УДК 004.6

С. В. Минухин, К. С. Молчанов, М. Г. Сызранцев

Харьковский национальный экономический университет имени Семена Кузнеца, Харьков

Информационная технология оценки показателей функционирования вычислительного кластера распределенной системы

Предложена информационная технология для решения задач планирования заданий на вычислительном кластере, базирующаяся на обеспечении информацией данными о состоянии узлов и заданий, формирующих базу данных на основе лог-файлов локального планировщика Maui. Рассмотрена схема ее реализации с использованием разработанных файлов сценария заполнения базы данных и формирования отчетов о показателях работы кластера по запросу за заданные периоды времени.

Ключевые слова: *распределенная вычислительная система, вычислительный кластер, база данных, информационная технология, файл сценария, генерация отчётов, система мониторинга.*

Введение

Одним из направлений повышения эффективности высокопроизводительных и распределенных вычислений является повышение оперативности получения необходимой для принятия решений о выборе стратегий планирования. Важным и актуальным является контроль над состоянием вычислительного кластера (кластеров) в составе мультикластерных и грид-систем (распределенных вычислительных систем, РВС). Вычислительный кластер представляет собой совокупность сервера (серверов) и вычислительных узлов, объединенных между собой коммуникационными каналами. Каждый вычислительный узел имеет в своём составе процессоры, архитектура которых включает в себя множество ядер, оперативную память и осуществляющих работу с помощью соответствующего программного обеспечения (ПО). Наиболее распространенным ПО являются системы пакетной обработки (СПО), в целом обеспечивающие решение таких задач, как определение состояния вычислительных узлов, обработку соответствующих запросов, поступающих от системы планирования, формирование ответных сообщений и их передачу системе планирования. В рассматриваемых в данной работе системах используются встроенные планировщики заданий или же планирование заданий производится внешними планировщиками (Maui/Moab) [1].

Кроме оптимизации физического расположения узлов кластерной системы проблемой является администрирование кластера, позволяющее обосновать и выбрать эффективные политики (стратегии) планирования заданий и ресурсов. В некоторых случаях это обеспечивается преимуществами архитектур с разделяемой памятью. В настоящее этот фактор позволил перейти к технологиям виртуализации в рамках реализации облачных вычислений.

При эксплуатации многопользовательского вычислительного кластера требуется доступ к большому количеству данных, в связи с чем вопросы решения задач планирования становятся достаточно серьезной проблемой. Поэтому распределение заданий среди вычислительных узлов, представленных ядрами процессоров и видеокартами, представляет собой значительную сложность. Важным аспектом является использование технологий получения и обработки данных о состоянии объектов контроля узлов и коммуникаций РВС, обеспечивающих необходимой информацией. При том все решения должны основываться на анализе достоверной и оперативной системной и служебной информации, получаемой администратором кластера или менеджером виртуальной организации [2].

Целью данного исследования является разработка информационной технологии для оценки показателей работы вычислительного кластера на основе оперативного формирования БД состояния объектов контроля и ее последующего анализа для принятия обоснованных решений о выборе стратегий планирования.

Общая схема построения информационной технологии РВС

Общая схема рассматриваемой информационной технологии базируется на использовании компонент систем мониторинга и данных, источниками которых являются лог-файлы локальных планировщиков и планировщиков локальных систем управления ресурсами (ЛСУР) [2–4].

Мониторинг осуществляется с помощью специализированного программного обеспечения –

программных расширений, устанавливаемых на серверах и узлах вычислительных кластеров РВС. Рассмотрим некоторые принципы работы системы мониторинга на примере системы Nagios [5]. В основе ее работы лежит использование удалённых программных агентов, которые установлены и запускаются на узлах кластера при помоши соответствующих протоколов и команд активации. Одним из таких агентов является Nagios Remote Plugin Executor (NRPE), задача которого заключается в том, чтобы осуществлять контроль над состоянием таких системных ресурсов как жёсткий диск, загрузка процессора, его ядер и балансировка загрузки, загрузка БД состояния узлов и заданий, выполняемых на них, критический уровень пропускной способности коммуникационных каналов и т. д. Для получения необходимой информации в контексте отслеживания состояния узлов система Nagios осуществляет удалённый опрос программного агента с использованием плагина check_nrpe, который производится через заданный интервал времени [5], определяемый соответствующими параметрами ОС управляющего узла РВС или ОС вычислительного кластера (сервера). Критически важным является контроль над состоянием узлов кластера – Up, Down, Unreachable, Pending; при этом развернутые сервисы могут быть в одном из следующих состояний: Ok. Warning, Unknown, Critical, Pending [1, 3, 4]. Для РВС с большим количество кластеров необходимость обработки такого объема служебной информации приводит к возможности критической загрузки каналов связи, и, как следствие, к увеличению временных задержек, связанных с передачей требуемой информации о состоянии узлов кластеров, заданий и нагрузки на БД.

Другим средством и источником организации информационного обеспечения для контроля над локальными ресурсами РВС и выполняемыми заданиями являются лог-файлы внешних планировщиков и планировщиков ЛСУР (например, Maui//Torque). Их основная функция состоит в том, чтобы осуществлять распределение заданий среди функциональных компонент РВС. Его работа может быть отслежена с помощью лог-файлов, которые генерируются системой для непосредственного чтения и последующего анализа и обработки данных согласно задачам системного администратора кластера (РВС). Пакет Maui является планировщиком заданий, поддерживающим множество политик планирования, позволяющий динамически изменять приоритеты заданий и определять исключения. Результаты работы данного пакета отражаются в лог-файлах, которые генерируются по требованию, то есть в промежутки времени, определяемые в конфигурационном файле, или же через определенные периоды (циклы) времени, определяемые длительностью цикла планирования.

Модель планирования и методы получения и обработки данных

Согласно двухуровневой модели планирования [2, 3], задания из входной очереди поступают в пул, при этом задаётся соответствующий период (длительность цикла) планирования и известна интенсивность потока. Пул является пакетом заданий и является стеком для временного хранения и последующего планирования заданий на доступные и свободные кластеры РВС. Размер пула определяется интенсивностью входного рабочего потока реальной системы (кластера). При этом задания из пула выгружаются через интервалы времени, определяемые периодом планирования, время которого зависит от размера пула, то есть от количества заданий, поступивших на блок планирования, и количества доступных и свободных на момент планирования вычислительных ресурсов РВС и назначаются на локальные ресурсы на основе решения задачи линейного целочисленного программирования [2]. Период планирования должен обеспечивать также и отсутствие простоя ресурсов, что определяется расчётом минимального или среднего времени освобождения ресурсов РВС [2]. Для эффективного контроля над состоянием ресурсов РВС необходимо построить такую последовательность их опроса (расписание), которая бы доставляла минимум общему времени опроса и доставки информации о завершении процессов активации и запуска программных агентов на вычислительных кластерах РВС [6].

Для решения данной задачи используется метод определения кратчайших путей и кратчайших гамильтоновых циклов в графе, разработанный в работе [6].

Реализация метода решения задачи минимизации суммарной задержки при проведении опроса и запуска удалённых программных агентов на узлах РВС включает 2 этапа:

Этап 1 – построение стянутого дерева всех путей и распараллеливание процесса решения заданий на узлах системы, что отчасти решает проблему потребности разработки специализированной архитектуры для решения поставленных задач [2].

Этап 2 – определение кратчайших гамильтоновых циклов, имеющий полиномиальную временную сложность, который позволяет достаточно эффективно решить задачу определения последовательности активации и запуска удалённых сервисов для осуществления контроля над

объектами – вычислительными ресурсами, коммуникациями и заданиями в режиме реального времени [3, 6]. Администратор виртуальной организации контролирует состояние вычислительных узлов РВС с помощью управляющего узла [3, 6]. Система Nagios через заданные интервалы времени запускает сервисы с помощью команд. Сервисы, выполняя заданные команды, идентифицируют состояния множества характеристик узлов и заданий и возвращают результаты запросов администратору. Таким образом, администратор получает информацию в оперативном режиме [3, 4].

Для получения информации о состоянии узлов и выполняемых заданий на вычислительном кластере используются данные лог-файлов планировщика Maui, представленные в формате SWF (Standard Workload Format). Для оперативного контроля над состоянием и формирования стратегий планирования администратор кластера запускает на выполнение скрипт для заполнения базы данных и скрипт на создание отчётов на основании полученных данных. В результате он получает статистическую информацию о работе кластера за определённый промежуток времени, который определяется периодом планирования. Проанализировав полученную информацию, администратор кластера может принять решение относительно выбора алгоритма и стратегий планирования.

Сбор и обработка данных

Имея информацию о формате необходимых исходных данных, представленных в лог-файлах, можно приступить к разработке информационной технологии, осуществляющей их сбор и обработку.

Результаты оценки состояния объектов контроля вычислительного кластера отражены соответствующими данными в наборах лог-файлов, необходимых администратору кластера. Объём этих данных является существенным, поэтому разбираться с ними в «сыром» виде представляется достаточно сложно. Для преодоления этого недостатка на первом этапе используется файл сценария, написанный на языке bash. Назначение данного исполняемого файла состоит в том, чтобы произвести синтаксический анализ входных данных из логов, чтобы извлечь требуемую для оценки результатов работы кластера информацию. На следующем этапе обработки реализуется организация управления хранением извлечённых на предыдущем этапе данных. Данный этап реализован с помощью использования базы данных под управлением PostgreSQL – объектно-реляционной СУБД [3, 4]. Среди её достоинств следует выделить возможность быстрого и простого развёртывания в распределённой среде. База данных состоит из таблиц и связей между ними, которые представляют собой сущности, описываемые в логах. Таким образом, результатом работы скрипта является заполненная данными база данных, спроектированная в соответствии с предварительно разработанной моделью предметной области [3, 4]. Для их окончательного представления в удобном для последующего использования виде реализован третий этап. Он заключается в выполнении составных запросов на выборку данных из базы данных с целью последующего получения значений показателей работы кластера РВС: коэффициента качества обслуживания, среднего коэффициента использования ресурсов и т. д. Для этого был определён набор функций в базе данных, которые вызываются с нужными параметрами из второго файла сценария для генерации текстовых файлов-отчётов, о показателях функционирования кластера за выбранные периоды времени.

Алгоритм реализации информационной технологии

Предлагаемая информационная технология реализована на основе следующего алгоритма.

Шаг 1. Заполнение БД данными лог-файлов планировщика Maui Cluster Scheduler под управлением СУБД PostrgreSQL в среде операционной системы Linux. Для этого использован исполняемый файл сценария, который содержит скрипт, написанный на языке команд bash (script.sh).

Шаг 2. Формирование отчетов, построенных на данных, представленных в заполненной базе данных, на основе использования второго файла сценария (report.sh).

Каждое задание описывается записью, имеющей в своём составе порядка 44 полей. Таким образом, администратор кластера получает статистическую информацию о работе кластера за некоторый период времени, данные относительно которого содержатся в базе данных. Проанализировал информацию двух видов, администратор получает возможность принять оптимальное решение о выборе алгоритма и стратегии планирования. Диаграмма последовательности технологии заполнения БД и формирования отчетов в нотации языка UML приведена на рис. 1.

Описание лог-файла Maui

Лог-файлы локального планировщика Maui создаются благодаря трассировки нагрузки (Workload traces) в формате SWF. Трассировка нагрузки полностью описывает все

соответствующие аспекты планирования пакетных заданий, в том числе необходимые для работы и используемые ресурсы, время всех основных событий по планированию полученных заданий (например, время помещения в очередь, время начала выполнения и т. д.) и конфигурацию среды выполнения. Состав параметров, которые в нем содержатся, представлены в табл. 1 [1].

Результаты практической реализации информационной технологии

Данная информационная технология была реализована на вычислительном кластере с целью получения результатов в ходе её практического применения для различных интервалов времени работы вычислительного кластера. В качестве первичных источников данных были использованы лог-файлы, сгенерированные в процессе планирования заданий планировщиком Maui в течении нескольких месяцев. В качестве СУБД для хранения и обработки статистических данных использована PostgreSQL 9.3. Для анализа результатов практической реализации предлагаемой информационной технологии выбраны 2 временных интервала длительностью по 7 дней.

Для каждого интервала в тестовом лог-файле формируется 2 500 – 3 000 записей, которые необходимо хранить в базе данных (объем требуемой для хранения одной записи памяти составляет 100 Б). Время их обработки (формирования БД) составляет порядка 45–60 сек. (до 1 мин.). После заполнения базы данных выполняется второй файл сценария для генерации отчётов значений параметров, представленных в табл. 2, для хранения одной записи которого требуется 100 Б. При этом общее время экспорта и обработки данных двух временных интервалов, приведенных выше, для расчета показателей работы кластера не превысило 15 минут, объем требуемой для хранения данных лог-файлов и сгенерированных отчетов памяти не превысил 1 МБ.

Состав показателей генерируемых отчетов представлен в табл. 3; результаты расчетов показателей функционирования кластера на тестовых данных лог-файлов приведены в табл. 4.

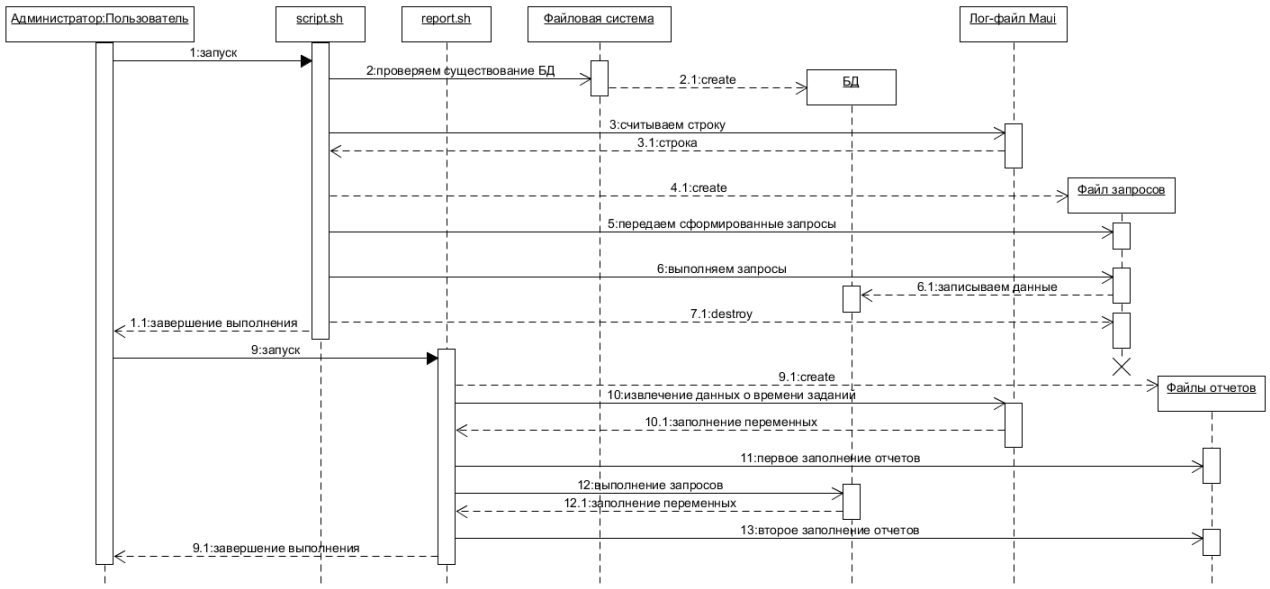

Рис. 1. Диаграмма последовательности технологии заполнения БД и формирования отчетов

Таблица 1

Состав параметров лог-файла Maui

Идентификатор	Тип данных	Значение по умолчанию	Описание
JobID	STRING	-	Идентификатор задания. Должен быть уникальным.
Nodes Requested	INTEGER	0	Количество необходимых узлов (0 = количество запросов на узел не определено).
Tasks Requested	INTEGER	1	Количество необходимых заданий.
User Name	STRING	-	Имя пользователя, отправившего задание.
Group Name	STRING	-	Первичная группа пользователя, отправившего задание.
Wallclock Limit	INTEGER	1	Максимальная допустимая продолжительность работы, сек.
Job Completion State	STRING	Completed	Возможные значения: Completed, Removed, NotRun.
Required Class	STRING	1	Класс/очередь, которые необходимы для задания, указанного в качестве списка квадратной скобкой <очередь> [:<экземпляр очереди>].
Submission Time	INTEGER	0	Unix-время, когда работа помещена очередь.
Dispatch Time	INTEGER	0	Unix-время, когда планировщик начал искать место выполнения работы.
Start Time	INTEGER	0	Unix-время, когда работа начала выполняться (Dispatch Time).
Completion Time	INTEGER	0	Unix-время, когда работа завершилась.
Required Network Adapter	STRING	-	Имя необходимого сетевого адаптера.
Required Node Architecture	STRING	-	Архитектура необходимого узла.
Required Node Operating System	STRING	-	ОС необходимого узла.
Required Node Memory Comparison	Знак порівняння (>, >=, =, <=, <)	>=	Сравнения, для определения соответствия памяти узла.
Required Node Memory	INTEGER	0	Объем необходимой оперативной памяти (МБ) на каждом узле.
Required Node Disk Comparison	Знак порівняння (>, >=, =, <=, <)	>=	Сравнения, для определения соответствия диска узла.
Required Node Disk	INTEGER	0	Количество необходимых настроенных локальных дисков (МБ) на каждый узел.
Required Node Attributes	STRING	-	В квадратных скобках находится список компонент узла, необходимых для работы, если он указан. Пример: [fast] [ethernet].
System Queue Time	INTEGER	0	Unix-время, когда работа выполнила все политики справедливости (fairness policies).
Tasks Allocated	INTEGER	Tasks requested	Количество заданий, фактически выделенных для работы.

Окончание табл. 1

Идентификатор	Тип данных	Значение по умолчанию	Описание
Required Tasks Per Node	INTEGER	-1	Количество заданий на один узел, необходимых для работы.
QOS	STRING	-	Качество обслуживания для запрошенного узла и узла получателя. Формат: [<узел-источник>] [узел-получатель].
JobFlags	STRING	-	[BACKFILL] [BENCHMARK][PREEMPTEE].
Account Name	STRING	-	Имя учетной записи, связанной с работой.
Executable	STRING	-	Название работы исполняемого файла.
Comment	STRING	-	Заданный список диспетчера ресурсов по атрибутам работы.
Bypass Count	INTEGER	-1	Количество рабочих часов, которые были использованы низкоприоритетными заданиями по заполнению.
ProcSeconds Utilized	DOUBLE	0	Количество процессорных секунд, которые на самом деле используются работой.
Partition Name	STRING	-	Название раздела, в котором было выполнено задание.
Dedicated Processors per Task	INTEGER	1	Количество процессоров, необходимых для решения каждой задачи.
Dedicated Memory per Task	INTEGER	0	Объем оперативной памяти (МБ), необходимый для решения каждой задачи.
Dedicated Disk per Task	INTEGER	0	Размер локального диска (МБ), необходимого для решения каждой задачи.
Dedicated Swap per Task	INTEGER	0	Объем виртуальной памяти (МБ), необходимый для каждой задачи.
Start Date	INTEGER	0	Unix-время, когда работа может начать выполняться.
End Date	INTEGER	0	Unix-время, когда работа должна была быть завершена
Allocated Host List	STRING	-	Список использованных узлов.
Resource Manager Name	STRING	-	Имя менеджера ресурса.
Required ,Host Mask	STRING		Список узлов, необходимых для выполнения работы. Используется планировщиком, если количество заданий меньше, чем количество хостов.
Reservation	STRING		Резервируемое имя, которое нужно для работы, если оно указано.
Set Description	STRING		Устанавливает ограничения, необходимые для узла в виде: <ограничение>:<тип>[:<список>]. Ограничение принимает значения: ONEOF, FIRSTOF, ANYOF. Тип принимает значение: PROCSPEED, FEATURE, NETWORK. Список содержит дополнительные атрибуты. Например: ONEOF:PROCSPEED:350:450:500
Application Simulator Data	STRING		Название модуля симулятора приложения и связанные с ним данные конфигурации.
RESERVED FIELD 1	STRING		Зарезервированное значение.

Таблица 2

Состав показателей функционирования кластера для построения отчетов

Идентификатор	Значение
$t_{общ}$	Общий интервал работы РВС (1)
$t_{общ\_очередь}$	Суммарное время нахождения заданий в очереди (2)
$t_{общ\_вып}$	Суммарная длительность выполнения заданий (3)
$k_{использования}$	Коэффициент использования РВС (4)
k_{over}	Средний коэффициент загруженности (использования) узлов (5)
$k_{обслуж}$	Коэффициент качества обслуживания за заданный промежуток времени (6)

$$t_{общ} = MAX(t_{окон}) - MIN(t_{очередь}), \quad (1)$$

где $t_{окон}$ – время окончания выполнения задания;

$t_{очередь}$ – время помещения задания в очередь.

$$t_{общ\_очередь} = \sum_{i=1}^{m} (t\_нач_i - t\_очередь_i), \quad (2)$$

где $t\_нач_i$ – время перехода *i*-го задания на выполнение;

$t\_очередь_i$ – время перехода *i*-го задания в очередь.

$$t_{общ\_вып} = \sum_{i=1}^{m} (t\_окон_i - t\_нач_i), \quad (3)$$

где $t\_завер_i$ – время завершения выполнения i-го задания.

$$k_{использования} = \frac{t_{общ\_вып}}{t_{общ}}. \quad (4)$$

$$k_{over} = \frac{k_{использования}}{n}, \quad (5)$$

где n – общее количество узлов.

$$k_{обслуж} = \frac{t_{общ\_очередь} \cdot m_{вып}}{m_{очередь} \cdot t_{общ\_вып}}, \quad (6)$$

где $m_{вып}$ – количество выполненных задач за промежуток времени;

$m_{очередь}$ – количество заданий в очереди на промежутке времени.

Таблица 3

Типы генерированных отчетов о показателях функционирования вычислительного кластера

Название отчета	Описание
CountJobsInQueue	Данные о количестве заданий в очереди на определенный момент времени
CountJobsInProgress	Данные о количестве заданий, которые выполняются в определенный момент времени
CountJobsCompeleted	Данные о количестве выполненных заданий на определенный момент времени
CoefficientOfServiceQuality	Значение коэффициента качества обслуживания
ResUptimeAndCoeffOfUtilization	Значение среднего коэффициента использования

Таблица 4

Значения показателей функционирования вычислительного кластера

Показатель	Интервал 1	Интервал 2
$k_{обслуж}$	0.4847	0.7225
Количество заданий на этапе выполнения в начале заданного временного интервала	294	35
Количество заданий в очереди на момент начала заданного временного интервала	184	47
$k_{использования}$	0.8892	0.8888

Выводы

Рассмотрена общая схема информационной технологии для оценки показателей функционирования вычислительного кластера на основе формирования и обработки данных о состоянии вычислительного кластера РВС, сформированных как результат запуска удалённых программных агентов и экспорта данных лог-файлов локального планировщика Maui. На основе полученных данных о работе объектов контроля кластера осуществляется заполнение базы данных системы с помощью разработанного файла сценария. Для оценки показателей работы кластера и их последующего анализа на основе сформированной базы данных о вычислительных ресурсах, выполняемых заданиях, включая очереди на узлы, входящие в кластер, программно реализован процесс генерации отчётов.

В результате реализации данной технологии системный администратор кластера (менеджер виртуальной организации) получает возможность проводить оперативный анализ состояния кластерной системы с требуемым временным периодом, позволяющий принимать соответствующие решения о выборе стратегии планирования и вносить изменения в настроечные параметры планировщика в режиме реального времени.

В дальнейшем предполагается продолжить исследования в направлении оптимизации объема передаваемой служебной и хранимой информации для обработки данных лог-файлов и генерируемых отчетов в зависимости от выбираемого периода формирования лог-файлов и периода планирования для повышения эффективности и обоснования выбираемых системными администраторами кластеров и менеджерами виртуальных организаций, функционирующих в составе РВС, стратегий планирования.

Список литературы

1. *Maui Trace File Format, version 310 [Электронный ресурс]. – Режим доступа: http://docs.adaptivecomputing.com/maui/trace.php.*
2. *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.*
3. *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. – Випуск 1(33). – С. 111–115.*
4. *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. – 2015. – Вип. 12 (137). – С.134–139.*
5. *Nagios – The Industry Standard in IT Infrastructure Monitoring [Электронный ресурс]. – Режим доступа: http://www.nagios.org.*
6. *Листровой С. В. Разработка метода мониторинга распределенной вычислительной системы на основе определения кратчайших путей и кратчайших гамильтоновых циклов в графе / С. В. Листровой, С. В. Минухин, Е. С. Листровая // Восточно-Европейский журнал передовых технологий. – 2015. – 6/4 (78).– С. 32–45.*

Рецензент: Алексеев В. О., д.т.н., профессор, профессор кафедры информационных систем ХНЭУ им. С. Кузнеца.

Інформаційна технологія оцінки показників функціонування обчислювального кластера розподіленої системи

С. В. Мінухін, К. С. Молчанов, М. Г. Сизранцев

Запропоновано інформаційну технологію для вирішення завдань планування завдань на обчислювальному кластері, що базується на забезпеченні інформацією даними про стан вузлів і завдань, які формують базу даних на основі лог-файлів локального планувальника Maui. Розглянуто схему її реалізації з використанням розроблених файлів сценарію заповнення бази даних і формування звітів про показники роботи кластера за запитом за задані періоди часу.

Ключові слова: *розподілена обчислювальна система, обчислювальний кластер, база даних, інформаційна технологія, файл сценарію, генерація звітів, система моніторингу.*

Information technology indicators for the evaluation of a distributed computing cluster system

S. V. Minukhin, K. S. Molchanov, M. G. Syzrantsev

An information technology solutions for job scheduling tasks on a computing cluster based on the provision of information, data on the state of the nodes and tasks, forming a database of local scheduler Maui log files is developed. A scheme for its implementation developed using the script file database filling and reporting on the performance of the cluster on demand for specified periods of time.

Keywords: *distributed computing system, the computing cluster, database, information technology, a script file, generation of reports, monitoring system.*