

МЕТОДЫ И СРЕДСТВА МЕТАМОНИТОРИНГА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕД¹

И.А. Сидоров, А.П. Новопашин, Г.А. Опарин, В.В. Скоров

Работа посвящена решению проблемы мониторинга распределенной вычислительной среды, состоящей из множества слабосвязанных разнородных вычислительных ресурсов. Главной отличительной особенностью и аспектом новизны разработанных методов и средств является использование универсальных программных агентов, способных осуществлять сбор данных от локальных систем мониторинга различных компонентов среды, унифицировать и анализировать эти данные, а также при необходимости вырабатывать и исполнять в автоматическом режиме управляющие воздействия, направленные на изменение режимов работы оборудования, в том числе его отключение при возникновении аварийных ситуаций. В составе каждого агента системы метамониторинга реализованы: подсистема сбора данных, подсистема взаимодействия с СУБД, экспертная подсистема и управляющая подсистема. Подход основан на применении веб-технологий, мультиагентных технологий, методов создания экспертных систем, методов децентрализованной обработки и распределенного хранения данных.

Ключевые слова: распределенные вычислительные среды, мониторинг, мультиагентные технологии, экспертные системы

Введение

Рассматриваемые в статье распределенные вычислительные среды (РВС) предназначены для проведения ресурсоемких вычислительных экспериментов и включают высокопроизводительные системы различной архитектуры и конфигурации: выделенные серверы, гибридные вычислительные системы с модулями ускорения вычислений на базе графических ускорителей, сопроцессоров и ПЛИС, вычислительные кластеры (суперкомпьютеры), функционирующие под управлением различных систем управления прохождением заданиями (PBS/Torque, Cleo, СУППЗ).

При организации таких РВС особое внимание уделяется средствам управления и мониторинга. Основными функциями средств управления РВС является распределение нагрузки по вычислительным узлам, реализация непараллельных и параллельных команд, запуск и остановка вычислительных узлов и ряд других. Назначение средств мониторинга состоит в обеспечении оператора РВС структурированной и унифицированной информацией о загрузке ресурсов вычислительных узлов (процессоров, оперативной памяти, системы ввода-вывода, коммуникационных интерфейсов), физическом состоянии вычислительного оборудования (температуре процессоров и материнских плат), работе устройств инженерной инфраструктуры (кондиционеров, чиллеров, систем бесперебойного питания) и текущем состоянии окружающей среды.

С ростом числа вычислительных узлов в составе РВС увеличивается объем данных, которые приходится анализировать оператору. Как следствие, возрастает вероятность возникновения ошибок, обусловленных человеческим фактором. В связи с этим, в больших РВС огромное значение имеет форма представления данных мониторинга, их

¹ Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии – 2014»

наглядность, информативность и актуальность – свойства, позволяющие обслуживающему персоналу РВС оперативно выявлять причины неисправностей и сбоев, своевременно реагировать на изменения в работе системы.

Известные на сегодняшний день средства мониторинга имеют ряд недостатков и ограничений, не позволяющих эффективно применять их в сложных вычислительных системах. В частности, в популярной системе Ganglia отсутствует возможность реализации механизмов обработки текущих значений и уведомлений о событиях. Две другие часто используемые системы – Nagios и Zabbix лишены этого недостатка, однако их первоначальная ориентация на мониторинг сетевого оборудования делает их малоприспособленными для РВС.

Следует выделить отдельно систему мониторинга LAPTA [1], предназначенную для многоаспектного анализа динамических характеристик параллельных программ, выполняемых на суперкомпьютерах. Система предполагает тесную интеграцию с известными средствами управления заданиями. Сбор информации о ресурсах осуществляется с привязкой к конкретной задаче пользователя, что представляется, с нашей точки зрения, наиболее верным направлением развития систем мониторинга вычислительных кластеров и РВС. В то же время, зависимость компонентов системы мониторинга LAPTA от специализированных интерпретируемых языков программирования и систем хранения данных значительно усложняет ее установку и настройку. Кроме того, реализация агента системы мониторинга на основе интерпретируемого языка программирования требует значительных системных ресурсов, что может приводить к снижению эффективности работы пользовательских приложений на узлах РВС.

В данной работе предлагается подход к созданию инструментального комплекса метамониторинга РВС, отличающейся от известных уникальным сочетанием свойств, к числу которых относятся: автоматический контроль программно-аппаратных ресурсов с использованием мультиагентных технологий, децентрализованная схема хранения данных метамониторинга, применение экспертных систем для принятия решений.

Исходные требования к разрабатываемому инструментальному комплексу метамониторинга РВС [2] предполагают, что он должен:

- базироваться на принципах организации мультиагентных систем;
- обеспечивать высокую степень автономности компонентов;
- иметь многоуровневую иерархическую структуру и средства, гарантирующие быстрое и надежное взаимодействие различных уровней, прямую и обратную связь между ними;
- обеспечивать возможность интеграции с популярными системами мониторинга высокопроизводительных ресурсов (Ganglia, Nagios и др.);
- предоставлять широкий набор функций сбора данных о состоянии программно-аппаратных компонентов вычислительных узлов, предоставлять средства разработки модулей сбора данных на различных языках системного программирования (C, Perl, Bash и др.);
- предоставлять средства сбора и анализа данных о состоянии оборудования вспомогательной (инженерной) инфраструктуры вычислительных кластеров;
- включать средства унификации данных, получаемых из различных источников;
- использовать технологию децентрализованного хранения данных, обеспечивающую снижение нагрузки на центральный узел системы метамониторинга;

- предоставлять средства автоматизированного экспертного анализа данных и генерации управляющих воздействий;
- включать средства визуализации, наглядно отображающие состояние ресурсов, задействованных в ходе решения задач;
- предоставлять прикладные программные интерфейсы на основе открытых стандартов для интеграции с другими программными комплексами.

1. Модель мультиагентной системы метамониторинга

Модель мультиагентной системы метамониторинга программно-аппаратных ресурсов (узлов) РВС можно представить в виде следующей структуры:

$$D = \langle N, M, T, A, R, S \rangle,$$

где:

- N – множество узлов РВС;
- M – множество измеряемых метрик (характеристик) узлов РВС;
- T – множество технических средств коммуникации узлов РВС;
- A – множество агентов системы метамониторинга;
- R – множество правил вывода экспертной подсистемы;
- S – множество управляющих воздействий.

Связи между элементами множеств заданы отношениями:

- $NM \subseteq N \times M$ – отношение, определяющее множество метрик, измеряемых на узлах РВС;
- $NT \subseteq N \times T$ – отношение, определяющее топологию сети узлов РВС;
- $NA \subseteq N \times A$ – отношение, определяющее иерархическую структуру мультиагентной среды;
- $RM \subseteq R \times M$ – отношение, определяющее множество правил вывода, используемых экспертной подсистемой;
- $SM \subseteq S \times M$ – отношение, определяющее множество управляющих воздействий, генерируемых экспертной подсистемой по результатам анализа значений метрик.

Задача метамониторинга сформулирована следующим образом. Пусть $n_i \subset N$ – отдельный узел РВС. Для i -го узла измеряются m_i метрик ($m_i \subset M$). $B_i(t) = \{b_{ik}(t)\}$ – множество значений метрик для i -го узла в момент времени t , где $k = 1, 2, \dots, m_i$.

Устойчивым состоянием метрик узла на отрезке времени $[t_0, t_1]$ считается такое, для которого выполняется неравенство:

$$|b_{ik}(t) - b_{ik}(t_0)| \leq c_{ik}, \forall t \in [t_0, t_1], \forall k \in [1, m_i],$$

где c_{ik} – положительная константа, характеризующая допустимый разброс значений для каждой метрики.

Значимым событием является переход узла из устойчивого состояния $B_i(t_0)$ в состояние $B_i(t_1)$, при котором происходит существенное изменение значения какой-либо метрики из множества m_i .

При возникновении значимого события на отрезке времени $[t_0, t_1]$ формируется перечень фактов для экспертной подсистемы, и на основе правил вывода R_i , определенных для узла i , машиной вывода генерируются управляющие воздействия S_i .

Под управляющим воздействием S_i понимается инструкция, предназначенная для реагирования на значимое событие. Среди таких управляющих воздействий могут быть:

- уведомление оператора системы метамониторинга РВС;

- уведомление пользователя, задача которого выполняется на узле n_i ;
- снятие задач пользователя, выполняющихся на узле n_i ;
- выключение узла n_i (к примеру, при длительном простое или выявлении критических значений заданных метрик);
- включение узла n_i (к примеру, при появлении задач в очереди).

Сгенерированное экспертной подсистемой управляющее воздействие передается управляющей подсистеме, где оно транслируется в системные команды и затем исполняется.

Далее состояние узла (B_i) передается агентам верхних уровней для анализа и генерации управляющих воздействий в соответствии с правилами, определенными на соответствующих уровнях.

2. Архитектура мультиагентной системы метамониторинга

Архитектура мультиагентной системы метамониторинга РВС включает следующие основные компоненты (показано на рис. 1):

- средства доступа пользователей, позволяющие взаимодействовать с системой метамониторинга как в пакетном, так и в интерактивном режимах;
- подсистемы уровня доступа, осуществляющие контроль прав доступа к запрашиваемым данным и реализующие серверную часть графического интерфейса пользователя;
- агент верхнего уровня, функционирующий в центральном узле РВС и выполняющий основную задачу по управлению системой метамониторинга;
- агенты промежуточного уровня, функционирующие на промежуточных узлах и решающие задачу снижения нагрузки на агенты верхних уровней;
- агенты нижнего уровня, функционирующие на исполнительных узлах РВС и осуществляющие сбор и первичную обработку данных о состоянии узла;
- подсистема децентрализованного хранения данных, предоставляющая функции для работы с данными для агентов разных уровней.

Иерархическая структура представляемой системы метамониторинга достигается путем организации многоуровневой мультиагентной среды, возглавляемой агентом верхнего уровня, которому подчинены агенты промежуточного и нижнего уровня. Агенты промежуточного уровня могут осуществлять управление как группой агентов нижнего уровня, так и группой агентов дополнительных промежуточных уровней, которых в системе может быть несколько. Такая иерархия позволяет обеспечить высокую масштабируемость системы метамониторинга для РВС, состоящей из большого числа узлов.

Каждый агент мультиагентной системы метамониторинга реализован в виде программы, функционирующей в фоновом режиме, и обладает свойствами, характерными для автономных программных агентов [3], а именно, реактивностью, автономностью, целенаправленностью и коммуникативностью.

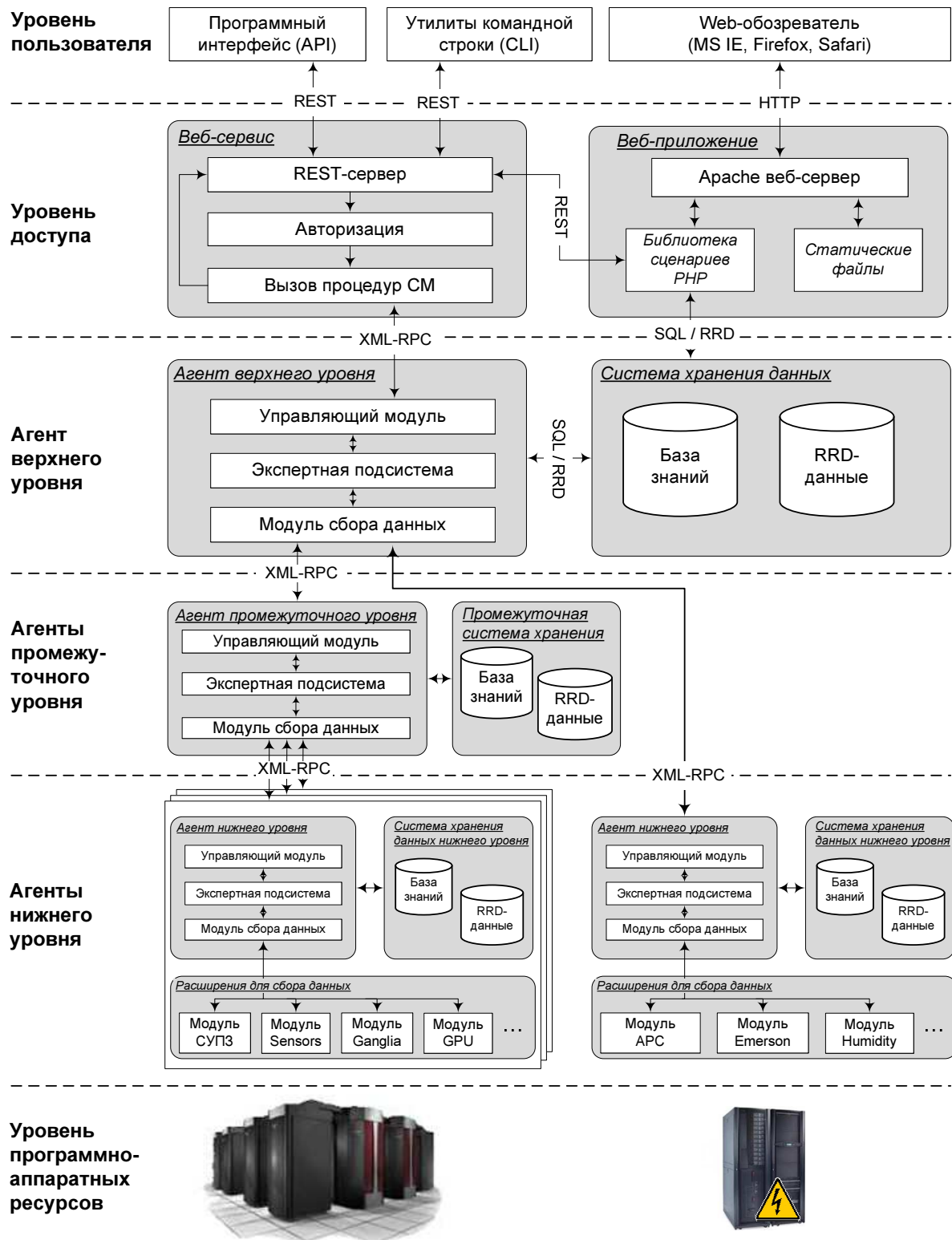


Рис. 1. Архитектура системы метамониторинга PBC

В составе агента системы метамониторинга реализованы следующие подсистемы:

- подсистема сбора данных, осуществляющая считывание показаний программных и аппаратных сенсоров различных устройств, прием данных от локальных систем мониторинга и агентов разных уровней, унификацию полученных данных и их трансляцию в промежуточный формат подсистемы взаимодействия с СУБД;

- подсистема взаимодействия с СУБД, выполняющая функции агрегации и контроля целостности данных;
- экспертная подсистема, выполняющая функции анализа данных, полученных за определенный интервал времени, и генерации управляющих воздействий на основе проведенного анализа;
- управляющая подсистема, реализующая функции исполнения управляющих воздействий и взаимодействия с агентами верхних уровней.

Агент нижнего уровня устанавливается непосредственно на узле РВС и выполняет функции сбора данных о состоянии узла, анализа этих данных, генерации и исполнения управляющих воздействий, а также функции взаимодействия с другими агентами.

Возможность анализа данных и принятия необходимых решений на стороне узла является ключевой особенностью реализуемого подхода. Как правило, клиенты систем мониторинга выполняют лишь функции сбора и периодической отправки данных в центральный узел, где производится обработка и анализ этих данных. При такой реализации существенно снижается количество операций по анализу данных на стороне узла, однако создается дополнительная нагрузка на стек сетевых протоколов, обработка которого (формирование, разбор и контроль сетевых пакетов) также требует процессорного времени. В нашем подходе анализ данных на стороне узла позволяет производить пересылку данных на центральный узел только в случае необходимости (либо по специальному запросу агентов верхних уровней). Установлено, что первичный анализ данных на стороне узла требует значительно меньше ресурсов, чем затраты на формирование сетевых пакетов и проверки их целостности. Сравнительный анализ работы агентов системы мониторинга Ganglia (gmond) и агентов разрабатываемой системы показал, что при одинаковой периодичности опроса датчиков (каждые 30 сек.), последний потребляет на 37% процессорного времени меньше. Кроме того, базовый набор сенсоров разрабатываемого агента значительно шире набора сенсоров агента системы мониторинга Ganglia. Таким образом, помимо снижения нагрузки на сеть и центральный узел системы метамониторинга удастся снизить негативное влияние агента на производительность решения задач на узлах РВС.

Сбор информации о состоянии узла и загрузке отдельных его компонентов (процессора, памяти, сетевых интерфейсов и др.) реализован с использованием функций библиотеки SIGAR [4]. Реализация сторонних датчиков выполнена на специализированном языке, являющемся подмножеством языка ECMA Script [5] и поддерживающем вызовы внешних команд, обработку выходного потока, регулярные выражения и ряд других механизмов, позволяющих получать данные о состоянии нестандартных программно-аппаратных средств.

Агент промежуточного уровня устанавливается на промежуточных узлах РВС и выполняет функции сбора данных от группы агентов нижних уровней, агрегации и анализа этих данных, генерации и исполнения необходимых управляющих воздействий (в соответствии с правилами вывода, определенными для агента данного уровня), а также при необходимости осуществляет отправку данных агентам верхних уровней. Основное назначение агента промежуточного уровня – снижение нагрузки на агенты верхних уровней.

Агент верхнего уровня устанавливается в центральном узле РВС и выполняет

функции ядра системы, владеет полной информацией о состоянии узлов РВС и предоставляет средства доступа к данным системы метамониторинга. Агент верхнего уровня может взаимодействовать с агентами промежуточного уровня, с агентами нижнего уровня, а также напрямую получать данные от локальных систем мониторинга и отдельных сенсоров. Собираемая информация накапливается в базе данных, унифицируется, агрегируется и предоставляется для доступа экспертной подсистеме и подсистемам уровня доступа. Экспертная подсистема агента верхнего уровня анализирует общее состояние РВС, генерирует управляющие воздействия, которые далее исполняются управляющей подсистемой. В базе знаний экспертной системы содержится информация о том, какие управляющие воздействия должны приниматься в автоматическом режиме, а какие при участии оператора РВС.

Экспертная подсистема. Для реализации функций экспертной подсистемы использована популярная среда CLIPS [6], основанная на продукционной модели знаний и предоставляющая программный интерфейс для работы с основными элементами экспертной подсистемы: базой знаний, содержащей списки фактов, списки объектов и правила вывода; механизмом вывода, реализующим логический вывод на основе сформированной базы знаний.

Для анализа значений метрик выбрана классификация по степени критичности, в соответствии с которой для каждой метрики определено четыре состояния: ошибка, критическое, предупреждение, удовлетворительное.

Значения метрик обрабатываются набором правил, которые оценивают эти значения в соответствии с введенными граничными параметрами и присваивают каждой метрике флаг состояния. Если в списке фактов появляется хотя бы одно состояние “ошибка” или “критическое”, то для избранных метрик проверяется история значений за определенный интервал, чтобы исключить явления, несущие случайный характер и неспособные в какой-либо мере повлиять на работу системы. Если информация о состоянии ошибки или критическом состоянии подтверждается, то генерируются управляющие воздействия, среди которых могут быть следующие: информирование агента верхнего уровня; создание и отправка сообщений оператору РВС; действия, направленные на автоматическое решение возникшей проблемы.

Подсистема накопления и обработки данных основана на принципах функционирования циклических баз данных (Round Robin Database) – баз данных, объем которых не меняется со временем. Фиксированный размер таких баз достигается благодаря заранее определенному количеству записей, используемых циклически для сохранения данных.

Несмотря на неоспоримые преимущества циклических баз данных, наиболее известные реализации (MRTG, RRDtools и др.) имеют недостатки, связанные, прежде всего, с низкой производительностью операций чтения/записи данных.

Проведенные эксперименты по созданию циклической базы данных на основе легкой встраиваемой реляционной базы данных SQLite выявили еще более худшую производительность таких баз данных в сравнении с RRDtools и MRTG. В результате было принято решение о создании собственной реализации циклической базы данных, использующей для хранения структурированной информации специализированный текстовый формат, основанный на расширяемом метаязыке XML. Были реализованы механизмы чтения/записи данных, агрегации данных за определенный интервал времени, вытесне-

ния устаревших данных, выборки данных по определенным критериям, а также механизмы кеширования данных. Собственная реализация циклической базы данных показала высокую эффективность в сравнении с универсальными системами организации циклических баз данных (RRDtools, MRTG).

Для хранения правил вывода экспертной подсистемы, списка фактов, сгенерированных управляющих воздействий, шаблонов для трансляции управляющих воздействий в системные команды и других знаний используется легковесная встраиваемая реляционная база данных SQLite. Для выполнения перечисленных функций производительность SQLite является приемлемой, а возможность встраивания средств управления базой данных в состав программного агента позволяет сохранить малый размер программы и обеспечить минимальную зависимость от стороннего программного обеспечения, установленного на узле.

Графический пользовательский интерфейс является одним из наиболее важных компонентов любой системы, предполагающей взаимодействие с пользователем. Информативность и наглядность представления данных, актуальность отображаемой информации, возможность формирования комплексных отчетов по выбранным параметрам и ряд других возможностей делают систему полезной для различных категорий пользователей.

В качестве одного из существенных отличий предлагаемого подхода к построению интерфейса системы метамониторинга является реализация средств генерации графиков на клиентской стороне веб-приложения (в веб-браузере пользователя). Такой подход, во-первых, позволяет снизить нагрузку на центральный узел системы метамониторинга за счет переноса части нагрузки на компьютер пользователя. Во-вторых, при таком подходе появляется возможность отображать графики в режиме реального времени без периодической перезагрузки веб-страниц. В-третьих, в значительной мере снижается объем передаваемого трафика между клиентской и серверной сторонами веб-приложения.

В качестве языков разработки пользовательского веб-интерфейса были выбраны: язык программирования PHP для реализации серверной части веб-приложения и совокупность веб-технологий, языков программирования и библиотек (таких как HTML5, JavaScript, Ajax, ExtJS, jQuery) для реализации клиентской части веб-приложения. Пример отображения графиков в режиме реального времени с использованием графических средств библиотеки ExtJS приведен на рис. 2.

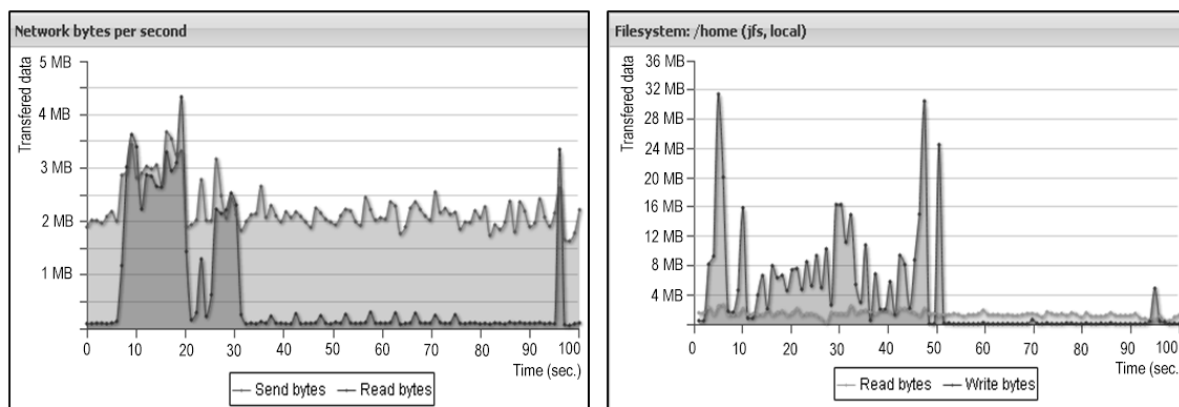


Рис. 2. Графики текущей загрузки сетевого интерфейса (слева) и файловой системы (справа)

На рис. 3 приведена схема размещения компонентов систем холодоснабжения и электроснабжения [7] вычислительного кластера «Академик В.М. Матросов» в машинном зале (слева) и на внешней площадке (справа) с отображением значений ключевых метрик. На рис. 4 представлен список значений метрик в табличной форме.

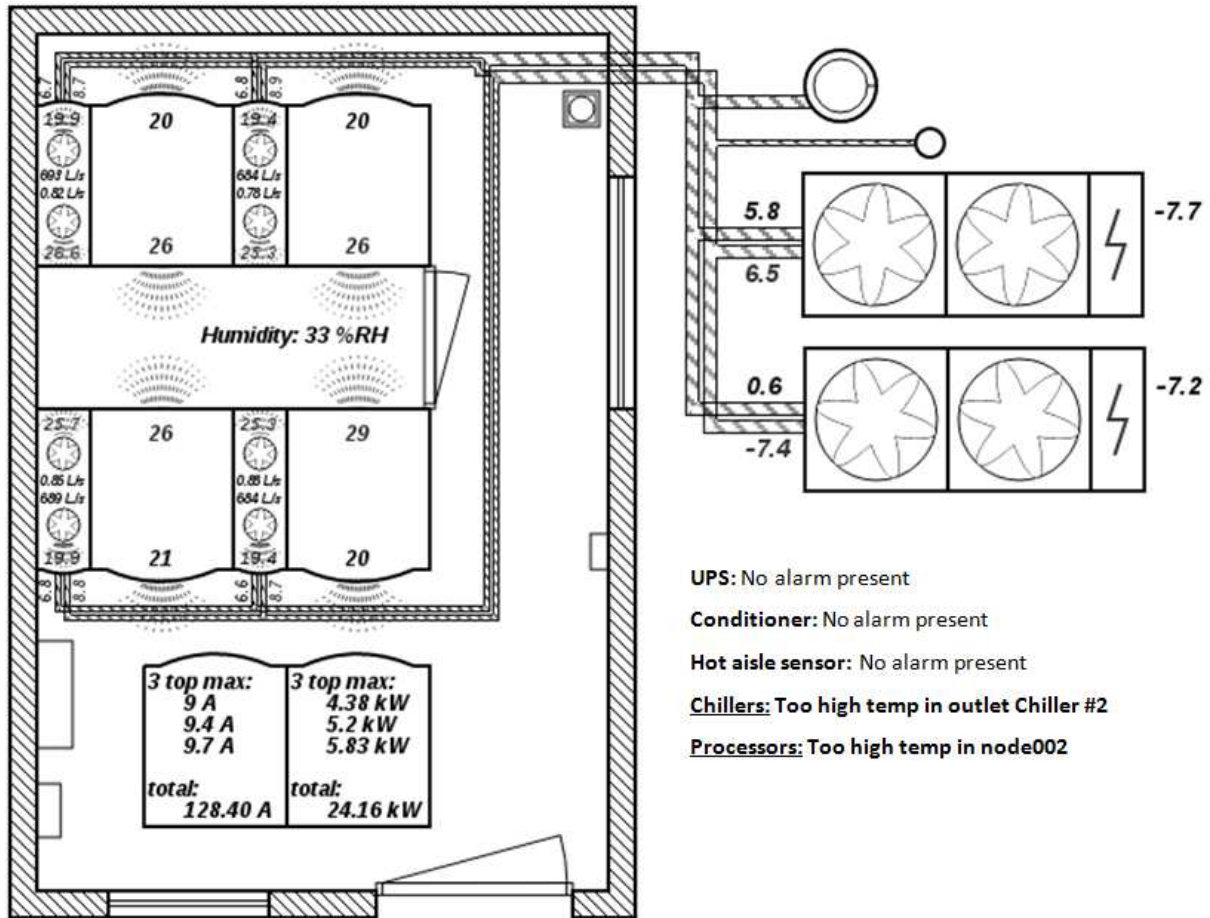


Рис. 3. Схема размещения компонентов систем холодоснабжения и электроснабжения с указанием значений ключевых метрик

Conditioners					UPS			
Статус работы	On	On	On	On	#	Rating	Current	Power
Выходная мощность охлаждения	12.0 kW	10.7 kW	13.3 kW	9.4 kW	1	16A	1.1 A	0.21 kW
Необход. мощность охлаждения	12.0 kW	10.7 kW	13.3 kW	9.4 kW		16A	0.0 A	0.00 kW
Температура на входе	21.3 °C	20.9 °C	21.3 °C	22.2 °C		16A	1.2 A	0.22 kW
Темп. возвращаемого воздуха	20.1 °C	19.9 °C	20.0 °C	20.1 °C	2	16A	1.1 A	0.19 kW
Темп. приточного воздуха	30.3 °C	30.9 °C	30.3 °C	29.8 °C		16A	0.0 A	0.00 kW
Поток воздуха	829 L/s	834 L/s	834 L/s	834 L/s		16A	1.1 A	0.21 kW
Скорость вентиляторов	53.4 %	53.9 %	53.9 %	53.9 %	3	32A	15.8 A	8.38 kW
Состояние на входе	Open	Open	Open	Open		32A	9.4 A	
Состояние на выходе	Closed	Closed	Closed	Closed		32A	13.7 A	
Активный источник питания	A	A	A	A	4	32A	16.4 A	10.47 kW
Перепад давления на фильтре	0 Pa	5 Pa	17 Pa	15 Pa		32A	18.9 A	
Положение клапана в жидкости	54 %	54 %	56 %	57 %		32A	11.9 A	
Поток жидкости	0.85 L/s	0.81 L/s	0.88 L/s	0.76 L/s	5	32A	15.8 A	10.65 kW
Температура жидкости на входе	6.2 °C	6.2 °C	5.9 °C	6.7 °C		32A	16.6 A	
Температура жидкости на выходе	9.9 °C	9.6 °C	9.8 °C	9.9 °C		32A	15.7 A	
Общие установ. значения темп.	22.2 °C	22.2 °C	22.2 °C	22.2 °C	6	32A	15.3 A	8.39 kW
Общие установленные значения	20.0 °C	20.0 °C	20.0 °C	20.0 °C		32A	9.8 A	
						32A	13.3 A	

Рис. 4. Список значений метрик в табличной форме

3. Вычислительный эксперимент

Разработанные методы и инструментальные средства метамониторинга РВС были успешно апробированы в суперкомпьютерном центре ИДСТУ СО РАН [8] при решении ресурсоемких задач в области биоинформатики, квантовой химии, логического поиска и других.

Экспериментальная РВС включала следующие программно-аппаратные ресурсы:

- 2 вычислительных узла с графическими ускорителями Nvidia Tesla C1060 с общим числом ядер 1920;
- 10 вычислительных узлов с процессорами AMD Opteron 6276 с общим числом ядер 320;
- 20 вычислительных узлов с процессорами Intel Xeon 5345 с общим числом ядер 160.

В процессе тестирования системы метамониторинга в РВС был выявлен перечень программно-аппаратных ресурсов, находившихся в критическом состоянии и в состоянии ошибки:

- (warning node node-4.blackford.icc.ru loadavg 43);
- (critical node node-13.blackford.icc.ru cpu-sys-p 77);
- (critical node node011.matrosov.icc.ru filesystem /store wtime 583816);
- (error node node-4.blackford.icc.ru available down).
- (warning node node-15.blackford.icc.ru memory-used-ten 81);
- (critical node node2.tesla.icc.ru memory-used-ten 92);
- (error node node-7.blackford.icc.ru filesystem /store du-free 0).

Анализ сведений о состоянии программно-аппаратных ресурсов РВС, собранных системой метамониторинга, позволил выявить неэффективную работу отдельных пользовательских приложений, произвести оптимизацию загрузки вычислительных ресурсов и повысить показатели надежности РВС.

Например, при обработке экспериментальных данных для аннотации генома байкальской диатомовой водоросли *Synedra acus* с использованием программного пакета MAKER [9] было выявлено преобладание операций чтения/записи в сетевую директорию по отношению к вычислительным операциям. Своевременная установка значений конфигурационных параметров пакета, указывающих на необходимость записи результатов вычислений в локальные директории узлов, позволила повысить эффективность работы пакета более чем на 30%.

Заключение

Новизна предложенного подхода к организации системы метамониторинга РВС заключается, в первую очередь, в создании универсальных программных агентов, способных вести сбор данных о состоянии узлов РВС, анализировать эти данные и принимать самостоятельные решения без участия оператора РВС. Мультиагентный подход позволил, во-первых, снизить общую нагрузку, создаваемую компонентами системы метамониторинга, во-вторых, повысить надежность РВС за счет децентрализованной генерации и исполнения управляющих воздействий. Универсальность архитектуры разработанного программного агента делает возможным его использование на различных уровнях иерархии системы метамониторинга, что, в свою очередь, обеспечивает высокую масштабируемость системы при ее использовании в РВС, состоящей из большого количества узлов.

Работа выполнена при частичной финансовой поддержке Совета по грантам Президента Российской Федерации для государственной поддержки ведущих научных школ (НШ-5007.2014.9).

Литература

1. Адинец, А.В. Мониторинг, анализ и визуализация потока заданий на кластерной системе / А.В. Адинец, П.А. Брызгалов, Вад.В. Воеводин // Высокопроизводительные параллельные вычисления на кластерных системах: Материалы XI Всерос. конф. – Нижний Новгород, 2011. – С. 10–14.
2. Сидоров, И.А. Один из подходов к реализации средств мониторинга высокопроизводительных вычислительных систем / И.А. Сидоров, В.В. Скоров // Информационные и математические технологии в науке и управлении: Тр. XVIII Байкальской Всерос. конф. – Ч. III. – Иркутск: Изд-во ИСЭМ СО РАН, 2013. – С. 177–184.
3. Wooldridge, M. Intelligent Agents: Theory and Practice / M. Wooldridge, N. Jennings // The Knowledge Engineering Review. – 1995. – Vol. 10, No 2. – P. 115–152.
4. System Information Gatherer and Reporter API. URL: <http://www.hyperic.com/products/sigar> (дата обращения: 08.04.2014).
5. Standard ECMA-262: ECMAScript Language Specification. URL: <http://www.ecma-international.org/publications/standards/Ecma-262.htm> (дата обращения: 08.04.2014)
6. Частиков, А.П. Разработка экспертных систем. Среда CLIPS / А. П. Частиков, Д. Л. Белов, Т. А. Гаврилова. – Санкт-Петербург: БХВ-Петербург, 2003. – 393 с.
7. Скоров В.В. Редактор графического представления объектов инженерной инфраструктуры вычислительного центра / В.В. Скоров, И.А. Сидоров, А.П. Новопашин // Ляпуновские чтения: Материалы конф. – Иркутск: Изд-во ИДСТУ СО РАН, 2013. – С. 53.
8. Иркутский суперкомпьютерный центр СО РАН. URL: <http://hpc.icc.ru> (дата обращения: 08.04.2014)
9. MAKER – genome annotation pipeline. URL: <http://gmod.org/wiki/MAKER> (дата обращения: 08.04.2014)

Сидоров Иван Александрович, к.т.н., н.с., Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления СО РАН (Иркутск, Российская Федерация), ivan.sidorov@icc.ru

Новопашин Алексей Петрович, к.т.н., зав. лаб., Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления СО РАН (Иркутск, Российская Федерация), arn@icc.ru

Опарин Геннадий Анатольевич, д.т.н., проф., зам. директора по научной работе, Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления СО РАН (Иркутск, Российская Федерация), oparin@icc.ru

Скоров Владимир Владимирович, программист, Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления СО РАН (Иркутск, Российская Федерация), skorov@icc.ru

Поступила в редакцию 10 апреля 2014 г.

METHODS AND TOOLS OF DISTRIBUTED COMPUTING SYSTEMS METAMONITORING

I.A. Sidorov, Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS (Irkutsk, Russian Federation)

A.P. Novopashin, Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS (Irkutsk, Russian Federation)

G.A. Oparin, Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS (Irkutsk, Russian Federation)

V.V. Skorov, Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS (Irkutsk, Russian Federation)

This paper discusses the problems of monitoring a distributed computing environment that includes a lot of loosely-coupled heterogeneous computing resources. The principal distinctive feature and novelty aspect of the developed metamonitoring system first of all consists in creation of the universal software agents, capable to collect data about state of different environment components from local monitoring systems, to unify and analyze these data, and if necessary to generate and execute automatic control actions directed on changing equipment working mode, including the shutdown in case of emergencies. The approach is based on web-technologies and multi-agent technologies usage, methods of expert systems creation, methods of decentralized processing and the distributed storage of data.

Keywords: distributed computing systems, monitoring, multiagent technologies, expert systems

References

1. Adinets, A.V., Bryzgalov P.A., Voevodin V.V. Monitoring, analiz i vizualizaciya potoka zadanii na klasternoi sisteme [Monitoring, analyze and visualization of tasks flow on cluster system]. Visokoproizvoditelnye parallelnye vychisleniya na klasternih sistemah [High-performance parallel computation on cluster systems]. Nizhny Novgorod, 2011. P. 10-14.
2. Sidorov I.A., Skorov V.V. Odin iz podhodov k realizacii sredstv monitoringa visokoproizvoditelnykh vychislitelnykh system [Approach to the monitoring system implementation for HPC clusters] Informacionnye i matematicheskie tehnologii v nauke i upravlenii [Informational and mathematical technologies in science and control]. Irkutsk, Publishing of MESI SB RAS, 2013. P. 177-184.
3. Wooldridge M., Jennings N. Intelligent Agents: Theory and Practice // The Knowledge Engineering Review, 1995. Vol. 10, No 2. P. 115-152.
4. System Information Gatherer and Reporter API. URL: <http://www.hyperic.com/products/sigar> (accessed: 08.04.2014).
5. Standard ECMA-262: ECMAScript Language Specification. URL: <http://www.ecma-international.org/publications/standards/Есma-262.htm> (дата обращения: 08.04.2014)
6. Chastikov A.P., Belov D.L., Gavrilova T.A. Razrabotka ekspertnykh sistem. Sreda CLIPS [Developing of expert systems. CLIPS Toolkit]. Saint-Petersburg, 2003. 393 P.

7. Skorov V.V., Sidorov I.A., Novopashin A.P Redaktor graficheskogo predstavleniya ob-
jectov inženernoi infrastruktury vichislitelnogo tsentra [Editor of graphical representation
of engineering infrastructure objects of HPC-center] // Lyapunovskie chteniya [Lyapun-
ov reading]. Irkutsk, Publishing at ISDCT SB RAS, 2013. P. 53.
8. Irkutsk supercomputer center URL: <http://hpc.icc.ru> (accessed: 08.04.2014)
9. MAKER – genome annotation pipeline. URL: <http://gmod.org/wiki/MAKER> (ac-
cessed: 08.04.2014)

Received 10 April 2014.