

УДК 004.94+547.022

# СИСТЕМА МОНИТОРИНГА СОСТОЯНИЯ РЕСУРСОВ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

**Базаров Д.К.**

м.н.с. Научно-инновационного центра информационно-коммуникационных технологий,  
тел.: +(99890) 187-48-02, e-mail: marvelday@mail.ru

Распределенная вычислительная система представляет собой набор территориально разнесенных физических и виртуальных узлов различного назначения, выход из строя которых может существенно снизить производительность и надежность всей вычислительной системы в целом. Описана активная система мониторинга состояния компонентов аппаратного обеспечения распределенной вычислительной системы, ориентированной на решение задач в области химической информатики. Приводятся общие понятия и сформулированы задачи мониторинга, дается обзор компонентов и готовых систем мониторинга.

**Ключевые слова:** распределенная вычислительная система, активная, пассивная, модульная система мониторинга, мониторинг задач и ресурсов.

## THE MONITORING SYSTEM OF RESOURCES OF A DISTRIBUTED COMPUTING SYSTEM

**Bazarov D.K.**

The distributed computing system is a set of geographically distributed physical and virtual nodes of different purposes. The failures of these nodes may significantly decrease the performance and reliability of the entire computing system as a whole. The article describes active system monitoring of the hardware components of the distributed computing system designed for chemical informatics problems. The general concept, monitoring tasks, review of components of a monitoring system are discussed.

**Keywords:** distributed computing system, active, passive, and modular monitoring system, monitoring of the tasks and hardware resources.

## TAQSIMLANGAN HISOBLASH TIZIMI RESURSLARI HOLATINI MONITORING QILISH TIZIMI

**Bazarov D.K.**

Taqsimlangan hisoblash tizimi turli maqsadlardagi xududiy tarqalgan fizik va virtual uzellar to'plamini o'zida mujassamlashtirgan. Ushbu uzellardan birortasining ishdan chiqishi barcha hisoblash tizimini to'raligicha unumdorligini va ishonchligini sezilarli darajada pasaytirishi mumkin. Maqolada kimyoviy informatika sohasidagi masalalarni yechishga yo'naltirilgan taqsimlangan hisoblash tizimi apparat ta'minoti komponentlarining holatini aktiv monitoring qilish tizimi yoritilgan. Umumiy tushunchalar keltirilgan, monitoring qilishning vazifalari rasmiylashtirilgan, komponentlarning tahlili va monitoring qilishning tayyor tizimi berilgan.

**Tayanch iboralar:** taqsimlangan hisoblash tizimi, aktiv, passiv va modulli monitoring tizimi, topshiriqlar va resurslarni monitoring qilish.

### 1. Введение

Распределенная вычислительная система (РВС) с точки зрения ее администратора представляет собой множество территориально разнесенных физических и виртуальных узлов различного назначения. Причем в РВС всегда есть такие компоненты, выход из строя или задержки в работе которых могут оказать существенное влияние на производительность и надежность работы всей системы в целом.

Чтобы своевременно обнаруживать неисправности, находить узкие - уязвимые с точки зрения производительности - места в работе РВС, избавляться от избыточных или, напротив, дублировать важные ее компоненты, администратор

должен иметь общий мониторинговый центр, в котором он в интерактивном режиме смог бы получать актуальную информацию о текущем состоянии всех аппаратных, программных и информационных ресурсов РВС.

Совместное использование методов управления и мониторинга помогут в планировании и заблаговременном обновлении РВС недостающими компонентами. Так, оценка скорости наполнения элементов подсистемы хранения РВС позволит спрогнозировать потребности каждого ресурсного центра в жестких дисках.

В статье представлены основные функции, компоненты и классификация существующих систем мониторинга (СМ). Рассмотрены особенности СМ

РВС «Информационной системы обработки физико-химических данных» (ИСОД) [1].

## 2. Методы, задачи, типы и компоненты систем мониторинга

В задаче мониторинга можно выделить две взаимодействующие системы: *подконтрольную систему*, для эффективного управления которой применяются *средства мониторинга*, и *систему мониторинга*, объединяющую эти средства. *Объект мониторинга* – это элемент подконтрольной системы, за состоянием которого осуществляется наблюдение с целью управления, анализа протекающих с его участием процессов, выявления и прогнозирования нештатных состояний [2].

Подконтрольной системой является РВС, а объектами мониторинга – компоненты узлов аппаратных ресурсов ресурсных центров РВС, информационные и программные ресурсы и выполняемые на этих ресурсах вычислительные эксперименты (ВЭ), представляемые в виде потоков заданий. В соответствии с этим различают [3] система-ориентированный и задача-ориентированный типы мониторинга. СМ первого типа предоставляют данные о функционировании подконтрольной системы в целом, второго – анализируют поведение заданий определенного ВЭ. СМ, ориентированные на задания, находят применение в задачах распределения нагрузки на ресурсы РВС. Они могут быть полезны для пользовательских приложений, способных изменять требования к ресурсам в процессе своего выполнения [4].

Далее в статье под термином «система мониторинга» будем понимать система-ориентированную СМ. Задание-ориентированные СМ – предмет дальнейших исследований.

СМ различают [5]:

- *по объектам мониторинга*: сеть (PRTG, Cacti, OpenNMS), аппаратное обеспечение (диагностика устройств хранения, центральных и графических процессоров, оперативной памяти, доступность ресурсов), программное обеспечение (системное, прикладное, промежуточное, интерфейсное); анализ конечных и промежуточных результатов вычислительных экспериментов;

- *по технологиям сбора данных*: по протоколам WMI, TCP, UDP, ICMP, HTTP, SNMP (Monitorix, Munin, PRTG, Cacti, OpenNMS), или на основе агентов (Shinken, Zabbix);

- *по условиям использования*: коммерческие, условно-бесплатные (Opmantek NMIS), свободные (Zabbix, Nagios core, Icinga);

- *по типу реакции*: пассивный (MRTG, Cacti) и активный мониторинг (Zabbix, HP Open View, IBM Trivoli);

- *по архитектуре построения*: централизованные (Munin) и распределенные (Ganglia, Shinken, openNMS);

- *по целям мониторинга*: обеспечение надежности, безопасности или оптимизация

производительности;

- *по способу реализации*: интегрированные и модульные.

СМ должна уметь:

- собирать данные – опрашивать объекты мониторинга с заданными временными интервалами для получения значений параметров этих объектов (Diamond, Riemann);

- накапливать, хранить и архивировать данные о результатах проверок. Для этого ей необходимы компоненты для работы с базами данных (OpenTSDB, InfluxDb, BigGraphite, MongoDB).

- проводить анализ накопленных данных с помощью так называемой системы сбора и корреляции событий [6] (ArcSight, Q1Radar, Security Information Manager, LogRhythm, GFI EventsManager, Netwrix Event Log Manager);

- оповещать администратора о событиях, происходящих на ресурсах РВС (PRTG Network Monitor, Zabbix, Seyren, Cabot, Nagios, Riemann);

- предоставлять информацию о состоянии объектов мониторинга РВС в удобном для восприятия виде;

- выполнять действия по устранению критических сбоев в компонентах РВС.

Примерами интегрированных СМ, выполняющих большинство из перечисленной выше функций, являются: Prometheus [7] – распределенная иерархическая пассивная СМ для контроля сотен серверов; Zabbix [8] – распределенная свободная СМ сетевого оборудования, серверов и сервисов; Ganglia [9] – масштабируемая распределенная СМ кластерных и грид-систем.

Среди СМ, ориентированных на анализ заданий, следует выделить LAPTA [10]. Она предназначена для анализа динамических характеристик параллельных программ, выполняемых на суперкомпьютерах. Эта система тесно интегрирована с известными средствами управления заданиями. Сбор информации о ресурсах осуществляется с привязкой к конкретной задаче пользователя, что является, по мнению авторов [11], наиболее верным направлением развития СМ вычислительных кластеров и РВС. Одним из первых шагов в этом направлении является разработка программного модуля визуализации промежуточных результатов вычислений, в частности молекулярного докинга [12] для РВС, ориентированных на исследования в области химической информатики.

В работе [13] описана СМ РВС Объединенного института ядерных исследований РФ (г. Дубна) на основе Nagios. Мониторинг ресурсов РВС на основе промежуточного ПО globus toolkit описан в работе [11]. Сравнительный анализ централизованной и иерархической архитектур СМ для РВС на основе грид-систем представлен в [14].

Мониторинг событий на основе ganglia при работе с облачными сервисами Webzilla описан в [15]. В работе [11] предлагается подход к созданию СМ программно-аппаратных ресурсов РВС с использованием мультиагентных технологий, децентрализованной схемы хранения данных и элементов экспертных систем. В [16] свободно-

распространяемые средства контроля и диагностики вычислительных ресурсов РВС, такие как Ganglia, Nagios, MonALISA, CeMon, и вовсе рассматриваются как неотъемлемая часть программного обеспечения пространственно-распределенных вычислительных систем.

К сожалению, почти у всех таких свободных решений есть один существенный недостаток: одной системой нельзя покрыть все задачи мониторинга. К примеру, в Ganglia нет механизмов обработки текущих значений и уведомлений о событиях. Nagios и Zabbix лишены этого недостатка, однако их ориентация на мониторинг сетевого оборудования делает их малоприспособленными для РВС [11]. В настоящее время интерес представляют модульные системы, в которых имеется возможность построения СМ интеграцией отдельных компонентов сбора, хранения, визуализации, оповещения. Одной из первых таких систем является Graphite [17] — библиотека для хранения и визуализации данных. Для создания полноценной СМ на ее основе требуются сторонние приложения, полный список которых представлен в [18].

СМ с модульной архитектурой содержат следующие компоненты.

**Источники данных** - сенсоры, осуществляющие опрос оборудования или программного обеспечения с целью получения текущего значения контролируемых величин в виде метрик и передачи их в центры сбора данных (ЦСД). ЦСД выполняют предварительную обработку значений, проверку выполнения условий на значения метрик, фильтрацию и генерацию уведомлений о событиях. В распределенных СМ ЦСД обычно организованы в иерархическую структуру.

Поскольку контролируемых параметров много и каждый из них имеет огромное число значений, хранить их все в оперативной памяти не имеет смысла. Метрики выгружаются в специализированные хранилища данных и извлекаются по мере необходимости. Для оповещения о состоянии объекта мониторинга используются средства визуализации, информация в которых представляется в виде дашбордов - документов с лаконично представленными статистическими данными и отчетами с элементами инфографики.

**Источники и средства сбора данных.** Данные мониторинга ресурсов могут быть получены или импортированы из множества источников: HTTP APACHE, Apache Tomcat, IBM DB2, IBM Tivoli, IBM Websphere PMI, JBOSS, JVM, ORACLE, SAP NetWeather, SNMP, rstatd UNIX, системный монитор WINDOWS, журналы событий, межсетевые экраны, активное сетевое оборудование, сканеры уязвимостей и др.

Поток данных с источников далее обрабатывается средствами сбора данных. Наиболее популярными из них являются:

**Collectl** - средство сбора метрик о состоянии центрального процессора, памяти, файловых систем (nfs, lustre), процессах, сетевой активности и т.п., запускаемое в интерактивном и в фоновом режимах.

Выдает данные в сжатом, удобном для чтения формате;

**DimondD** - демон сбора системных метрик. Написан на языке Python, предоставляет возможность разработки пользовательских средств для сбора метрик из любого источника. Состоит из коллекторов - отдельных демонов-парсеров, добывающих информацию из системы. Фактически коллекторами покрыт весь спектр объектов мониторинга [19];

**Riemann** собирает информацию о событиях как с аппаратных узлов, так и с приложений, установленных на этих узлах, объединяет события в потоки и передает их другим приложениям для дальнейшей обработки и хранения. Отслеживает состояние событий, что позволяет создавать проверки и рассылать уведомления;

**Telegraf** - агент сбора, предварительной обработки, агрегации и записи метрик [20].

**Хранилища данных.** В качестве хранилищ в зависимости от объема и характера сохраняемой информации используют или реляционные СУБД типа MySQL, PostgreSQL, или специализированные базы данных временных рядов (БДВР, англ. time series databases) имеющих, как правило, не реляционную модель хранения. Главная отличительная черта этих СУБД - цикличность записи. Их еще называют кольцевыми БД (англ. Round Robin database), поскольку общий объем хранимых в ней данных не изменяется со временем: ячейки памяти перезаписываются спустя определяемое администратором «время жизни», в течение которого данные актуальны.

Временной ряд порождается в результате наблюдения за одним или несколькими параметрами некоторого процесса. При наблюдении фиксируются значения параметров и привязываются к моменту наблюдения. В результате и образуются последовательности значений, упорядоченные по времени, называемые временными рядами.

Элемент временного ряда - измерение (measurement) - состоит из имени метрики, *временной метки* (timestamp), значения (value) и набора тегов (tag) - пар «ключ - значение» [21].

В [22] приведена классификация БДВР по архитектуре построения, способам внесения и извлечения данных для анализа, модели хранения и политике лицензирования. Сравнительная характеристика реляционных (PostgreSQL) и нереляционных СУБД (OpenTSDB, InfluxDB, MongoDB) для мониторинга технологических процессов на промышленных объектах приведена в [23]. В последней работе оценка проводилась по критериям зависимости пропускной способности на запись от числа пользователей, объема одновременно записываемой информации, времени исполнения запроса на чтение; степени сжатия данных; пропускной способности при нагрузочном тестировании (т.е. в течение нескольких суток).

Поскольку практически по всем приведенным выше показателям лучшей оказалась InfluxDB, то в создаваемой СМ в качестве СУБД для хранилища метрик используется именно эта БДВР.

InfluxDB (октябрь 2013 г.) [24] - открытая СУБД для хранения временных рядов, метрик и информации о событиях. Имеет распределенную архитектуру, отдельные узлы которой формируют так называемый кластер. Извлечение и сохранение данных на всех узлах кластера выполняются одновременно. Время жизни (цикл перезаписи) и число узлов кластера определяются политикой хранения. Данные могут вноситься через консольный клиент (CLI), клиентские библиотеки языков Python, Java, Javascript, Ruby, Go и непосредственно по протоколу http. Поддерживается собственная версия этого протокола, а также протокол системы мониторинга Graphite и базы данных OpenTSDB.

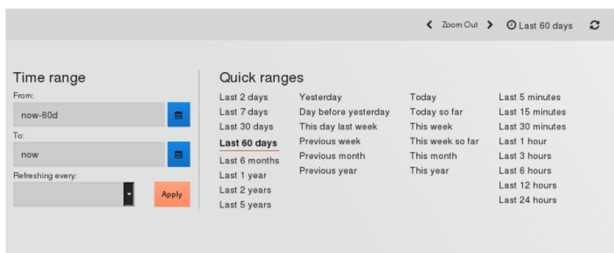
Для низкоуровневого хранения используется способ построения журнально-структурированного дерева со слиянием или LSM-дерево (Log Structured merge tree), что обеспечивает быстрый доступ по индексу в условиях частых запросов на вставку, например, при хранении журналов транзакций [25].

Запрос данных осуществляется на специально написанном для этой БДВР языке InfluxQL. Приводим пример использования этого языка:

```
> select temperature, from hddtemp where time > now() - 4m;
name: hddtemp
time           temperature
-----
1491828310000000000    36
```

Команда `select` с аргументами `temperature`, `hddtemp` и `time > now() - 4m` извлекает из таблицы `hddtemp` значения полей `time` и `temperature` за последние четыре минуты наблюдений (4 m). Здесь `now()` - функция, возвращающая текущее время. В столбце `time` указана временная метка `1491828310000000000`, характеризующая время получения данного значения температуры (36° C).

Модель хранения данных, используемая в БДВР InfluxDB, теоретически позволяет хранить, извлекать, а в дальнейшем и визуализировать в виде графиков измерения за период времени от пяти минут до пяти лет (рис. 1 а).



а)



б)

Рис. 1. Допустимый диапазон значений времени (*time range*) для извлечения и анализа данных в БДВР Influxdb (а) и показатели загрузки памяти и центрального процессора по ядрам за последнюю неделю (б)

**Визуализация.** После извлечения метрик из источников и сохранения их в БДВР возникает необходимость их обработки и визуализации. На сегодняшний день имеется огромное число средств визуализации данных из БДВР; рамки данной статьи не позволяют перечислить их все. Рассмотрим самые известные.

Grafana [26] — редактор графиков и дашбордов, основанный на данных из Graphite, InfluxDB или OpenTSDB, специализирующийся именно на отображении и анализе информации. Новые версии Grafana могут работать и с другими источниками данных, такими, например, как Elasticsearch. Помимо predefined системных метрик (CPU, IO, etc.), можно сконфигурировать любой набор произвольных метрик — онлайн, профит и т.д. Grafana содержит две темы оформления, что позволяет создавать произвольное число дашбордов. Grafana по сути - это замена Graphite-web, которая помогает пользователям проще создавать и редактировать дашборды, она содержит уникальный парсер для редактирования метрик и функций.

Kibana — это платформа для аналитики и визуализации, позволяющая исследовать, визуализировать и строить дашборды поверх данных лог-файлов, сохраняемых в кластерах. Можно также выполнить дополнительный анализ и визуализацию данных, используя различные типы графиков, таблиц и отображений.

Основное различие между grafana и kibana в том, что первый ориентирован на представление данных временных рядов, основанных на таких метриках, как центральный процессор и операции ввода-вывода. Kibana позволяет создавать дашборды с комплексной аналитикой. А вот grafana не может выполнять поиск и анализ данных [27].

**Средства оповещения.** Поскольку ресурсы РВС работают круглосуточно, а администраторы способны анализировать показатели только в дневное время, возникает необходимость создания средств оповещения об изменении состояния объектов наблюдения в подконтрольной системе. Если объектом наблюдения является вычислительный эксперимент, то пользователя можно оповестить об изменении статуса, отправленного им задания на удаленных вычислительных ресурсах. В случае некорректного завершения задания пользователь обращается к администратору ресурсного центра с целью выяснения, что послужило тому причиной: перегрев аппаратных компонентов ресурса, его отключение или программная ошибка. Администратор может дать соответствующее заключение на основе анализов данных мониторинга из БДВР. Оповестить администратора или пользователя можно по почте или по телефону. Рассмотрим два средства оповещения.

В Seugen по каждой метрике создаются «проверки», имеющие пределы WARN (тревога) и ERROR (ошибка), указывается канал (Email, Flowdock, HipChat, HTTP, Hubot, IRCcat, PagerDuty, Pushover, SLF4J, Slack, SNMP, Twilio), по которому происходит оповещение «подписчиков» в

соответствии с «расписанием оповещений», например, по будням [19]. В более популярном [28] Sabot имеется возможность создавать проверки на результат применения функций к метрикам Graphite. Каналы оповещения: e-mail, hipchat, sms, telephone [29].

**Активная система мониторинга.** Иногда требуется немедленная реакция на некоторые события, такие, например, как стихийные бедствия, несанкционированный доступ в серверное помещение, отсутствие удаленного доступа к ресурсам. Администратор не в состоянии быстро отреагировать на то или иное событие, последствия которого обычно приводят к потере критически важных данных. В таких случаях есть смысл разработать сценарии реакции на конкретные события. Далее приводим пример сценария отключения рабочего узла в случае превышения температуры жесткого диска выше определенного критического значения:

```
#!/bin/sh
TEMP_CRIT=50; #предельная температура
POLLING_TIMEOUT="5s"; #интервал проверки
DEVICE="/dev/sda" #устройство
echo "Starting with PID" `pidof sh $0`;
echo "получаем температуру\n"
TEMP=`hddtemp -n $DEVICE`;
echo "пока температура не достигнет предельно-допустимой"
while ((TEMP <= TEMP_CRIT)); do
  TEMP=`hddtemp -n $DEVICE`;
  sleep $POLLING_TIMEOUT;
done
echo "Температура жесткого диска достигла критической
отметки\n";
echo "Выключаем машину";
poweroff
```

Такое часто происходит в летнее время при отключении системы охлаждения в серверном помещении в результате скачков напряжения.

В данном конкретном примере при `TEMP > TEMP_CRIT` узел отключается командой `poweroff`. В активной системе мониторинга ИСОД, описываемой в следующем разделе, реакцией на данное событие является внеочередной вызов программы полного резервного копирования на удаленные информационные ресурсы дампа базы исходных данных, интерфейсных, инфраструктурных узлов.

### 3. Система мониторинга информационной системы обработки физико-химических данных Института химии растительных веществ АН РУз

Рассматриваемая ИСОД предназначена для накопления, хранения, обработки и систематизации новых соединений, синтезируемых учеными-химиками Узбекистана. Она состоит из подсистем доступа, хранения и обработки физико-химических данных о соединениях — кандидатах в лекарственные средства.

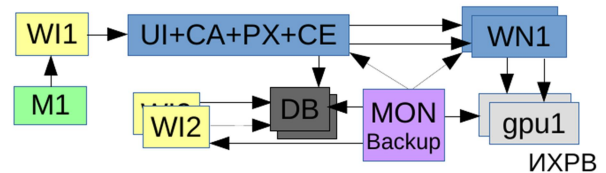


Рис.2. Ресурсный центр Института химии растительных веществ АН РУз

Компонентами ресурсного центра ИСОД являются (рис. 2):

- *интерфейсные узлы* к подсистемам обработки (**web-interfaces**, WI1) и хранения исходных физико-химических данных WI2,3;
- *база исходных данных и конечных результатов* обработки (**data base**, DB) и ее копия на удаленном информационном ресурсе;
- *инфраструктурный узел* ИСОД, состоящий из пользовательского интерфейса (**user interface**, UI), центра сертификации новых ресурсов и пользователей (**certificate authority**, CA), хранилища пользовательских сертификатов (**проху**, PX) и узла управления (**computing element**, CE) кластером, состоящим из рабочих узлов (**worker nodes**, WNx);
- *графические устройства* (**graphics processing units**, gpu) при рабочих узлах ресурса;
- *система мониторинга* (**mon**) и *резервного копирования* (**backup**) важных компонентов ИСОД;
- *модуль визуализации* (**M1**) исходных данных для вычислительных экспериментов.

Предлагаемая СМ ИСОД построена в виде модульной структуры, основу которой составляют три свободно распространяемые компоненты: **Telegraf** [20], **influxDB** [24], **grafana**[26], осуществляющих, соответственно, сбор, хранение и визуализацию метрик. Установка и конфигурирование этих компонент (модулей) представлены в [30].

Основные отличия модульного подхода к построению СМ от готовых решений состоят в следующем:

1. Возможность подбора компонентов от различных производителей под конкретные нужды администраторов мониторируемых ресурсов.
  2. Процедура подбора, установки и настройки компонентов СМ по трудоёмкости сравнима с процедурами построения и проблемной ориентации самих РВС.
  3. Модульные СМ в большей степени ориентированы на свободно-распространяемые версии базового программного обеспечения: **unix**-подобные операционные системы, чем на операционные системы семейства **MS Windows**. С одной стороны, это позволяет комбинировать продукты различных производителей, не оглядываясь на лицензионные соглашения, с другой стороны, постоянно обновляемые компоненты модульной СМ не застрахованы от ошибок, что в известной степени затрудняет их использование в коммерческих крупномасштабных проектах.
- Набирающий силу процесс унификации и стандартизации протоколов взаимодействия

программных средств, используемых в качестве компонентов модульных СМ, дает основание полагать, что в будущем перечисленные выше недостатки модульных СМ будут устранены [31].

**Объектами мониторинга** являются сервер хранения физико-химических данных (DB), интерфейсные (WI), инфраструктурный UI+CA+PX+CE) и вычислительные (WN+GPU) узлы.

СМ интегрирована в интерфейсный узел ИСОД (WI1).

Источники данных для агента сбора Telegraf - утилиты lmsensors, hddtemp, снимающие показания физических датчиков различного оборудования. Графическое представление их значений формируется на сервере мониторинга WI1.

В представленной СМ в реальном времени осуществляется контроль следующих параметров аппаратных ресурсов:

- системная информация: число ядер в центральном процессоре (CPU) и его загрузка (inputs.cpu), объем оперативной памяти и распределение свободной и используемой памяти (inputs.mem), усредненный показатель загрузки сервера по процессам в операционной системе и время работы сервера с момента последнего включения (inputs.system), процент использования дискового пространства в МВ (inputs.disk). Показатели температур жестких дисков, собираемых утилитой hddtemp (inputs.hddtemp), южного моста, CPU, GPU, скорости вращения системных вентиляторов, собираемых утилитой lm-sensors (inputs.sensors);

- объем входящего и исходящего трафиков сетевых интерфейсов (inputs.net);

- информация о состоянии базы данных по физико-химическим свойствам органических соединений (inputs.mysql): число подключенных пользователей, количество удачных и неудачных попыток подключения к БД, объем внесенных и извлеченных из БД данных за промежуток времени, задаваемый пользователем.

#### Контроль состояния графических устройств.

Агент telegraf может принимать данные с различных источников. Если при перегреве, например, центрального процессора срабатывает механизм защиты путем снижения его тактовой частоты (throttling) или отключения питания материнской платы, то для GPU такого механизма нет. Следовательно, возникает необходимость создания системы контроля состояния и защиты от перегрева графического устройства, возникающего в результате интенсивных вычислений или выхода из строя системы его охлаждения.

При установке драйвера графического устройства устанавливается и утилита nvidia-smi, отображающая температуру графического процессора, скорость вращения вентиляторов видеокарты, объем занятой глобальной памяти устройства. В Telegraf имеется возможность получения данных с датчиков, изначально не поддерживаемых этой программой. Для этого в конфигурационном файле /etc/telegraf/telegraf.conf прописывается путь к программе сбора данных, в данном конкретном

случае выдаваемых консольной утилитой nvidia-smi. Эти данные сохраняются в БДВР и визуализируются с помощью Grafana. В этом состоит пассивный мониторинг. Полученные данные контролируются и анализируются другой программой, которая при превышении заранее прописанных параметров принудительно выключает узел, на котором установлено графическое устройство. Так осуществляется активный мониторинг графических устройств узлов PVC. Рассмотрим подробнее содержимое конфигурационного файла telegraf.conf.

В разделе inputs.exec приведены команды вызова скриптов: ppmgpu.sh, tempgpu.sh, memgpu.sh, allmemgpu.sh, которые собирают информацию о загрузке глобальной памяти, скорости вращения вентиляторов и температуры графического процессора соответственно. Эти данные в виде метрик записываются в БДВР InfluxDb:

```
[[inputs.exec]]
# ## Commands array
commands = [
  "/home/telegraf/sensors/ppmgpu.sh",
  "/home/telegraf/sensors/tempgpu.sh",
  "/home/telegraf/sensors/memgpu.sh",
  "/home/telegraf/sensors/allmemgpu.sh"
]
name_suffix = "gpu"
```

Далее приводим пример скрипта tempgpu.sh, расположенного в каталоге /home/telegraf/sensors/. Этот скрипт выдает температуру графического сопроцессора:

```
#!/bin/bash
/bin/echo {"GPU Temperature": $(awk '/MiB/ {printf "$3\n"}' <(nvidia-smi))} | tr -d "C" | json_pp
```

Здесь nvidia-smi — утилита для сбора данных о видеокарте, json\_pp — программа, преобразующая данные в формат json, воспринимаемый БДВР InfluxDB.

А вот результат консольного запуска этого скрипта, представленный в json-формате:

```
dima@rudeb:~$ ./memgpu.sh
{
  "GPU Temperature" : 27
}
```

На рис. 3 представлены полученные с помощью визуализатора Grafana текущие и предельно допустимые значения датчиков температур жестких дисков, центрального и графического процессоров, южного моста, а также соответствующие значения скоростей вращения системных вентиляторов.

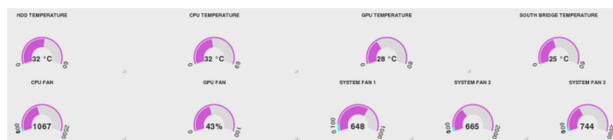


Рис. 3. Визуализация значений датчиков материнской платы объекта мониторинга

Для отображения текстовой информации, выдаваемой скриптами полного резервного

копирования интерфейсов доступа и дампа базы данных химических соединений, в Grafana используется AJAX Plugins. На рис. 4, в частности, показаны результаты выполнения процедуры копирования измененных файлов интерфейса пользовательского доступа с WI2 на WI3 (см. рис.2). Файлы указанного объема с указанной скоростью были скопированы 27.02.2017 г. в 15:31 по местному времени.



Рис. 4. Информация о результатах копирования измененных файлов с одного интерфейса доступа (WI2) к БД на другой (WI3)

Администратор может получить информацию о количестве пользовательских обращений (например, 256) к базе данных химических соединений, объеме исходящего (например, 137.7МВ) и входящего (например, 6.3МВ) трафика за указанный промежуток времени (например, за 1 день). Эти результаты приведены на рис. 5.

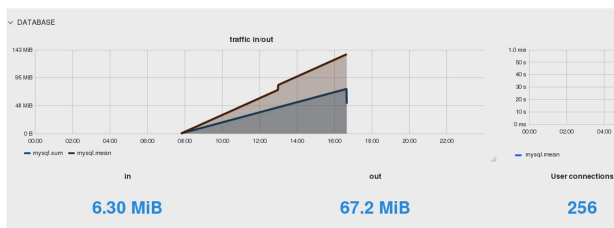


Рис. 5. Информация о входящем и исходящем трафике БД химических соединений ИСОД

Согласно графику, представленному на рис. 5, все 256 пользовательских обращений к БД осуществлялись в период времени с 8:00 до 15:20.

**Распределенная СМ.** В модульной системе мониторинга есть возможность разнести ее компоненты на несколько территориально удаленных серверов. Например, визуализатор grafana может содержать несколько дашбордов, каждый из которых строит графики по метрикам с нескольких серверов, доступных как по локальной сети, так и через интернет. Хранилища и агенты сбора данных устанавливаются на объектах мониторинга. Суточный объем потребляемого трафика по указанным выше метрикам составляет около 200 МВ.

Использование распределенной СМ имеет свои ограничения, из которых самым главным является небезопасность передачи данных между визуализатором grafana и БДВР influxdb. Это связано с тем, что протокол передачи не использует шифрование. В этой связи можно использовать vpn-соединение между указанными выше компонентами

распределенной СМ.

Вторым ограничением, но не таким существенным, является потребляемый трафик. Но так как местные оптические интернет-каналы поддерживают высокую скорость передачи, например, Tasics от 2MBit/sec и выше, то объемом потребляемого трафика можно пренебречь. Система легко масштабируется за счет добавления новых элементов в сервер grafana и установки агента telegraf и БДВР influxdb на объекты мониторинга.

#### 4. Обсуждение результатов и выводы

Модульный подход к созданию СМ состоит в переходе от индивидуального к кооперативному проектированию СМ, в активном вовлечении конечных пользователей в сам процесс создания СМ. Это сокращает сроки и стоимость проектирования, дает возможность повторного использования универсальных компонентов и пользовательских наработок.

Таким образом, разработанная модульная СМ ИСОД перед интегрированными аналогами имеет следующие преимущества:

1. Открытая, состоит из бесплатных свободно-распространяемых компонентов.

2. Используемая для долговременного хранилища БДВР InfluxDB превосходит аналоги, входящие в состав готовых СМ.

3. Агент сбора данных telegraf позволяет принимать информацию из любого доступного источника. Для этого достаточно указать соответствующую команду в секции [[inputs.exec]] конфигурационного файла: telegraf.conf.

Визуализатор Grafana в предлагаемой структуре СМ интегрирован в портал WS-PGRADE gUSE (рис. 6). На рис. 2 он обозначен через WI1. Помимо Grafana, WI1 содержит вкладку Workflow, где пользователь формирует поток заданий на ресурсы, информацию о которых он получает на вкладке Information. Storage обеспечивает доступ к ресурсам хранения PVC, Security предназначена для получения пользовательских сертификатов на ресурсы, а Docking — модуль визуализации M1 (рис. 2).

Таким образом, пользователи регистрируются в PVC, просматривают структуры химических соединений — исходных данных для ВЭ, запускают задания, анализируют результаты, а администратор с того же интерфейса доступа к ресурсам ИСОД следит за состоянием компонентов аппаратного обеспечения этой PVC и, в случае возникновения ошибок в пользовательском задании, сможет оперативно установить природу их возникновения: аппаратный или программный сбой.

На скрипты полного резервного копирования интерфейсов доступа и БД по физико-химическим свойствам органических соединений получено свидетельство [32].

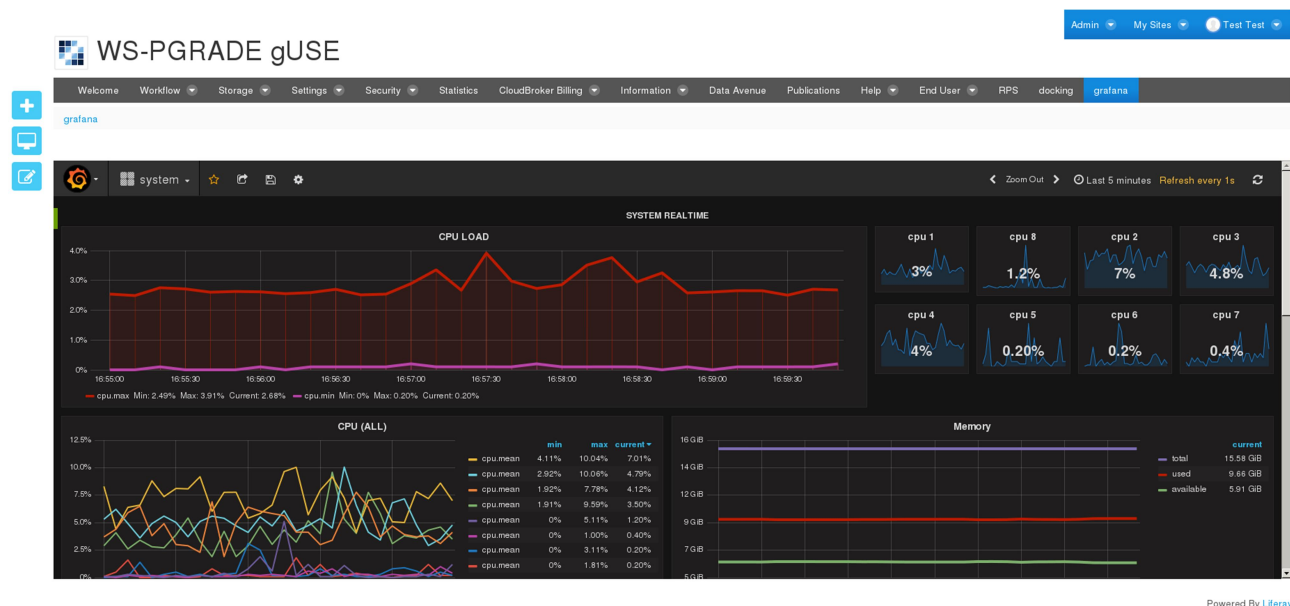


Рис. 6. Мониторинг ресурсов РВС ИСОД

## Литература

- [1] Базаров Р.К., Базаров Д.К., Брусков В.П. Информационная система для обработки физико-химических данных органических соединений // Современное состояние и перспективы применения информационных технологий в управлении: Доклады Республиканской научно-технической конференции. - Джизак, 5-6 сентября 2016. - Ташкент, 2016. - С. 213-219.
- [2] Дмитриенко П.В. Методика оценки эффективности систем мониторинга вычислительных ресурсов // Компьютерные исследования и моделирование. - 2012. - Т.4, № 3. - С. 661-668.
- [3] Тарасов А.Г. Система мониторинга вычислительного кластера расширенной функциональности: Дис... канд. техн. наук. - Хабаровск, 2011. tarasov\_diser\_20101125.pdf
- [4] Поляков А.Ю., Молдованова О.В., Пазников А.А., Курносоев М.Г., Мамоиленко С.Н., Ефимов А.В. Алгоритмы отказоустойчивого управления ресурсами пространственно-распределенных вычислительных систем // Вестник СибГУТИ. - 2014. - № 4. - С. 11-29.
- [5] Mauno Ronkko, Ville Kotovirta, Mikko Kolehmainen. Classifying Environmental Monitoring Systems // Proceedings of 10th IFIP WG 5.11 International Symposium, ISESS 2013, Neusiedl am See, Austria, October 9-11. - 2013. - Pp. 533-542.
- [6] Цветков А.А. Задачи обработки данных мониторинга ресурсов распределенной вычислительной сети // Интернет-журнал «Науковедение». - 2014. - № 4 (23).
- [7] Prometheus - Monitoring system & time series database [Электронный ресурс] / Prometheus Authors. - Режим доступа: <https://prometheus.io>
- [8] Zabbix - The Enterprise-class Monitoring Solution for Everyone [Электронный ресурс] / Zabbix LLC. - Режим доступа: <http://www.zabbix.com>.
- [9] Ganglia Monitoring System [Электронный ресурс] / Ganglia. - Режим доступа: <http://ganglia.info>
- [10] Адинец А.В., Брызгалов П.А., Воеводин Вад.В. Мониторинг, анализ и визуализация потока заданий на кластерной системе // Высокопроизводительные параллельные вычисления на кластерных системах: Материалы XI Всерос. конф. - Нижний Новгород, 2011. - С. 10-14.
- [11] Сидоров И.А., Новопащин А.П., Опарин Г.А., Скороев В.В. Методы и средства метамониторинга распределенных вычислительных сред // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. - 2014. - № 2. - С. 30-42.
- [12] Брусков В.П., Базаров Д.К., Базаров Р.К. Программный модуль визуализации промежуточных результатов молекулярного докинга // Проблемы вычислительной и прикладной математики. - Ташкент, 2016. - № 2. - С. 101-107.
- [13] Кореньков В.В., Мицын В.В., Дмитриенко П.В. Архитектура системы мониторинга центрального информационно-вычислительного комплекса ОИЯИ // Информационные технологии и вычислительные системы. - 2012. - № 3. - С. 31-42.
- [14] Симоненко В.П., Щербина О.В. Выбор архитектуры системы мониторинга ресурсов в глобальных GRID системах // Вестник НТУУ «КПИ». Информатика, управление и вычислительная техника. - Сборник научных трудов. - 2012. - № 57. - С. 25-29.
- [15] Мониторинг на основе данных [Электронный ресурс] / Блог компании Webzilla. - Режим доступа: <https://habrahabr.ru/company/webzilla/blog/220707>
- [16] Хорошевский В.Г., Курносоев М.Г., Мамоиленко С.Н., Поляков А.Ю. Архитектура и программное



- обеспечение пространственно-распределённых вычислительных систем // Вестник СибГУТИ. – 2010. - № 2. – С. 112-122.
- [17] Graphite [Электронный ресурс] / The Graphite Project. - Режим доступа: <https://graphiteapp.org>
- [18] Tools That Work With Graphite [Электронный ресурс] / The Graphite Project. - Режим доступа: <http://graphite.readthedocs.io/en/latest/tools.html>
- [19] В поисках идеального мониторинга [Электронный ресурс] / Блог компании Tinkoff.ru. - Режим доступа: <https://habrahabr.ru/company/tinkoff/blog/252907/>
- [20] Telegraf - The plugin-driven server agent for collecting & reporting metrics. [Электронный ресурс] / GitHub, Inc. - Режим доступа: <https://github.com/influxdata/>
- [21] *Намиот Д.Е.* Базы данных временных рядов и средства обработки // Актуальные проблемы системной и программной инженерии: Материалы 4-й Научно-практической конференции. – М.: НИУ ВШЭ, 2015. - С. 151-158.
- [22] A Comparison of Time Series Databases and Netsil's Use of Druid [Электронный ресурс] / Netsil's blog. - Режим доступа: <https://blog.netsil.com/a-comparison-of-time-series-databases-and-netsils-use-of-druid-db805d471206>.
- [23] Выбираем СУБД для хранения временных рядов [Электронный ресурс] / П. Филонов. Лаборатория Касперского. - Режим доступа: [https://www.slideshare.net/profyclub\\_ru/ss-68909134](https://www.slideshare.net/profyclub_ru/ss-68909134)
- [24] The Platform for Time-Series Data [Электронный ресурс] / InfluxData, Inc. - Режим доступа: <https://www.influxdata.com>
- [25] *O'Neil P. Cheng E., Gaulick D.* The log-structured merge-tree (LSM-tree) // Acta Informatica. – 1996. – Т. 33. – № 4. – Pp. 351-385.
- [26] Grafana. Beautiful metric & analytic dashboards [Электронный ресурс] / Torkel Ödegaard & Coding Instinct AB. - Режим доступа: <http://grafana.org>
- [27] *Grafana vs. Kibana: The Key Differences to Know* [Электронный ресурс] / Asaf Yigal Blog. - Режим доступа: <https://logz.io/blog/grafana-vs-kibana>
- [28] Compare Seyren and Cabot's popularity and activity [Электронный ресурс] / Awesome SysAdmin — Режим доступа: <https://sysadmin.libhunt.com/project/seymen/vs/cabot>
- [29] Cabot - monitor and alert [Электронный ресурс] / GitHub Project. - Режим доступа: <http://cabotapp.com/use/alerting.html>
- [30] Basic Telegraf, InfluxDB and Grafana setup [Электронный ресурс] / Jon Hayward. - Режим доступа: <https://fattylewis.com/Basic-Telegraf-InfluxDB-Grafana-setup/>
- [31] *Княгинин В.Н.* Модульная революция: распространение модульного дизайна и эпоха модульных платформ: учеб. пособие / Под ред. М.С. Липецкой, С.А. Шмелевой. – СПб., 2013. - 80 с.
- [32] *Брусков В.П., Базаров Д.К., Абдуллаев Н.Д.* Скрипты резервного копирования базы данных по физико-химическим свойствам органических соединений и интерфейса пользовательского доступа на удаленные информационные ресурсы на основе rsync // Агентство по интеллектуальной собственности РУз. Свидетельство № DGU03919. Заявитель и правообладатель Институт химии растительных веществ АН РУз. – № DGU20160424. 12.08.2016.