папка /etc/nagios3/conf.d/ содержит основные конфигурационные файлы с информацией об узлах кластера; папка ./hosts_nagios2.cfg – имена и IP-адреса узлов [8, 9].

Процесс установки и настройки NRPE включает следующую последовательность этапов [8].

1. Подготовительный этап.

Реализуется на основе выполнения следующей последовательности команд:

```
apt-getinstallxinetd
apt-get install libssl-dev
```

```
apt-get install openssl
groupaddnagios
useraddnagios -d /home/nagios -g nagios –m
```

2. Этап установки.

Реализуется на основе выполнения следующей последовательности команд:

```
cd /home/manage
wget citylan.dl.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.13/nrpe-2.13.tar.gz
tar -xzf ./nrpe-2.13.tar.gz
cd ./nrpe-2.13
./configure --with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/x86_64-linux-gnu
make all
makeinstall&&makeinstall-xinetd.
```

3. Этап конфигурирования.

Осуществляется на основе выполнения следующей последовательности шагов:

редактирование файла

/etc/xinetd.d/nrpe: only_from = 127.0.0.1,

IP-адрес узла NRPE, дописывается IP-адрес узла Nagios (в общем случае – узлы, на которых будет осуществляться запуск скриптов);

добавление в файл

/etc/servicesописаниеNRPE: nrpe 5666/tcp # NRPE

повторный старт службы:

servicexinetdrestart.

4. Этап настройки оповещений и уведомлений.

Реализуется путем редактирования файла на основе выполнения следующей последовательности шагов:

добавление в файл

vim /etc/nagios3/conf.d/contacts_nagios2.cfg

в группу admins описания контактов для рассылки уведомлений на основе выполнения следующей последовательности команд:

```
define contact{
contact_name manage
alias Nagios Administrator
service_notification_period 24x7
host_notification_period 24x7
service_notification_optionsc,r,
```

где параметр сопределяет сообщения о критических состояниях, r – сообщение о восстановлении сервиса;

host_notification_optionsd,r,

где параметрdопределяет сообщение об отключении системы, параметрr – сообщение о восстановлении системы;

service_notification_commandsnotify-service-by-email

host_notification_commands notify-host-by-email

emailadmin@yandex.ru}.

В файле конфигурации

vim /etc/nagios3/conf.d/generic-service_nagios2.cfg

указываются настроечные параметры времени и событий – временной интервал и перечень событий для высылки писем:

```
notification_interval 240 //240 = 4 часа
notification_optionsc,r,
```

где параметр с определяет сообщения о критических состояниях, параметр r –о восстановлении сервиса;

```
notification_interval 240
notification_optionsd,u,r,
```

где параметр d определяет сообщение об отключении системы, u – сообщение о недоступности системы, r – сообщение о восстановлении системы.

После установки и настройки агентов NPRE для удаленного вызова сервисов на узлах вычислительного кластера необходимо в соответствии с решаемыми задачами планирования определить состав сервисов удаленного доступа к узлам вычислительного кластера.

Выбор состава сервисов удаленного доступа

Для получения и обработки данных о состоянии программно-аппаратных средств и сетевых компонент вычислительного кластера РВС предлагается использовать следующие типы сервисов удаленного доступа [7]:

определение доступности и уровня загрузки узлов вычислительного кластера;

определение состояния сети, включая коммуникационные каналы к вычислительному кластеру;

определение доступности и уровня средней загрузки многоядерных процессоров узлов вычислительного кластера;

определение доступности и количества свободных узлов вычислительного кластера;

определение состояния заданий и узлов вычислительного кластера;

определение доступности, производительности, нагрузки и объема используемой БД на узле кластерной СУБД.

Для использования приведенных типов сервисов они сгруппированы в соответствии с их функциональностью и сведены в табл. 1 – 3. Работа каждого сервиса удаленного доступа включает: процедуру инициализации запуска, осуществляемого агентамиNRPE на основе команды check_nrpe; выполнение сервиса; получение результатов работы сервиса; передачу данных медиатору и создание запросов на запись данных в БД РВС (рис. 1). На рис. 1 программный модуль 1 является программной реализацией метода минимизации суммарного запаздывания заданий [5], программный модуль 2 – программной реализацией методов управления частотой и напряжением процессоров узлов кластера.

Таблица 1

Состав сервисов удаленного доступа
для получения данных о состоянии узлов кластера

№ п/п	Сервис	Характеристика
1	check_linux_net	Мониторинг сети
2	check_time_ping	Время пингования узла
3	check_net.sh	Пропускная способность сети на локальном узле
4	check_proc_cpu.sh: check process cpu usage	Проверка процессора, используемого в текущем процессе
5	check_cpu.sh	Использование процессора (система, ввод/вывод, простой, %)
6	multi core load average checks	Количество ядер, доступных в системе, текущая средняя загрузка многоядерного узла, %
7	check_cpu_stats fixed	статистика центрального процессора (система, ввод/вывод, простой, загрузка, потери, %)

Таблица 2

Состав сервисов удаленного доступа
для получения данных о заданиях и ресурсах вычислительного кластера

№ п/п	Сервис	Характеристика
1	check_pbssched	Мониторинг состояния планировщика PBSMAUI, «прослушивание» заданного порта
2	check_pbs_E	Задание, находящееся состоянии E: задание завершено после выполнения определенного набора операций
3	check_pbsnodes	Нефункционирующие узлы вычислительного кластера, использующего пакеты Moab/Maui&Torque для планирования заданий и очередей путем вызова команды showq
4	check_job_overtime	Задания, для которых превышен директивный срок

Таблица 3

Состав сервисов удаленного доступа
для получения данных о доступности и состоянии кластера СУБДи базы данных

№ п/п	Сервис	Характеристика
1	check_database_pgsql	Подключение к базе данных PostgreSQL на узле и выполнение простого запроса
2	check_pgactivity	Мониторинг работы кластера PostgreSQL.Пороги: ограничение значения в %; интервал времени: s (секунды), м (минуты), h (часы), d (день); объем – b (байт), k (Килобайт), m (Мегабайт), g (Гигабайт), t(терабайт)
3	pg_db_size	Объем БД PostgreSQL. Требуются библиотека pqxx (C ++ классы обертки в PostgreSQL) и библиотека libpq PostgreSQL
4	check_psql_query	Запрос к PostgreSQL. Запуск запроса на сервере Pg и проверка порогов времени ответа на запрос
5	check_pgsql_connections	Количество подключений, доступных на сервере PostgreSQL. Используется "SELECTCOUNT (*) FROMpg_stat_activity" / "show" max_connections. Определение количества доступных соединений, меньшего нижнего порога
6	check_pgsql_queries	Суммирование запросов (SELECT, INSERT, DELETE,UPDATEALTER, CREATE, TRUNCATE), выполняемых на сервере PostgreSQL. Используется для ограничения количества запросов к БД PostgreSQL

Для записи данных, полученных в результате выполнения сервисов, в БД РВС используется двухуровневая архитектура GParGRES и СУБД PostgreSQL.

Для оценки доступности и состояния узлов кластерной СУБД PostgreSQL, оценки объемов и длительности запросов к БД использованы сервисы (2), для оценки состояния узлов под управлением локального менеджера ресурсов Torque, использованы сервисы (3), для контроля работы планировщика заданий Mauii определения запаздываемых заданий используются сервисы (4), для

запуска всех сервисов используются агенты NRPE (5).

Для поставщика информации – вычислительного кластера, в качестве которого выступает лог-файл MauiTraceFileFormat [10], следует использовать модифицированную процедуру check_pbssched, в которую необходимо включить процедуру записи данных из лог-файла в таблицы БД СУБД PostgreSQL. Медиатор в рассматриваемой системе выполняет функции управляющего узла, определяющего доступность и работоспособность каждого вычислительного кластера грид-сегмента, распределенной СУБД, функционирующей на основе системы промежуточного уровня ParGRES; осуществляет запуск на заданных интервалах времени команды check_nrpe и обработку полученных результатов работы установленных на узлах агентов NRPE для оценки состояния сети, узлов вычислительного кластера и кластера БД.

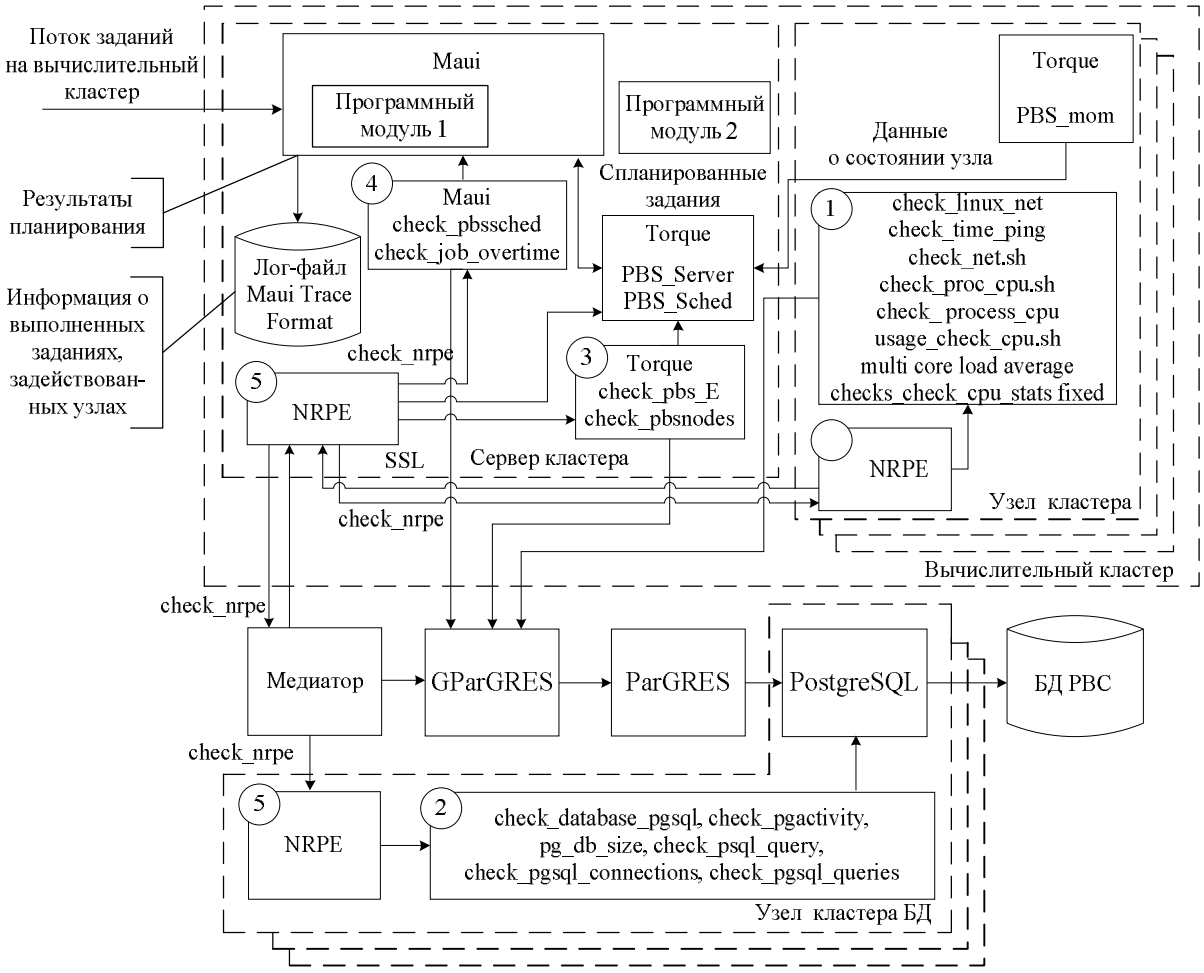

Рис. 1. Информационная технология на основе интеграции сервисов удаленного доступа к узлам вычислительного кластера

Для снижения нагрузки на медиатор при решении задач масштабирования вычислительного кластера и определении доступности его узлов используется распределенная система агентов NRPE(5) (рис. 1).

Модель формирования структуры сервисов удаленного доступа

Модель мультисервисного удаленного доступа к данным о состоянии ресурсов и заданий кластера представляется в следующем виде:

$$Rem\_Serv = \langle agent\_check\_nodes, agent\_check\_util, agent\_check\_net, agent\_check\_jobs, agent\_check\_DB \rangle, \quad (1)$$

где *agent_check_nodes* = = {*check_time_ping*,*check_linux_net*} – множество сервисов определения доступности узлов вычислительного кластера;

agent_check_util = {*check_proc_cpu.sh: checkprocesscpuusage*,*check_cpu.sh*,*multicoreloadaveragechecks*} – множество сервисов определения загрузки процессоров кластера;

agent_check_net = = {*check_linux_net*,*check_net.sh*} – множество сервисов определения состояния сети;

agent_check_jobs={*check_pbssched*,*check_pbs_E*, *check_pbsnodes*,*check_job_overtime*} – множество сервисов определения состояния заданий;

agent_check_DB={*check_pgsql_connections*, *check_pgactivity*,*pg_db_size*,*check_psql_query*,*check_database_pgsql*, *check_pgsql_querie*} – множество сервисов определения состояния баз данных и СУБД.

Приведенные множества сервисов связаны между собой следующими отношениями:

agent_check_nodes×*number_nodes*–отношение, определяющее мощность множества общего количества сервисов на всех узлах вычислительного кластера;

agent_check_net×*agent_check_nodes* – отношение, определяющее мощность множества сервисов для оценки совместной доступности коммуникационных (сетевых) каналов и узлов вычислительного кластера;

agent_check_net×*agent_check_nodes*×*agent_check_jobs* – отношение, определяющее мощность множества сервисов для оценки состояния выполняемых заданий на доступных сетевых каналах и узлах;

agent_check_nodes×*agent_check_util* – отношение, определяющее мощность множества сервисов для оценки загруженности узлов вычислительного кластера;

agent_check_nodes×*agent_check_jobs*×*agent_check_DB* – отношение, определяющее мощность множества сервисов для оценки интегрированного взаимодействия вычислительных узлов, выполняемых заданий и БД.

Для обеспечения работы двухуровневой модели планирования заданий [6] требуется согласование этапов ее работы с множеством сервисов удаленного доступа к вычислительным ресурсам. Для этого на основе этапов планирования пакетов заданий [4, 5], принципов работы модели, выбора параметров компонент модели и периодичности планирования можно построить отображения на выбранные сервисы удаленного доступа (табл. 4).

Таблица 4

Алгоритмическая модель
отображения работы модели планирования на сервисы удаленного доступа

Последовательность	Наименование этапа	Сервисы удаленного доступа
Шаг 1	Формирование входной очереди заданий	*agent_check_net*∪*agent_check_DB*
Шаг 2	Формирование пакета (пула) заданий из заданий входной очереди	*agent_check_net*∪*agent_check_DB*
Шаг 3	Определение доступных и свободных ресурсов	*agent_check_nodes*∪ *agent_check_net*∪*agent_check_DB*
Шаг 4	Определение загруженности ресурсов	*agent_check_nodes*∪*agent_check_util*∪ *agent_check_net*∪*agent_check_DB*
Шаг 5	Выбор и назначение требуемых ресурсов для заданий пула	*agent_check_nodes*∪*agent_check_util*∪ *agent_check_net*∪*agent_check_jobs*∪ *agent_check_DB*
Шаг 6	Планирование пакета заданий	*agent_check_net*∪*agent_check_jobs*∪ *agent_check_DB*
Шаг 7	Помещение спланированных заданий в очереди на локальные ресурсы	*agent_check_util*∪*agent_check_net*
Шаг 8	Перепланирование заданий в случае занятости или высокой загруженности ресурсов	*agent_check_util*∪*agent_check_net*∪ *agent_check_jobs*∪*agent_check_DB*
Шаг 9	Планирование заданий на локальных ресурсах (MAUI)	*agent_check_util*∪*agent_check_net*∪
Шаг 10	Выполнение заданий на локальных ресурсах	*agent_check_jobs*∪*agent_check_DB*
Шаг 11	Отправка результатов выполненных заданий пользователям	*agent_check_net*
Шаг 12	Определение периодичности планирования на основе информации БД	*agent_check_util*∪*agent_check_net*∪ *agent_check_jobs*∪*agent_check_DB*

Периодичность планирования T_{Shed} с учетом времени реализации процедур удаленного доступа

$$t_{Rem\_Serv} = t_{send\_Rem\_Serv} + t_{exec\_Rem\_Serv} + t_{receive\_Rem\_Serv} + t_{insert\_Rem\_Serv}, \quad (2)$$

где $t_{Rem\_Serv}, t_{send\_Rem\_Serv}, t_{exec\_Rem\_Serv}, t_{receive\_Rem\_Serv}, t_{insert\_Rem\_Serv}$ – время посылки, выполнения, получения и записи результата в БД соответственно, и времени освобождения ресурсов системы $T_{dealloc}$ [5]определится следующим образом:

$$t_{Rem\_Serv} < T_{Shed} \le T_{dealloc}. \quad (3)$$

Уточнение величины периодичности (3) с учетом времени посылки запросов и получения результатов (2) выполнения сервисов позволит повысить эффективность функционирования кластерных систем и грид-сегментов.

Выводы

Предложена информационная технология планирования выполнения заданий на вычислительных кластерах распределенных систем, использующая сервисы удаленного доступа для сбора информации о состоянии выполняемых заданий и узлов кластера и ее записи в БД РВС.

В основе технологии лежит интеграция выбранных программных агентов, включая агентов локального планировщика Mauii локального менеджера ресурсов Torque, а также использование данных лог-файлов, отражающих результаты планирования заданий и состояние узлов вычислительного кластера. Отличительной особенностью предложенной технологии является использование множеств программных агентов на этапах работы модели планирования, что позволяет построить отображения, показывающие связи этапов планирования заданий со структурой программных агентов. Это предоставляет возможность сформировать и адаптировать модульную структуру сервисов к стратегиям и методам планирования заданий на вычислительном кластере.

В дальнейшем предполагается продолжить исследования в направлении унификации компонент информационной технологии для управления вычислениями на кластерах грид-систем.

Список литературы

1. Методы и средства метамониторинга распределенных вычислительных сред / И.А. Сидоров, А.П. Новопашин, Г.А. Опарин, В.В. Скоров // Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика». – 2014. – Т. 3, № 2. – С. 30-42.

2. Об одном подходе к мониторингу, анализу и визуализации потока заданий на кластерной системе / А.В. Адинец, П.А. Брызгалов, Вад.В. Воеводин, С.А. Жуматий, Д.А. Никитенко // Вычислительные методы и программирование. – 2011. – Т. 12. – С. 90-93.

3. Линёв А.В. Реализация унифицированного доступа к информации о состоянии мультикластера в программном компоненте управления платформами исполнения MCMS / А.В. Линёв, В.Д. Кустикова // Информационные технологии. Вестник Нижегородского университета им. Н.И. Лобачевского. – 2011. – № 3(2). – С. 248-252.

4. Листровой С.В. Модель и подход к планированию распределения ресурсов в гетерогенных Грид-системах / С.В. Листровой, С.В. Минухин // Международный научно-технический журнал «Проблемы управления и информатики». – 2012. – № 5. – С. 120-133.

5. 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.

6. 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.

7. Nagios – The Industry Standardin IT Infrastructure Monitoring [Электронный ресурс]. – Режим доступа к ренсурсу: http://www.nagios.org.

8. Развертывание и настройка Nagios v.3 [Электронный ресурс]. – Режим доступа к ресурсу: http://habrahabr.ru/sandbox/65382/.

9.Ganglia и Nagios: Часть 2. Мониторинг коммерческих кластеров с помощью Nagios [Электронный ресурс]. – Режим доступа к ресурсу: http:// www.ibm.com/ developerworks/ru/library/l-ganglia-nagios-2/.

10. Maui Trace File Format, version 310 [Электронный ресурс]. – Режим доступа к ресурсу: http://docs.adaptivecomputing.com/maui/trace.php.

Поступила в редколлегию 2.11.2015

Рецензент: д-р техн. наук, проф. В.О. Алексеев, Харьковский национальный экономический университет имени Семена Кузнеца, Харьков.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ПЛАНУВАННЯ ЗАВДАНЬ НА ОБЧИСЛЮВАЛЬНИХ КЛАСТЕРАХ РОЗПОДІЛЕНОЇ СИСТЕМИ НА ОСНОВІ ІНТЕГРАЦІЇ СЕРВІСІВ ВІДДАЛЕНОГО ДОСТУПУ

С.В. Мінухін

Запропоновано інформаційну технологію для планування завдань на обчислювальних кластерах розподілених систем, що використовує сервіси віддаленого доступу систем моніторингу та кластерну СУБД для роботи з даними про стан вузлів та завдань, що виконуються на кластері. Розглянуто процедуру вибору сервісів і розроблено модель мультисервісного віддаленого доступу, отриману на основі інтеграції сервісів. Побудовано відображення етапів роботи моделі планування на множину сервісів віддаленого доступу, яка оптимізує кількість сервісних модулів, що використовуються.

Ключові слова: *розподілена обчислювальна система, обчислювальний кластер, база даних, кластерна СУБД, сервіс, віддалений доступ, програмний агент, інформаційна технологія.*

INFORMATION TECHNOLOGY FOR TASK SCHEDULING ON COMPUTING CLUSTERS OF DISTRIBUTED SYSTEM BASED ON THE REMOTE ACCESS SERVICES INTEGRATION

S.V. Minukhin

An information technology for task scheduling on computing clusters of distributed systems based on using remote access services of monitoring systems and cluster database management system for processing data, that concerns the state of nodes and tasks performed by the cluster, is provided. The procedure of service choice is considered and the model of multi-remote access obtained through the services integration is developed. Mapping stages of model tasks scheduling on the set of remote access services that optimizes the number of used service modules are developed.

Keywords: *distributed computing system, computing cluster, database, cluster database management system, service, remote access, software agent, information technology.*