ПРОГРАМНО-ТЕХНІЧНІ КОМПЛЕКСИ

ЧЖЕНБИН ХУ, В.Е. МУХИН, Я.И. КОРНАГА, О.Ю. ГЕРАСИМЕНКО

УДК 004.04

УПРАВЛЕНИЕ РЕСУРСАМИ РАСПРЕДЕЛЕННОЙ КОМПЬЮТЕРНОЙ СИСТЕМЫ С УЧЕТОМ УРОВНЯ ДОВЕРИЯ К ВЫЧИСЛИТЕЛЬНЫМ КОМПОНЕНТАМ

Аннотация. Рассматривается обеспечение безопасной обработки данных в распределенных компьютерных системах (РКС), что является важным для выполнения определенного класса задач. Предложен подход к управлению ресурсами РКС, который в соответствии с требованиями пользователя позволяет учесть как временные затраты на выполнение задания, так и уровень защищенности ресурсов, привлекаемых для его выполнения.

Ключевые слова: распределенные вычисления, управление ресурсами, планирование задач, безопасная обработка данных, мониторинг состояния вычислительного узла, локальный агент данных.

ВВЕДЕНИЕ

В настоящее время стратегия выполнения сложных вычислительных задач сместилась от использования суперкомпьютеров к использованию распределенных компьютерных систем (РКС), в частности GRID-систем, которые позволяют объединить географически распределенные вычислительные ресурсы для выполнения сложных вычислений. Кроме того, исследователи все больше уделяют внимание привлечению мобильных устройств к выполнению вычислений в РКС, например в GRID-системах [1, 2]. В то же время реализация распределенных компьютерных систем требует решения ряда сложных задач, одной из которых является планирование выполнения задач и распределение ресурсов между ними. Эффективное их распределение является одним из основных требований к высокопроизводительным вычислительным системам, а механизм распределения ресурсов считается центральной темой высокопроизводительных вычислений [3].

Кроме эффективного распределения ресурсов, процедура управления ресурсами РКС призвана обеспечить требуемый пользователем уровень качества обслуживания (QoS). Качество обслуживания определяется набором параметров, основными из которых принято считать время выполнения задачи и стоимость услуги. Тем не менее, используя РКС для обработки своих данных, пользователь должен быть уверен в обеспечении их конфиденциальности и целостности, что свидетельствует о необходимости учитывать безопасность обработки данных в процессе управления ресурсами распределенной системы.

ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ К УПРАВЛЕНИЮ РЕСУРСАМИ РКС

В настоящее время разработано множество подходов к управлению ресурсами в РКС. Планирование задач для различных видов РКС имеет свои особенности

в связи с разной архитектурой систем и их функциональными показателями. Так, в работе [3] выделяются три вида высокопроизводительных вычислительных систем: кластеры, Grid- и Cloud-системы, а также рассматривается планирование задач в каждой из них. Сравнительная характеристика различных видов РКС, в том числе и в вопросах управления ресурсами, представлена в работах [4, 5].

Существует много подходов, применяемых для управления ресурсами распределенной системы. Так, в работе [6] систематизирована информация о различных подходах в распределении ресурсов Grid-системы, кратко описаны особенности каждого из них. Отмечено, что несмотря на наличие большого количества публикаций, посвященных механизмам управления ресурсами Grid-системы, до сих пор эти механизмы нуждаются в улучшении.

В работе [7] представлена стратегия планирования задач типа поток работ в Cloud-среде, основанная на доверии. Данный механизм планирования учитывает время выполнения задачи, стоимость услуги, надежность и защищенность сервисов, а приложение типа поток работ представляет набор атомарных взаимозависимых задач, данные для выполнения которых находятся в хранилищах, являющихся сервисами Cloud-среды.

В работе [8] представлен механизм планирования задач с учетом уровня защищенности ресурсов системы для распределенной вычислительной среды с интенсивным обменом данных. Недостаток такого подхода — отсутствие учета изменения уровня защищенности ресурсов в процессе функционирования распределенной среды.

Одним из подходов к управлению ресурсами РКС является подход, основанный на анализе надежности компонентов РКС, причем как аппаратных, так и программных. Действительно, в случае частого отказа компонентов системы части задания (а также входных данных для них), которые были не выполнены ввиду отказа, должны передаваться на другие вычислительные узлы (ВУ) для повторного запуска, что соответственно приводит к увеличению времени обработки задания в целом. В [9–11] разработаны механизмы управления ресурсами РКС на основе этого подхода.

Динамический децентрализованный алгоритм управления ресурсами CASA (Community-Aware Scheduling Algorithm) представлен в статье [12]. Работа алгоритма базируется на взаимодействии вычислительных узлов посредством обмена сообщениями и совместном поведении как «общности».

Одним из наиболее распространенных в управлении ресурсами РКС является подход, базирующийся на обеспечении требуемого показателя QoS, который соединяет в себе несколько критериев и определяет уровень удовлетворенности клиента качеством оказанных ему услуг. Показатель QoS выражается измерением качества таких параметров, как полное время выполнения задания, задержка и стоимость выполнения задания, уровень потери пакетов, выработка и надежность системы [13]. Однако данный перечень не является исчерпывающим. Такой подход используется в различных видах РКС. В работах [13–16] представлены механизмы планирования, ориентированные на обеспечение требуемого показателя QoS.

Существует также подход, применяемый в управлении ресурсами распределенных систем. В таком подходе при планировании должны быть учтены параметры коммуникационной среды, а также объем и местонахождение передаваемых данных. Несмотря на то, что алгоритм планирования основывается, как правило, на одном из указанных факторов, все эти факторы взаимосвязаны между собой, поэтому принято решение ограничиться рассмотрением одного общего подхода. В связи с тем, что распределенные системы часто используются как хранилища данных, механизмы управления ресурсами, основанные на этом подходе, находят все более широкое применение. В работах [17–21] представлены алгоритмы, основанные на данном подходе.

Для управления ресурсами в распределенных системах разработаны разные экономические модели. Например, в [22] рассматриваются экономические модели для управления ресурсами в Grid-системе, отмечены их преимущества и недостатки.

Распределенные системы большого масштаба, например Grid-системы, потребляют значительное количество энергии в связи с их огромным размером. Это привело к созданию алгоритмов планирования заданий, направленных на уменьшение количества потребляемой энергии. В работах [23, 24] представлены такие алгоритмы.

ПОСТАНОВКА ЗАДАЧИ

Управление ресурсами распределенной системы обусловливает эффективность используемых ресурсов и гарантирует качество обслуживания, которое обеспечивается для пользователей [3]. Однако часто возникает ситуация, когда пользователю при выполнении некоторой задачи более важно обеспечить безопасную обработку данных, а не получить быстрый результат. Иными словами, пользователь считает возможным потратить больше времени и средств для поддержки безопасной обработки данных, если предусмотрена плата за пользование вычислительной системой. Таким образом, целесообразно разработать планирование задач в РКС, при котором учитывались бы как временные параметры качества обслуживания, так и безопасность обработки данных в системе.

В любой системе реализация средств безопасности приводит к тому, что часть ее производительности неизбежно расходуется на функционирование средств безопасности, а поэтому обеспечение надлежащего уровня QoS является особенно важным. Так как основную работу по управлению ресурсами РКС выполняет планировщик распределенной системы, то разработка механизма планирования задач в РКС с учетом безопасности ресурсов и обеспечения надлежащего уровня QoS является в настоящее время важным и актуальным вопросом.

Поскольку существует несколько видов РКС, каждая из которых имеет свои особенностяи построения, управления и функционирования, то необходимо уточнить, что следует понимать под РКС в настоящей работе. В данном исследовании будем представлять РКС как совокупность гетерогенных географически распределенных ВУ, которые в качестве системы передачи данных используют компьютерную сеть и объединяются (взаимодействуют между собой) для выполнения сложных вычислительных задач, т.е. задач с высокой степенью распараллеливания. Задача представляет собой поток работ, но размещение компонентов задачи на вычислительные узлы системы не является предметом данного исследования. Показатель QoS может учитывать множество параметров. Под QoS в данной статье будем понимать временные затраты на выполнение задачи и обеспечение надлежащего уровня безопасной обработки данных.

Таким образом, существует РКС с гетерогенными ВУ и гетерогенными каналами передачи данных, причем для системы характерна многоканальность, которая может быть физической и топологической. Под физической многоканальностью будем понимать реализацию частотного или временного разделения единого физического канала передачи данных. В качестве такого канала могут выступать волоконно-оптические линии связи, физическая природа которых предусматривает дифференциацию мод по частотам, в том числе дуплексирование. Под топологической многоканальностью будем понимать реализацию множественных альтернативных маршрутов передачи данных между узлами РКС, которые соединены между собой топологически избыточной системой связи. Возможно комбинирование физической и топологической многоканальности для формирования высокоскоростной надежной и защищенной среды передачи данных. Структура сети, объединяющей ВУ, заведомо неизвестна и может изменяться во время функционирования РКС.

Разработаем механизм управления ресурсами распределенной компьютерной системы на основе мониторинга состояния вычислительных компонентов

РКС в целях обеспечения надлежащего уровня качества обслуживания по временным затратам на выполнение задачи и необходимой степени безопасной обработки данных.

РАЗРАБОТКА МЕХАНИЗМА УПРАВЛЕНИЯ РЕСУРСАМИ РКС С УЧЕТОМ БЕЗОПАСНОЙ ОБРАБОТКИ ДАННЫХ

Существуют различные виды планирования в РКС, а именно централизованное, иерархическое и децентрализованное. Централизованное планирование предполагает наличие одного планировщика (метапланировщика или глобального планировщика), который владеет информацией о всех ресурсах системы и выполняет планирование всех задач. Такой планировщик легче реализовать, но при выходе его из строя блокируется работа всей РКС.

Иерархическое планирование подразумевает наличие в РКС одного глобального планировщика и нескольких локальных планировщиков. Глобальный планировщик контролирует процесс планирования и передает задания локальным планировщикам.

При децентрализованном планировании предполагается наличие не одного центрального планировщика, контролировавшего процесс планирования во всей системе, а нескольких планировщиков, которые, взаимодействуя между собой, совместно выполняли бы планирование заданий. Такое планирование в реализации достаточно сложное и, кроме того, оно может приводить к снижению производительности системы в целом.

На первом этапе разработаем механизм планирования на базе централизованной архитектуры ввиду простоты ее реализации. Задания на выполнение в РКС будут поступать одному центральному планировщику, который в дальнейшем будем называть метапланировщиком. Он выполняет планирование, подбор ресурсов, контроль выполнения заданий и другие действия, присущие планировщику. Сформулируем основную задачу метапланировщика, которая состоит в подборе безопасных ресуров для выполнения задачи и обеспечении при этом приемлемых временных затрат на выполнение задания:

а) определить такое количество узлов системы, при котором достигается указанный уровень безопасной обработки данных и оптимальное время выполнения задачи;

б) выбрать из всего имеющегося набора ресурсов РКС те, которые в своей комбинации обеспечат указанный уровень безопасной обработки данных и оптимальное время выполнения задачи.

Решение этого вопроса требует наличия у метапланировщика знаний о функциональных характеристиках компонентов РКС, таких как производительность ВУ, скорость передачи данных по каналам связи системы и защищенность компонентов распределенной системы. Для получения функциональных характеристик компонентов РКС метапланировщику необходимо получить информацию о состоянии ВУ системы и каналов связи системы. Таким образом, вычислительный узел должен предоставить метапланировщику данные о своих параметрах производительности, защищенности и скорости передачи данных ему. Для уменьшения нагрузки на метапланировщика целесообразно принять, что каждый узел должен определять свои параметры самостоятельно в синхронном или асинхронном режиме. Такой подход позволит освободить метапланировщика от сбора и анализа данных по каждому узлу в отдельности, ситуация же выхода из строя ВУ и отсутствие информации не повлияют на работу системы. Следовательно, на каждом ВУ системы необходимо разместить локальный агент данных (ЛАД), в функции которого входит сбор, анализ и предоставление метапланировщику информации о текущем состоянии ВУ. Логическая схема связей компонентов системы представлена на рис. 1.

Информацию о состоянии ресурсов метапланировщик может получить непосредственно от каждого узла системы (так как рассматривается централизо-

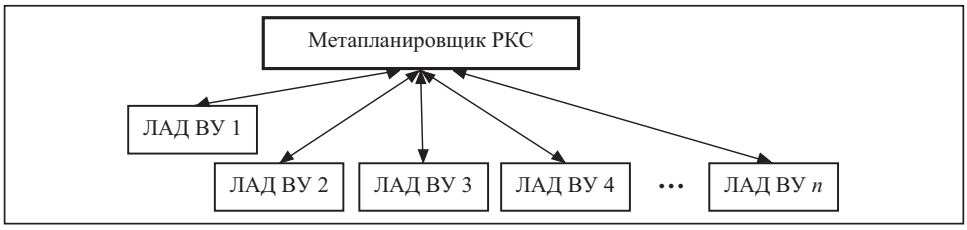

Рис. 1. Логическая схема связей метапланировщика и ЛАД вычислительных узлов в распределенной системе

ванный метапланировщик РКС), в случае масштабной системы возможна реализация соответствующих сервисов, обеспечивающих такую функциональность, например, с использованием GMA (Grid Monitoring Architecture) [25].

Для запуска задания на выполнение в такой системе от пользователя требуется задать параметры ресурсов РКС, которые будут использованы для выполнения задачи. Как уже отмечалось, метапланировщик может оценить три параметра системы: производительность, скорость передачи данных и защищенность, но нецелесообразно, чтобы все оценки определял пользователь. Защищенность системы указывается пользователем в пределах от 0 до 1, поскольку именно пользователь определяет уровень ценности своих данных. Производительность и скорость передачи данных нецелесообразно представлять конкретными значениями, так как пользователю могут быть неизвестны параметры составляющих распределенной системы. Эти параметры являются основными факторами, влияющими на скорость выполнения задания, поэтому их следует учитывать как основные факторы QoS. Другой возможный вариант — использовать максимально доступные характеристики РКС для скорейшего выполнения задач. Оценка стоимости предоставленных услуг РКС в соответствии с уровнем безопасной обработки данных и QoS выходит за рамки данного исследования.

СТРУКТУРА И ФУНКЦИОНИРОВАНИЕ ЛАД

Локальный агент данных (Local Data Agent, LDA) предназначен для отслеживания параметров узла и каналов системы (так как система многоканальная) и передачи их метапланировщику. Передача параметров может осуществляться синхронно через равные промежутки времени или же асинхронно по требованию метапланировщика. ЛАД осуществляет определение таких параметров, как производительность, скорость передачи данных и защищенность. Информация о состоянии узла определяется через установленные промежутки времени и сохраняется локально в памяти ВУ, например в flash-памяти; назовем место хранения данных хранилищем данных. Поскольку система является динамической по своей природе, то хранить такую информацию за длительный период времени нецелесообразно. Исключение может составлять информация о защищенности узла, период хранения которой может быть более длительным ввиду анализа действий узла в РКС и их влияния на работу распределенной системы.

Определение производительности ВУ. Производительность ВУ определяется на основе системных характеристик компьютера, для этого могут использоваться определенные системные утилиты. Будем считать, что ВУ являются одноядерными и однопроцессорными. В настоящее время существует большое количество утилит, которые позволяют оценить производительность компьютера. Некоторые из них разрабатываются производителями процессоров, а другие — производителями программного обеспечения (ПО) для тестирования и анализа системных характеристик компьютера. Тесты могут определять производительность в MIPS или FLOPS. Учитывая, что замена процессоров не является частым явлением, можно считать показатель производительности константой, который целесообразно определить один раз (например, при запуске системы) и сохранить для передачи метапланировщику.

Определение скорости передачи данных. Скорость передачи данных по каналу связи в многоканальной системе зависит от нескольких факторов. Основными из них можно считать пропускную способность канала и задержку при прохождении данных через сетевую аппаратуру. Для определения скорости передачи данных по каналу целесообразно использовать специализированные утилиты. Данный параметр может значительно изменяться во времени, например в зависимости от загруженности сети, а поэтому определять его необходимо периодически с занесением результатов в хранилище данных. По требованию метапланировщика может передаваться последнее значение, полученное в результате тестирования, или снова запускаться утилита тестирования и передаваться полученное текущее значение теста. Важно отметить, что используемая утилита должна поддерживать определение скорости передачи данных для многоканальной среды передачи данных.

Определение защищенности ВУ. Защищенность ВУ в каждом конкретном случае может определяться по-разному, поскольку нет однозначного параметра, которым она измеряется. В данном случае предлагаем определять защищенность вычислительного узла РКС как уровень доверия к узлу, установленный на данный момент. Уровень доверия определяется максимальным уровнем ценности информации, которая может быть передана на обработку данному узлу системы.

Новому вычислительному узлу при подключении к РКС назначается начальный уровень доверия (минимальный), который может быть увеличен или уменьшен в зависимости от дальнейшего действия ВУ. В процессе функционирования узла в составе РКС постоянно выполняется мониторинг параметров его функционирования и данных, передаваемых узлом в РКС. Необходимость в постоянном мониторинге вызвана возможностью возникновения ситуации, когда узел с низким уровнем доверия фактически выступает как агент вторжения, например реализует несанкционированный доступ к ресурсам компьютерной системы.

Защищенность каналов РКС также может быть учтена аналогичным образом, но поскольку она значительно зависит от технических и технологических средств защиты каналов связи, то в данной ситуации она учитываться не будет, так как эта тема требует дополнительного исследования. Будем считать, что в случае необходимости защиты данных при передаче их по сети пользователи могут применять криптографические средства защиты информации, например шифрование данных.

Функционирование ЛАД. При загрузке операционной системы ВУ в автоматическом режиме выполняется запуск ЛАД. Функционирование ЛАД происходит в фоновом режиме. После загрузки выполняется запуск утилиты для определения производительности ВУ, полученный результат заносится в хранилище данных и остается неизменным до следующей перезагрузки системы.

Важной функцией ЛАД является мониторинг действий пользователя на предмет выявления несанкционированного доступа или атак, осуществляемых пользователем. Существуют различные подходы для выявления таких фактов [26, 27], но для уменьшения вычислительной нагрузки на узел предлагается использовать одну из числа наиболее простых моделей, базирующуюся на статистическом подходе. Результат мониторинга влияет на уровень доверия к ВУ, т.е. приводит к его повышению, если действия ВУ в системе корректные и не создают опасных ситуаций, или к снижению в противном случае. При постоянном выявлении угроз безопасности, связанных с ВУ, целесообразно такой узел исключить из РКС, как несущий угрозу всей системе. Определение скорости передачи данных и защищенности происходит периодически с занесением полученных результатов с маркером времени в хранилище данных.

На запрос метапланировщика ЛАД получает из хранилища данных последние параметры ВУ или определяет их заново и передает метапланировщику как ответ на запрос. Параметры производительность, скорость передачи данных и защищенность передаются в виде вектора соответственно из трех значений $\langle p_i, v_i, s_i \rangle$ для i-го вычислительного узла.

Структурная схема ЛАД. В соответствии с функциональными характеристиками ЛАД разработана его структурная схема (рис. 2).

Блок управления является основным элементом ЛАД, реализующим основные его функции: сбор информации от утилит, запись ее в хранилище данных, а также установление связи с метапланировщиком системы. Сбор информации осуществляется периодически с занесением ее в хранилище данных, которые записываются с маркером времени. Исключение составляет производительность, определяемая однократно сразу после запуска ЛАД, который автоматически загружается после запуска операционной системы вычислительного узла и записывается в хранилище данных. На запрос метапланировщика ЛАД определяет последние записанные в хранилище данных параметры ВУ и передает их в виде вектора $\langle p_i, v_i, s_i \rangle$ метапланировщику.

Хранилище данных предназначено для хранения параметров ВУ, собранных за определенный период времени, причем для каждого параметра период может существенно отличаться. Например, значение производительности нерационально записывать несколько раз, достаточно хранить одно текущее значение. Если исходить из условия, что производительность ВУ зависит от локальных задач, то целесообразно ее измерять с определенным периодом и записывать в хранилище данных аналогично другим параметрам. В дальнейшем эти данные могут быть использованы для прогнозирования фактической производительности ВУ.

Блок управления уровнем доверия предназначен для определения уровня доверия к действиям узла и передачи его блоку общего управления. При подключении нового ВУ к системе устанавливается минимальный уровень доверия к узлу. В дальнейшем проводится мониторинг действий узла с помощью соответствующей утилиты и уровень доверия корректируется в зависимости от полученных в результате мониторинга данных.

Мониторинг действий узла направлен на выявление случаев нарушения защищенности системы и их подсчет. На основании полученного числа нарушений защищенности рассчитывается уровень доверия к действиям ВУ системы, который функционально связан с уровнем защищенности системы. Определять уровень доверия для i-го ВУ можно следующим образом [28]:

$$Tn_i(t) = T_0 \frac{A_{\lim}}{A(t)+1},$$

где T_0 — начальный уровень доверия к узлу; $A(t)$ — количество нарушений защищенности, инициированных или связанных с i-м узлом на интервале времени $(0; t)$, $A_{\lim}$ — критическое число случаев нарушений защищенности на том же интервале времени.

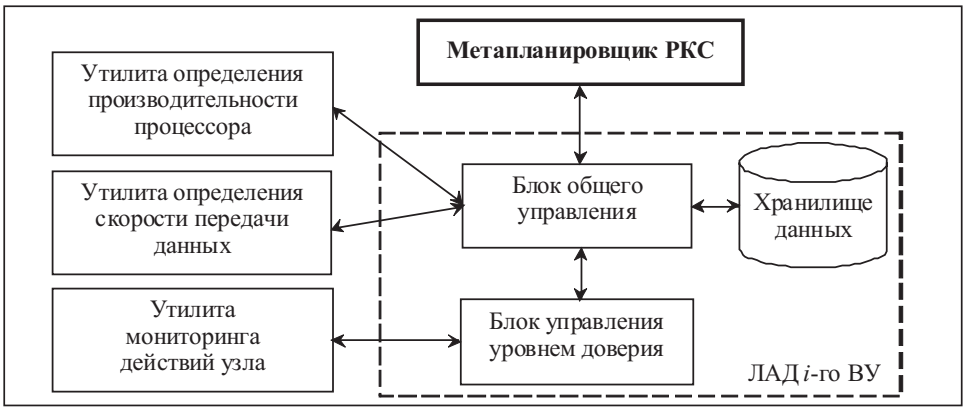

Рис. 2. Структурная схема ЛАД вычислительного узла РКС

События нарушения защищенности РКС могут быть преднамеренными и случайными действиями субъектов. В общем случае субъекты РКС и соответственно узлы, через которые они взаимодействуют с РКС, могут случайным образом осуществлять непреднамеренные ошибочные действия, формально связанные с попыткой нарушения защищенности. Управление параметром $A_{\lim}$, который фактически определяет предельно допустимое количество нарушений защищенности, вызванных случайными действиями субъектов, позволяет более гибко учитывать случаи нарушения защищенности.

В общем случае уровень доверия к *i*-му узлу в процессе его функционирования в составе РКС может повышаться, снижаться или оставаться постоянным. Так, если число случаев нарушения защищенности $A(t)$, связанных с *i*-м узлом на интервале наблюдения $(0; t)$, превышает установленное критическое число $A_{\lim}$ ($A(t) > A_{\lim}$), то уровень доверия к данному узлу снижается. Если число случаев нарушения защищенности меньше значения $A_{\lim}$ ($A(t) < A_{\lim}$), то уровень доверия к узлу повышается.

В результате такого механизма вычисления полученное значение уровня доверия может быть больше единицы. Поэтому в целях нормализации необходимо его значение разделить на значение максимально возможного уровня доверия, допустимого в системе. Максимально возможный уровень доверия, допустимый в системе, определяется администратором системы.

Для поддержки мониторинга действий вычислительных узлов используется несколько утилит.

Утилита определения производительности процессора — специализированное ПО, которое, используя соответствующие тестовые вычислительные нагрузки, определяет производительность ВУ в MIPS.

Утилита определения скорости передачи данных — специализированное ПО, определяющее на основе использования соответствующих тестов скорость передачи данных по каналам системы, к которым подключен ВУ. Необходимо отметить, что данные тесты должны поддерживать работу с многоканальной системой передачи данных.

Утилита мониторинга действий узла — специализированное ПО, которое выполняет мониторинг действий ВУ на основе статистического подхода.

ПОСТАНОВКА ЭКСПЕРИМЕНТА

Размещение на вычислительных узлах дополнительного программного обеспечения приводит к снижению их производительности и снижению производительности распределенной системы в целом. Так как система мониторинга действий пользователя должна функционировать непрерывно, наблюдая за целым списком параметров, то существует опасность значительного снижения производительности РКС при проведении вычислений. Таким образом, важно определить, какая часть производительности вычислительного узла РКС (например, в процентном соотношении к номинальной производительности) может использоваться в целях мониторинга, существенно не снижая при этом производительности системы в целом. Общепринято, что материальные затраты на обеспечение физической безопасности материальных ценностей не должны превышать 10 % их стоимости. Исходя из этого, допустимое снижение производительности распределенной системы примем в пределах 10 %, т.е. производительность не должна снизиться больше, чем на 10 % при размещении на вычислительных узлах ПО, которое производит мониторинг.

Для определения потерь производительности системы необходимо выполнить моделирование функционирования РКС на некотором наборе задач с учетом применения системы мониторинга и без ее использования при остальных равных условиях. Изменяя долю производительности вычислительных узлов, выделенную на функционирование системы мониторинга, рассмотрим, как будет изменяться производительность РКС в целом. Важно, чтобы при моде-

лировании использовался алгоритм планирования задач, который повторно может размещать задания на ресурсы системы по одной и той же схеме.

Существует несколько сред моделирования распределенных систем, обзор и анализ которых дан в работах [29, 30]. Для проведения эксперимента выберем среду GridSim [31]. Цель эксперимента — исследовать изменение параметров функционирования РКС при размещении на ее ВУ системы мониторинга действий узлов. Для этого промоделируем выполнение задач изначально без дополнительной вычислительной нагрузки на ВУ РКС, а затем для каждого узла введем нагрузку. Моделирование выполним на наборах задач, число которых составляет 250 (набор 1), 500 (набор 2), 750 (набор 3), 1000 (набор 4). В каждом наборе присутствуют задачи различной вычислительной сложности. Характеристики наборов задач представлены в табл. 1, где MI — million instructions (миллион команд). Задачи поступают в систему через разные временные промежутки от различных пользователей.

Модель РКС, созданная в среде GridSim, состоит из 10 однопроцессорных вычислительных узлов, характеристики вычислительных элементов задавались в соответствии с данными, приведенными в [31]. Порядок поступления задач во всех экспериментах не изменялся; обслуживание задач системой проводилось в соответствии с политикой FCFS (First Come First Served). В случае отсутствия для размещения задач свободного ресурса, они заносились в очередь в порядке поступления. Освободившимуся ресурсу передавалась на выполнение задача, находящаяся в начале очереди на текущий момент времени. Для каждого набора задач было проведено по три эксперимента (табл. 2).

Таблица 1

Набор задач	Характеристика наборов				
	Число задач	Интенсивность поступления, ед/с	Минимальная длина задачи, MI	Максимальная длина задачи, MI	Средняя длина задачи, MI
1	250	0,02	28950	3469052	1820483
2	500	0,02	12574	3499635	1755994
3	750	0,02	5511	3479489	1737141
4	1000	0,02	844	3498998	1729340

Таблица 2

Набор задач	Вид эксперимента	Среднее время ожидания в очереди, ч	Увеличение среднего времени ожидания в очереди, %	Среднее время пребывания в системе, ч	Увеличение среднего времени пребывания в системе, %
1	Первый эксперимент	13,53	—	14,79	—
	Второй эксперимент	14,75	9,01	16,11	8,92
	Третий эксперимент	22,59	66,96	24,61	66,4
2	Первый эксперимент	25,64	—	26,86	—
	Второй эксперимент	27,97	9,08	29,28	9,0
	Третий эксперимент	42,87	67,2	44,81	66,83
3	Первый эксперимент	39,44	—	40,64	—
	Второй эксперимент	43,02	9,08	44,32	9,06
	Третий эксперимент	65,83	66,91	67,76	66,73
4	Первый эксперимент	53,45	—	54,64	—
	Второй эксперимент	58,28	9,04	59,58	9,04
	Третий эксперимент	89,23	66,57	91,15	66,82

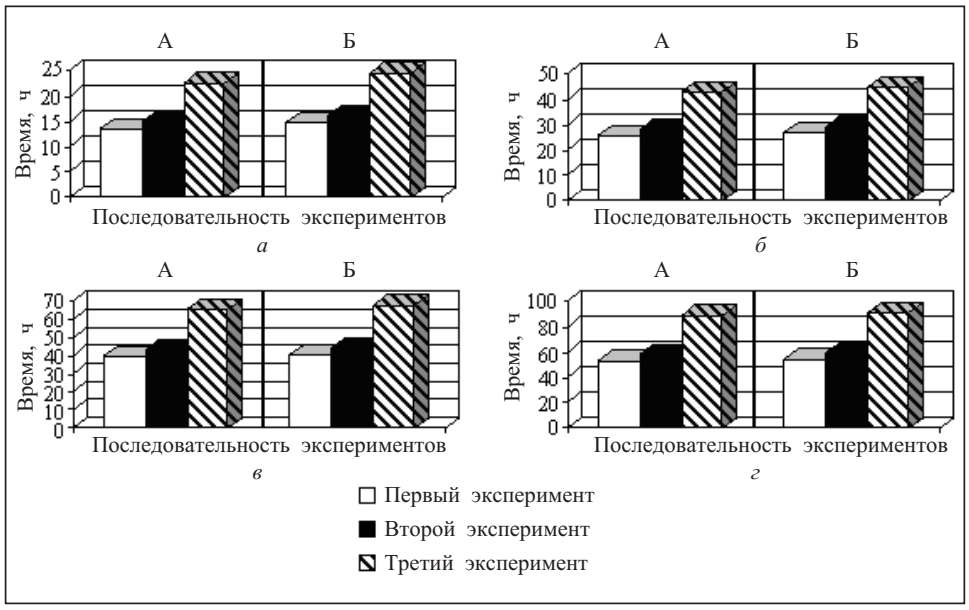

Рис. 3. Гистограмма влияния системы мониторинга действий вычислительного узла на функционирование РКС при среднем времени ожидания, ч (А) и среднем времени пребывания в системе, ч (Б): *а* — для набора 1 (250 задач), *б* — для набора 2 (500 задач), *в* — для набора 3 (750 задач), *г* — для набора 4 (1000 задач)

Первый эксперимент: без мониторинга. При данном эксперименте ВУ работали без дополнительной нагрузки. Для каждой задачи фиксировалось время поступления в систему, время начала обработки на вычислительном элементе и время завершения обработки. По полученным результатам для каждой задачи вычислялось время ожидания в очереди и время пребывания в системе, а также усредненные значения времени ожидания в очереди и времени пребывания в системе по всем задачам.

Второй эксперимент: система мониторинга задействует 5 % вычислительной мощности процессора вычислительного элемента. Методика проведения и сбора результатов аналогична предыдущему эксперименту. Среднее время ожидания задачи в очереди и среднее время пребывания задачи в системе увеличилось по сравнению с предыдущим экспериментом.

Третий эксперимент: система мониторинга задействует 30% вычислительной мощности процессора ВУ. Методика проведения и сбора результатов аналогична предыдущим двум экспериментам.

Из табл. 2 видно, на сколько процентов увеличилось среднее время ожидания задачи и среднее время пребывания задачи в системе для второго и третьего экспериментов относительно эксперимента, когда мониторинг не применялся.

На рис. 3 представлена гистограмма, которая демонстрирует влияние системы мониторинга действий вычислительного узла на функционирование РКС.

Таким образом, по результатам эксперимента можно сказать, что дополнительное ПО, применяемое для мониторинга состояния ВУ, снижает производительность РКС. Например, использование 30 % вычислительной мощности узла РКС для системы мониторинга привело к увеличению среднего времени ожидания задачи в очереди и среднего времени пребывания задачи в системе на уровне 66 %. В случае выделения 5% вычислительной мощности узла РКС для системы мониторинга наблюдалось увеличение среднего времени ожидания задачи в очереди и среднего времени пребывания задачи в системе на уровне 9 % по сравнению с тем, когда мониторинг не применялся. Определив допустимые затраты на обеспечение защищенной обработки данных на уровне 10 %, можно констатировать факт возможности применения системы мониторинга состояния ВУ с этой целью.

Однако выбор и разработка ПО, осуществляющего мониторинг, должны проводиться с учетом функциональных особенностей и предназначения распределенной системы.

Также следует отметить, что среднее время ожидания задачи в очереди и среднее время пребывания задачи в системе существенно зависят как от характеристик набора ВУ РКС, так и от параметров входного потока задач. В данных экспериментах они взяты в целях демонстрации влияния системы мониторинга состояния ВУ на работу РКС и не являются определяющими для оценки функциональности системы. В процессе эксперимента также не учтены и такие факторы, как выбор планировщиком ресурсов системы с учетом уровня доверия к узлам, нагрузка на сеть при обмене данными между ЛАД вычислительного узла и метапланировщиком (или информационным сервисом Grid-системы). Эти и другие вопросы требуют дальнейшей разработки и исследования.

ЗАКЛЮЧЕНИЕ

В настоящей статье предложена концепция управления ресурсами распределенной компьютерной системы, которая учитывает требования пользователя к безопасной обработке данных. Концепция базируется на возможности выбора ресурсов системы для выполнения задания, исходя из данных мониторинга функционирования системы. Предлагается использовать специальное программное обеспечение — локальный агент данных (ЛАД) для мониторинга состояния вычислительных узлов и каналов передачи данных системы. Мониторинг предлагается выполнять по трем параметрам: производительность, скорость передачи данных по каналам связи и уровень доверия к ВУ. В дальнейшем возможно расширение или изменение набора параметров в зависимости от цели использования РКС. Как было продемонстрировано экспериментально в рамках данного исследования, система мониторинга состояния ВУ, потребляя ресурсы РКС, приводит к снижению ее производительности. Поэтому данный факт должен учитываться как при реализации предложенной концепции в РКС, так и при разработке специализированного ПО, представляющего собой ЛАД.

Дальнейшее развитие работы предполагает разработку распределенного планировщика выполнения заданий в РКС на основе предложенной концепции, а также исследование показателей его функционирования.

СПИСОК ЛИТЕРАТУРЫ

1. Mariela J., Curiel H. Wireless Grids: Recent advances in resource and job management. Handbook of research on next generation mobile communication systems. Hershey, PA: Information Science Reference, an imprint of IGI Global, 2016. Ch. 12. P. 293–320.
2. Furthmüller J., Waldhorst O.P. Survey on Grid computing on mobile consumer devices. Handbook of research on P2P and Grid systems for service-oriented computing: Models, methodologies and applications (2 Vol.). N. Antonopoulos, G. Exarchakos, M. Li, A. Liotta (Eds). Hershey, PA: Information Science Reference, an imprint of IGI Global, 2010. Vol. 1, Ch. 13. P. 313–337.
3. Hussain H., Saif Ur Rehman Malik, Hameed A., Khan S.U., Bickler G., Min-Allah N., Qureshi M.B., Zhang L., Yongji W., Ghani N., Kolodziej J., Zomaya A.Y., Xu Ch.-Zh., Balaji P., Vishnu A., Pinel F., Pecero J.E., Kliazovich D., Bouvry P., Li H., Wang L., Chen D., Rayes A. A survey on resource allocation in high performance distributed computing systems. *Parallel Computing*. 2013. Vol. 39, Iss. 11. P. 709–736.
4. Sadashiv N., Kumar Dilip S.M. Cluster, Grid and Cloud computing: A detailed comparison. The 6th International Conference on Computer Science & Education (ICCSE 2011), August 3–5, 2011. P. 477–482.
5. Brandic I., Dustdar Schahram. Grid vs Cloud — a technology comparison. Information technology. *Methoden und Innovative Anwendungen der Informatik und Informationstechnik*. 2011. Vol. 53, Iss. 4. P. 173–179.
6. Qureshi M.B., Dehnavi M.M., Min-Allah N., Qureshi M.Sh., Hussain H., Rentifis I., Tziritas N., Loukopoulos Th., Khan S.U., Xu Cheng-Zhong, Zomaya A.Y. Survey on Grid resource allocation mechanisms. *Journal of Grid Computing*. 2014. Vol. 12, Iss. 2. P. 399–441.

7. Yang Y.L., Peng X.G., Cao J.F. Trust-based scheduling strategy for cloud workflow applications. *Informatica*. 2015. Vol. 26, N 1. P. 159–180.
8. Liu H., Abraham A., Snášel V., McLoone S. Swarm scheduling approaches for work-flow applications with security constraints in distributed data-intensive computing environments. *Information Sciences*. 2012. Vol. 192. P. 228–243.
9. Kaebeh Yaeghoobi S.B., Soni M.K., Tyagi S.S. Dynamic and real-time sleep schedule protocols for energy efficiency in WSNs. *International Journal of Computer Network and Information Security (IJCNIS)*. 2016. Vol. 8, N 1. P. 9–17. DOI: 10.5815/ijcnis.2016.01.02.
10. Tang X., Li K., Li R., Veeravalli Bh. Reliability-aware scheduling strategy for heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing*. 2010. Vol. 70, Iss. 9. P. 941–952.
11. Wang X., Yeo C.S. , Buyya R., Su J. Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm. *Future Generation Computer Systems.* 2011. Vol. 27, Iss. 8. P. 1124–1134.
12. Huang Y., Bessis N., Norrington P., Kuonen P., Hirsbrunner B. Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm. *Future Generation Computer Systems.* 2013. Vol. 29, Iss. 1. P. 402–415.
13. Mohan R., Gopalan N.P. Task assignment for heterogeneous computing problems using improved iterated greedy algorithm. *International Journal of Computer Network and Information Security (IJCNIS)*. 2014. Vol. 6, N 7. P. 50–55. DOI: 10.5815/ijcnis.2014.07.07.
14. Chauhan S.S., Joshi R.C. A heuristic for QoS based independent task scheduling in Grid environment. 5th International Conference on Industrial and Information Systems, ICIIS 2010, Jul. 29–Aug. 01, 2010. Delhi, India. P. 102–106.
15. Tan Fong Ang, Teck Chaw Ling, Keat Keong Phang. Adaptive QoS scheduling in a service-oriented grid environment. *Turk. J. Elec. Eng. & Comp. Sci*. 2012. Vol. 20. N 3. P. 413-424.
16. Conejero J., Tomás L., Caminero B., Carrión C. QoS provisioning by meta-scheduling via advance within SLA-based Grid environments. *Computing and Informatics*. 2012. Vol. 31, N 1. P. 73–88.
17. Lin W., Liang Ch., Wang J.Z., Buyya R. Bandwidth-aware divisible task scheduling for cloud computing. *Software-Practice and Experience*. 2014. Vol. 44, Iss. 2. P. 163–174.
18. Caminero A., Rana O., Caminero B., Carrión C. Network-aware heuristics for inter-domain meta-scheduling in Grids. *Journal of Computer and System Sciences*. 2011. Vol. 77, Iss. 2. P. 262–281.
19. Jin J., Luo J., Song A., Dong F., Xiong R. BAR: An efficient data locality driven task scheduling algorithm for Cloud computing. *Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. 2011. P. 295–304.
20. Yang C.-T., Leu F.-Y., Chen S.-Y. Network Bandwidth-aware job scheduling with dynamic information model for Grid resource brokers. *The Journal of Supercomputing*. 2010. Vol. 52, Iss. 3. P. 199–223.
21. McClatchey R., Anjum A., Stockinger H., Ali A., Willers I., Thomas M. Scheduling in data intensive and network aware (DIANA) Grid environments architecture [online]. URL: https:// arxiv.org/ ftp/ arxiv/papers/0707/0707.0862.pdf.
22. Haquea A., Alhashmia S.M., Parthiban R. A survey of economic models in grid computing. *Future Generation Computer Systems*. 2011. Vol. 27, Iss. 8. P. 1056–1069.
23. Kołodziej J., Khan S.U., Wang L., Zomaya A.Y. Energy efficient genetic-based schedulers in computational grids. *Concurrency Computation: Practice and Experience.* 2015. Vol. 27, Iss. 4. P. 809–829.
24. Sheikh H.F., Tan H., Ahmad I., Ranka S. Energy- and performance-aware scheduling of tasks on parallel and distributed systems. *ACM Journal on Emerging Technologies in Computing Systems*. 2012. Vol. 8, Iss. 4. P. 1–37.
25. Aydt R., Gunter D., Smith W., Swany M., Taylor V., Tierney B., Wolski R. A Grid monitoring architecture. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.6602&rep=rep1&type=pdf.
26. Abebe Tesfahun, Bhaskari D. Lalitha. Effective hybrid intrusion detection system: A layered approach. *International Journal of Computer Network and Information Security (IJCNIS)*. 2015. Vol. 7, N 3. P. 35–41. DOI: 10.5815/ijcnis.2015.03.05.
27. Bhuyan M.H., Bhattacharyya D.K., Kalita J.K. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*. 2014. Vol. 16, N 1. P. 303–336.

28. Mukhin V.Ye., Bidkov A.Ye., Thinh Vu Duc. The forming of trust level to the nodes in the distributed computer systems. *Proc. of XI International Conference "Modern Problems of Radio Engineering, Telecommunications and Computer Science TCSET'2012*". Lvov – Slavsko, 21–24 February 2012. P. 362.

29. Минухин С.В., Коровин А.В. Моделирование планирования ресурсов GRID средствами пакета GRIDSIM. *Системи обробки інформації*. 2011. № 3 (93). С. 62–68.

30. Скитер И.С. Прелая О.А., Гуза Т.А. Обзор инструментов разработки моделей GRID-среды. *Вільне програмне забезпечення в освіті, науці та бізнесі*. Тези доповідей V Міжнародної науково-практичної конференції. Чернігів: Черн. нац. технол. ун-т, 2014. С. 28–29.

31. Buyya R., Murshed M. GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing. *The Journal of Concurrency and Computation: Practice and Experience (CCPE)*. Wiley Press, 2002. 37 p. URL: http://arxiv.org/ abs/cs/0203019.

Надійшла до редакції 29.07.2016

Чженбін Ху, В.Є. Мухін, Я.І. Корнага, О.Ю. Герасименко
УПРАВЛІННЯ РЕСУРСАМИ РОЗПОДІЛЕНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ З УРАХУВАННЯМ РІВНЯ ДОВІРИ ДО ОБЧИСЛЮВАЛЬНИХ КОМПОНЕНТІВ

Анотація. Розглянуто гарантування безпечного оброблення даних у розподілених комп'ютерних системах (РКС), що є критично важливим для виконання певного класу задач. Запропоновано підхід до управління ресурсами РКС, який відповідно до вимог користувача дозволяє врахувати як витрати часу на виконання завдання, так і рівень захищеності ресурсів, які залучаються для його виконання.

Ключові слова: розподілені обчислення, управління ресурсами, планування завдань, безпечне оброблення даних, моніторинг стану обчислювального вузла, локальний агент даних.

Zhengbing Hu, V.Ye. Mukhin, Ya.I. Kornaga, O.Yu. Herasymenko
RESOURCE MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM TAKING INTO ACCOUNT THE TRUST LEVEL TO THE COMPUTATIONAL NODES

Abstract. The safe data processing in distributed computer system (DCS) is critical for a certain class of computational tasks. This paper describes an approach to resource management in DCS, which, according to user's requirements, allows taking into account the task execution time as well as the security level of system's resources.

Keywords: distributed computing, resource management, scheduling, secure data processing, computing node state monitoring, local data agent.

Чженбин Ху,
кандидат техн. наук, доцент, Педагогический университет Центрального Китая, Ухань, e-mail: hzb@mail.ccnu.edu.cn.

Мухин Вадим Евгеньевич,
доктор техн. наук, профессор Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского», e-mail: v.mukhin@kpi.ua.

Корнага Ярослав Игоревич,
кандидат техн. наук, доцент Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского», e-mail: ya.kornaga@kpi.ua.

Герасименко Оксана Юрьевна,
ассистент, Киевский национальный университет имени Тараса Шевченко, e-mail: oksgerasymenko@gmail.com.