

## СИСТЕМА УПРАВЛЕНИЯ КЛАСТЕРОМ ДЛЯ КОМПЛЕКСОВ СЕМЕЙСТВА «ИНПАРКОМ»

Р.А. Ющенко

Институт кибернетики имени В.М. Глушкова НАН Украины,  
03187, проспект Академика Глушкова, 40,  
(044) 526 3603, pgt@ukr.net

Современные операционные системы не содержат собственных средств по управлению ресурсами и координации заданий вычислительного кластера. Поэтому разработчиками кластерных комплексов семейства «Инпаркком» создана система управления кластером АСМС. Рассмотрены предпосылки, архитектура и функциональность этой системы, а также особенности реализации.

Modern operating systems do not have means to control cluster resources and manage parallel jobs. So developers of INPARCOM HPC - cluster complex developed a cluster management system АСМС. Background, architecture, functionality and design features are described in this paper.

### Введение

Сложность разработки параллельных программ увеличивается за счет необходимости управлять не только логикой исполнения, а также данными (доступом и распределением), процессами и синхронизацией. Кроме того, инструменты и среды параллельного программирования не достигли уровня последовательных программ. Лет 10–15 назад у параллельного программиста практически не было высокоуровневых инструментов, всё приходилось программировать на низком уровне [1]<sup>1</sup>. Сейчас ситуация улучшилась в силу главным образом высокой популярности параллельных компьютеров с распределенной памятью, в особенности кластеров архитектуры Weowulf [2], построенных на основе широкодоступных комплектующих. Разработано множество бесплатно распространяемых средств управления и мониторинга параллельных компьютеров, разработана ставшая стандартом де-факто модель параллельных вычислений и программный интерфейс MPI, имеющие множество реализаций и обеспечившие разработку как готовых параллельных решений, так и параллельных математических библиотек. Поскольку такие библиотеки можно использовать и в последовательных программах, разрабатывать конкретные решения на параллельных компьютерах стало значительно проще [3].

Кластерные комплексы семейства «Инпаркком», которые Институт кибернетики имени В.М. Глушкова НАН Украины разрабатывает совместно с ГНПП «Электронмаш», – это пример готового инструмента, с одной стороны, предоставляющего системную инфраструктуру для выполнения параллельных приложений и разработки параллельных программ, а с другой, – позволяющего решать задачи без программирования [4]. За 5 лет работы линейка кластерных комплексов «Инпаркком» включает диапазон от 8 до 256 ядер. На начальных этапах разработки выяснилось, что операционная система не имеет собственных средств управления ресурсами кластерного комплекса как единого целого, поэтому их выполняют программы сторонних разработчиков. Именно эти программы формируют инфраструктуру параллельных вычислений и грид. Недостаточность стандартизации в области высокопроизводительных вычислений понижает интероперабельность компонентов инфраструктуры. Кроме того, часто централизованные средства интеграции и управления кластером жестко привязаны к аппаратуре и программному обеспечению конкретного комплекса. Параллельные прикладные программы для кластеров часто поставляются со всей инфраструктурой параллельных вычислений, как правило, с конкретной операционной системой, а иногда и вместе с компьютером. Поэтому уже на втором году исследований принято решение разрабатывать собственную систему управления кластером АСМС<sup>2</sup> для комплексов «Инпаркком», в задачи которой вошла интеграция компонентов инфраструктуры кластера и управление его ресурсами.

Сегодня имеем разнообразные среды параллельного программирования и соответствующие инструменты, а также готовые коммерческие решения, как правило, в комплекте с оборудованием комплекса. При разработке кластерных комплексов «СКИТ» и «Инпаркком» использованы исключительно бесплатно-распространяемые средства, которые в этой отрасли развиваются весьма стремительно.

Сегодня Linux – наиболее популярная операционная система для комплексов кластерной архитектуры. Дистрибутивы Linux дешевы и часто бесплатны, Linux поддерживает большинство современных процессоров, и обладает хорошей базой инструментов для разработчика и параллельных вычислительных сред.

<sup>1</sup> API операционной системы предоставляет средства создания новых потоков и семафоры для синхронизации. Для компьютеров с распределенной памятью необходимы программы на уровне сетевых протоколов.

<sup>2</sup> Аббр. от Advanced Cluster Management System

Независимо от выбранных компонентов, эффективность параллельного комплекса зависит главным образом от того, насколько хорошо удалось задействовать оборудование и программное обеспечение для решения конкретной задачи. При построении параллельного комплекса существует множество компонентов – операционные системы, компиляторы, библиотеки, реализации MPI, файловые системы и пр. Одной из важнейших задач при построении параллельного комплекса – достичь максимальной эффективности при взаимодействии этих компонентов. Международный проект «Scalable Systems Software» исследует взаимодействие различных программных компонентов для разработки единого эффективного интерфейса в программном обеспечении параллельных компьютеров (<http://www.scidac.gov/ScalableSystems/>).

Системы управления кластером (Cluster Management Systems) исполняют интегрирующую роль в архитектуре кластера, соединяя его ресурсы и предоставляя пользователю доступ к ним. Часто сервис-ориентированная система управления кластером представляет собой набор отдельных служб (компонентов системы), взаимодействующих по открытым интерфейсам. Это позволяет построить управление комплексом с учетом особенностей его архитектуры и аппаратной реализации, а многие компоненты доступны бесплатно. Коммерческие кластерные комплексы включают полный набор программного обеспечения по управлению кластером: Windows HPC Server, IBM Cluster Systems Management, HP Cluster Management Utility (HP CMU), Moab Cluster Manager, Bright Cluster Manager. Из бесплатных реализаций наиболее известны: OpenMosix, HeartBeat, IPVS, Keepalived, Nimbus, Piranha, Ultra Monkey [5], но по функциональности они значительно уступают коммерческим.

Основные компоненты системы управления кластером включают подсистемы для управления ресурсами, планирования выполнения задач, мониторинга системы, обеспечения безопасности, журнализации, пользовательского интерфейса управления кластером.

### Роль системы управления кластером ACMS на комплексах «Инпарком»

Комплексы семейства «Инпарком» состоят из хост-компьютера<sup>1</sup>, дискового хранилища, и узлов, выполняющих вычисления по требованию. Ресурсы на нижнем уровне управляются системным программным обеспечением и инфраструктурой параллельных вычислений и грид. Роль системы управления кластером состоит в координации служб комплекса и предоставлении к нему пользовательских интерфейсов. На рис.1. показана архитектура системного программного обеспечения комплексов семейства «Инпарком» как взаимодействующие уровни.

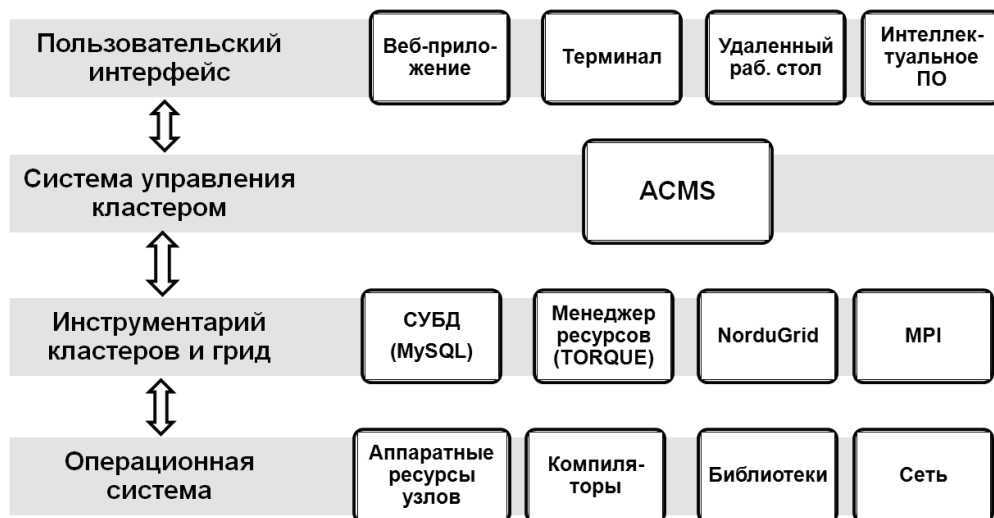


Рис. 1. Роль системы управления кластером на комплексах семейства «Инпарком»

Система управления кластером во многом опирается на менеджер ресурсов. Находясь на уровне middleware, он является надстройкой над операционной системой и координирует выполнение программ (не обязательно параллельных) на кластере. Как правило, менеджер ресурсов состоит из набора служб, распределенных на компьютерах кластера. На управляющем компьютере кластера обычно располагается главная служба, принимающая решение о выделении и возвращении ресурсов. На остальных узлах действуют «дочерние» службы, отвечающие за управление и мониторинг конкретного узла и за запуск на нем программ.

Менеджеры ресурсов активно развиваются OpenSource-сообществом. Среди множества менеджеров ресурсов, разработанных для конкретных кластеров в середине 1990-х годов выделим OpenPBS<sup>2</sup>, созданный MRJ для кластерных комплексов лаборатории NASA, решающих астрофизические задачи. Впоследствии компания Veridian, позже Altair Engineering получили права собственности на проект OpenPBS и их продукт

<sup>1</sup> Хост-компьютер на комплексах «Инпарком» предоставляет доступ к комплексу по сети и является управляющим компьютером для кластера.

<sup>2</sup> Open Portable Batch Scheduler, <http://www.openpbs.org>

PBS Pro стал коммерческим. Но лицензионные условия на OpenPBS позволили отдельным группам разработчиков взять исходные тексты OpenPBS и создать на их основе свои версии этого менеджера ресурсов. Так в Интернете появилось разнообразие «заплаток», позволяющих исправить некоторые известные ошибки и добавить новую полезную функциональность. Многие заплатки несовместимы между собой и в Интернете имелись инструкции о том, в каком порядке их нужно применять, чтобы получить «правильную» версию OpenPBS. В результате группа программистов из ClusterResources выпустила исправленную и улучшенную версию OpenPBS, названную TORQUE [6]. С тех пор TORQUE стал отдельным весьма популярным продуктом, особенно в грид-инфраструктурах.

Менеджер ресурсов SLURM, работающий на кластерных комплексах СКИТ, разработала лаборатория LLNL<sup>1</sup>. Он имеет много общего с TORQUE и завоевал популярность хорошей работой на большом числе узлов (сотни и тысячи), простым, но продуманным дизайном системы и служб, а также наличием API для написания надстроек (plugins). Используя их можно запрограммировать планировщик заданий, контроль безопасности, мониторинг и прочее.

Роль менеджера ресурсов не ограничивается управлением очередями заданий и выделением ресурсов. Менеджер ресурсов позволяет быстро запускать MPI-программы на узлах. Обычно реализации MPI используют удаленный безопасный терминал (SSH) для запуска параллельной программы на большом числе узлов. Эта процедура занимает длительное время, особенно на больших комплексах. Так, TORQUE и SLURM поддерживают запуск MPI-программ через сеть служб на узлах. Причем запросы на запуск распространяются между узлами иерархически. Кроме того, менеджер ресурсов контролирует завершение всех процессов по окончании их работы. А SLURM предоставляет адаптирующий интерфейс OpenPBS/TORQUE для адаптации в грид-инфраструктуру. Подробнее об истории и функциональности менеджеров ресурсов см. в RCE-Podcast [7].

### Архитектура системы управления кластером

Система управления кластером ACMS для комплексов семейства «Инпарк» состоит из взаимосвязанных служб, показанных на рис.2. Основные компоненты ACMS включают такие подсистемы: управление ресурсами, планирование выполнения задач, монитор системы, обеспечение безопасности, журнализация, пользовательский интерфейс управления кластером.

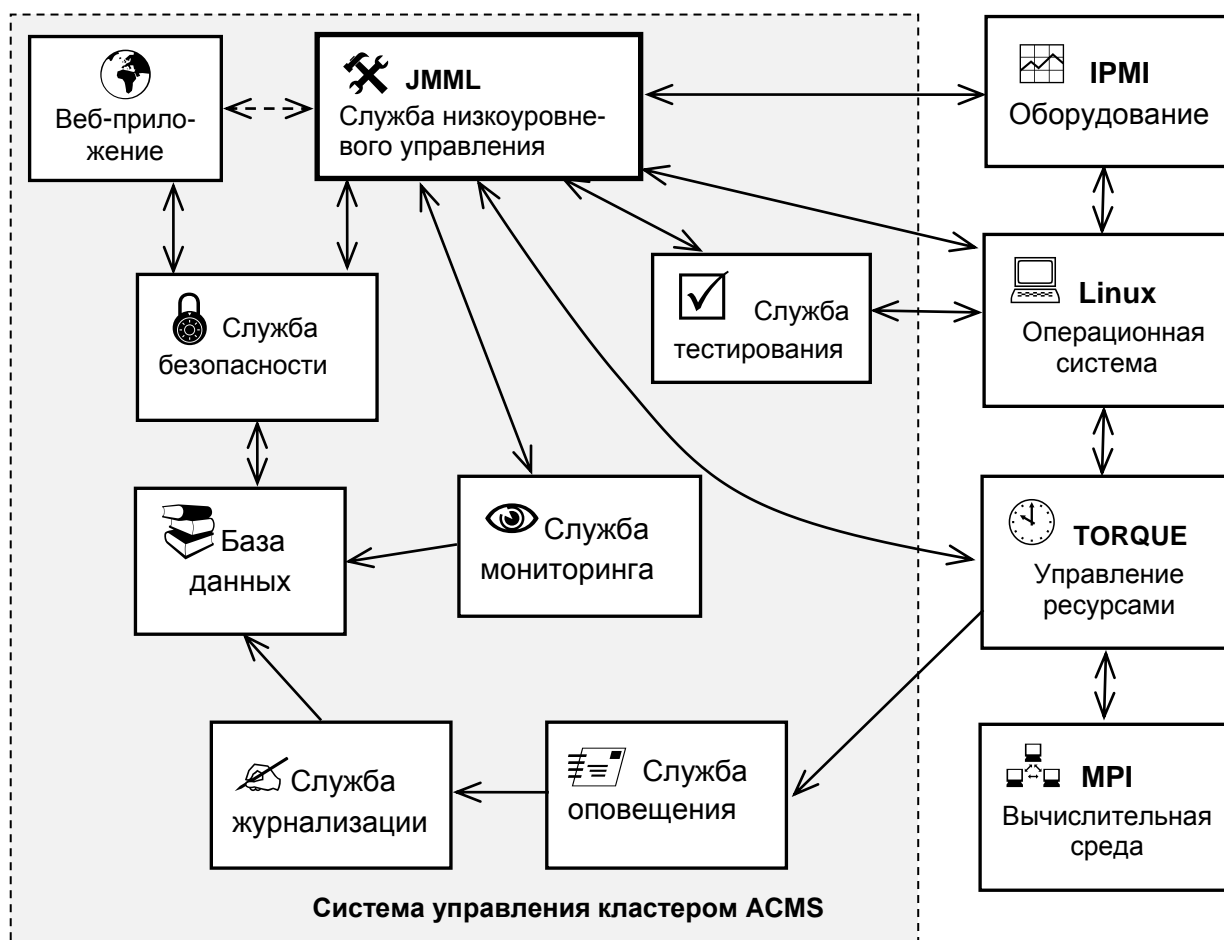


Рис. 2. Схема взаимодействия служб системы управления кластером ACMS

<sup>1</sup> Lawrence Livermore National Laboratory, <https://www.llnl.gov/>

Веб-приложение – основное средство доступа пользователей к управлению кластером. Служба безопасности контролирует доступ к данным и действиям в рамках установленных полномочий. Наибольшими полномочиями обладает системный администратор комплекса. Большая часть учетной информации находится в базе данных, большая часть действий осуществляется через службу низкоуровневого управления кластером.

Служба мониторинга – это демон или системная служба, которая выполняется на узлах кластера и на управляющем компьютере. На узлах эта служба измеряет параметры загруженности процессора и объем занятой и свободной памяти. Центральная служба мониторинга собирает эти данные с узлов и записывает информацию в базу данных. Служба тестирования позволяет проверить работоспособность узлов и включает тестирование памяти, сети и дисков.

Для поддержки одновременно нескольких операционных систем и менеджеров ресурсов разработана служба низкоуровневого управления кластером (JMML), предоставляющая единый интерфейс управления ресурсами. Веб-приложение в среде ACMS использует JMML для мониторинга и управления заданиями, оборудованием и пр. Вычислительная среда включает компиляторы, специальные библиотеки, среды обмена сообщениями.

## Управление заданиями

Со стороны пользователей работа с кластером через веб-приложение<sup>1</sup> осуществляется по такому сценарию: создание профиля задания, настройка параметров задания, передача задания на выполнение, ожидание уведомления о завершении, просмотр журнала и трассировка задания. Под *заданием* подразумевается совокупность приложения, данных, вычислительной среды, выделенных ресурсов. Задание можно рассматривать по-разному, в зависимости от его текущего состояния (запланировано, выполняется, выполнено, прервано), но свойства заданий не меняются, а только переходят из запланированного/желаемого состояния в фактическое. На рис. 3 показано окно управления заданиями в веб-приложении ACMS.

Под *профилем задания* подразумевается совокупность исходных текстов и данных, параметров компиляции и запуска, а также условия выполнения задания. Профиль задания – его шаблон, содержащий параметры, в основном неизменные от запуска к запуску. Профиль задания включает следующие параметры: имя профиля (при передаче на выполнение становится именем задания), каталог профиля (в котором располагаются исходные файлы и данные), способ задания исходных файлов и компиляции, язык программирования, временная квота, количество необходимых процессоров (либо список узлов, если требуются конкретные узлы), параметры командной строки.

ACMS Cluster Management System  
Управление заданиями  
Logged on as ru Ющенко P. A., used time:363:01:54, cpu time:309:16:33

Вы имеете права - Администратор

Ваши профили программ			Ваши задания кластеру							
№	Имя	Действия	№	Имя	Проц.	Время запуска	Время старта	Время окончания	Статус	Действия
1.	alla_zgoin	запустить редактировать удалить	1	skampi_mpich2	128	2008-05-12 11:12:50	2008-05-12 13:13:49	2008-05-12 13:46:14 (0:32:25)	прервано	трассировка журнал
2.	cri	запустить редактировать удалить	2	skampi_mpich2	128	2008-05-12 10:44:43	2008-05-12 12:46:42	2008-05-12 13:12:29 (0:25:47)	прервано	трассировка журнал
3.	dwm3d_center	запустить редактировать удалить	3	skampi_mpich2	128	2008-05-12 10:42:55	2008-05-12 12:43:53	2008-05-12 12:43:55 (0:0:2)	выполнено	трассировка журнал
4.	dwm3d_center_lam	запустить редактировать удалить	4	skampi_mpich2	128				прервано	трассировка журнал
5.	dwm3d_center_lam	запустить редактировать удалить	5	example 1	112				выполнено	трассировка журнал
7.	dwm3d_example1	запустить редактировать удалить	6	example 1	112	2008-05-06 16:37:15	2008-05-06 18:39:23	2008-05-06 18:39:36 (0:0:15)	выполнено	трассировка журнал
8.	dwm3d_old_example2	запустить редактировать удалить	7	example 1	112	2008-05-06 16:35:24	2008-05-06 18:37:33	2008-05-06 18:37:54 (0:0:21)	выполнено	трассировка журнал
10.	Eig	запустить редактировать удалить			112	2008-05-06 16:27:43	2008-05-06 18:29:52	2008-05-06 18:30:13 (0:0:21)	выполнено	трассировка журнал
11.	etiv	запустить редактировать удалить			112	2008-05-06 16:17:25	2008-05-06 18:19:43	2008-05-06 18:20:05 (0:0:22)	выполнено	трассировка журнал

Рис. 3. Окно управления профилями и заданиями в системе управления кластером ACMS

<sup>1</sup> Запускать параллельные MPI-программы на комплексах семейства «Инпарком» можно и без веб-приложения, используя безопасный удаленный терминал (например, PuTTY), указывая команды компиляции и запуска с консоли. При этом регистрация задания и выделение узлов проходит прозрачно для пользователя. Если узлов для запуска программы не достаточно, программа в терминале будет ожидать в очереди.

Окно управления заданиями предоставляет пользователю фактически всю необходимую информацию по его заданиям: профили заданий, задания в очереди, выполняющиеся задания и выполненные задания. Выполняющиеся задания можно прервать, не дожидаясь их завершения. Поскольку все программы в задании выполняются в пакетном режиме, все данные, которые она выводила в консоль, используя стандартные средства вывода (stdout/stderr), попадают в так называемый журнал задания, который можно посмотреть при выполнении задания и после завершения.

Когда пользователь отправляет задание на выполнение через веб-интерфейс ACMS, параметры задания передаются промежуточному слою JMML, который компилирует (при необходимости) программу задания, готовит пакетный скрипт и передает его менеджеру ресурсов. Пакетный скрипт инициализирует вычислительную среду, запускает исполняемый файл на выделенных планировщиком узлах, освобождает ресурсы после выполнения задания. Если на стадии компиляции произошла ошибка, задание не передается на выполнение. Возникшие при компиляции ошибки доступны через интерфейс пользователя ACMS.

Из веб-приложения пользователям доступен мониторинг состояния узлов. Узел может быть «доступен», «не доступен», «занят» и «на техническом обслуживании». Кроме того, можно получить информацию о том, кто и на каких узлах выполняет задание, сколько используется при этом ресурсов (рис. 4).

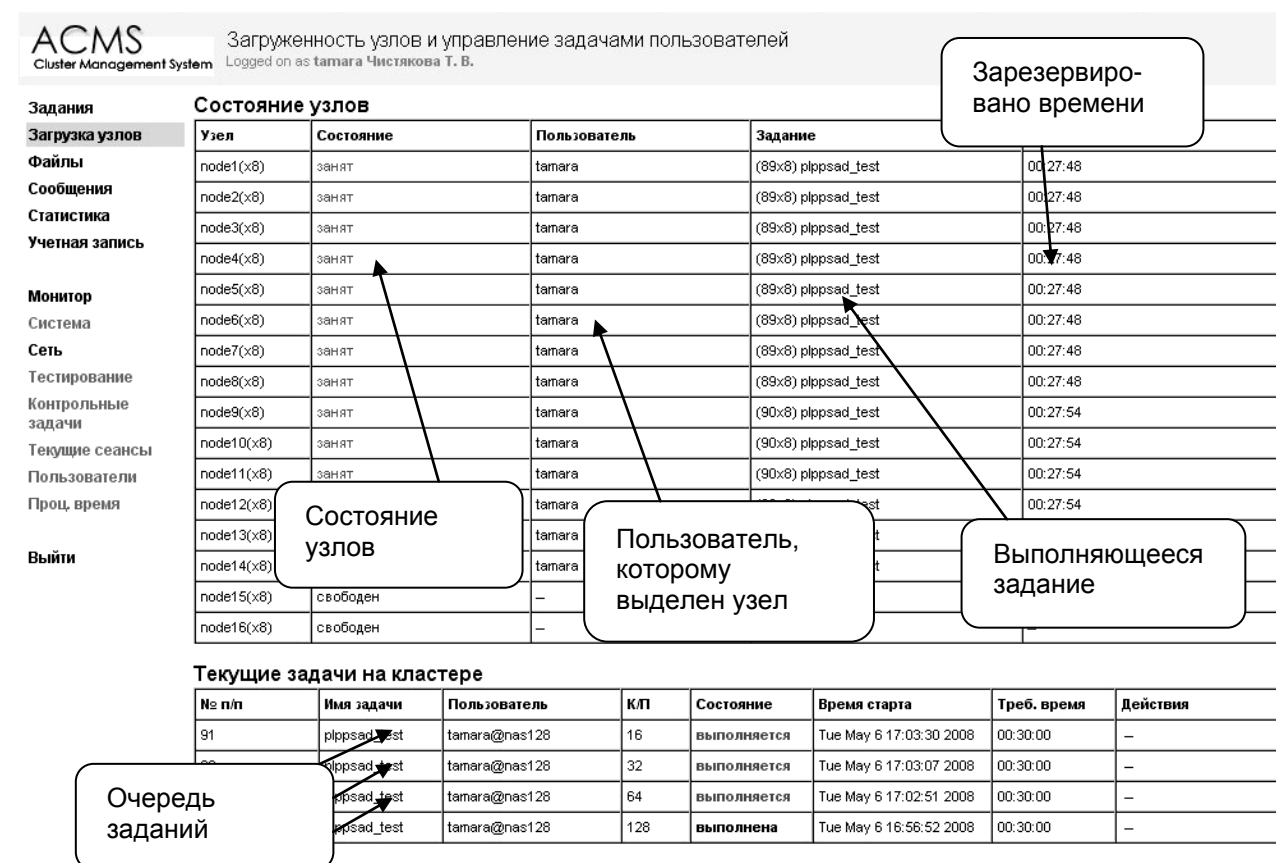


Рис. 4. Обзор выполняемых заданий

Управление очередью и алгоритм выбора задач на выполнение определяет менеджер ресурсов. Обычно используется порядок FCFS с некоторыми модификациями, например, «backfill». На комплексах семейства «Инпарк» используется порядок обслуживания заданий FCFS.

### Мониторинг загруженности комплекса

Служба мониторинга на каждом узле периодически проверяет статистические данные, собираемые операционной системой (например, /proc/stat, /proc/meminfo). Периодически служба мониторинга, находящаяся на управляющем компьютере, опрашивает узлы для определения их состояния и данных о загруженности ресурсов (процессора и оперативной памяти). Далее эти данные сохраняются в базе данных, их можно посмотреть в виде графика из веб-приложения ACMS (см. рис. 5).

Кроме того, для каждого задания в базу данных записывается информация о состоянии и загруженности узлов во времени. Поскольку все выполняющиеся на комплексе задания связаны с учетными записями пользователей, можно получить хорошую картину активности параллельных программ пользователей. Это удобно для анализа эффективности программ постфактум. Например, если при выполнении расчетов процессор загружен не полностью, возможны простои программы при ожидании сообщений либо доступе к устройствам ввода/вывода. На рис. 6. показан пример графиков по двум узлам.

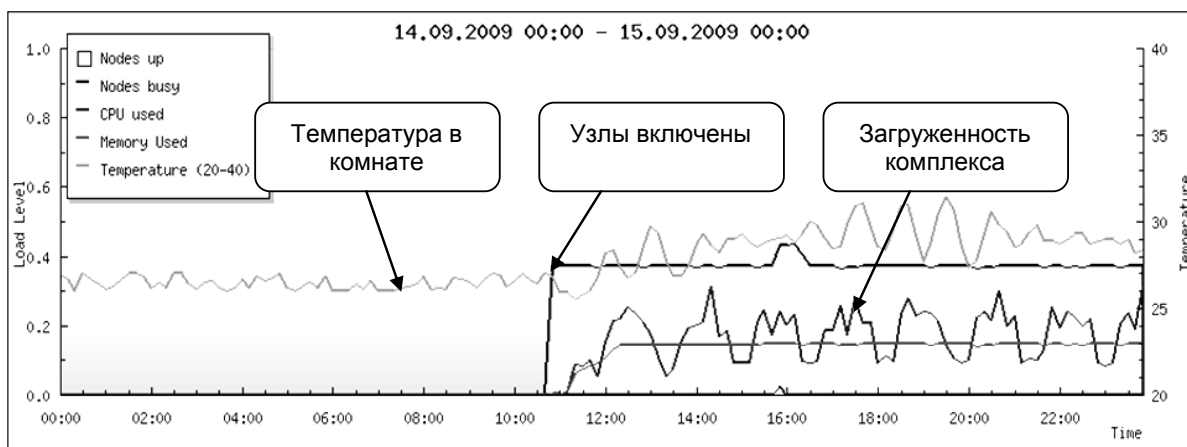


Рис. 5. График загрузки кластера

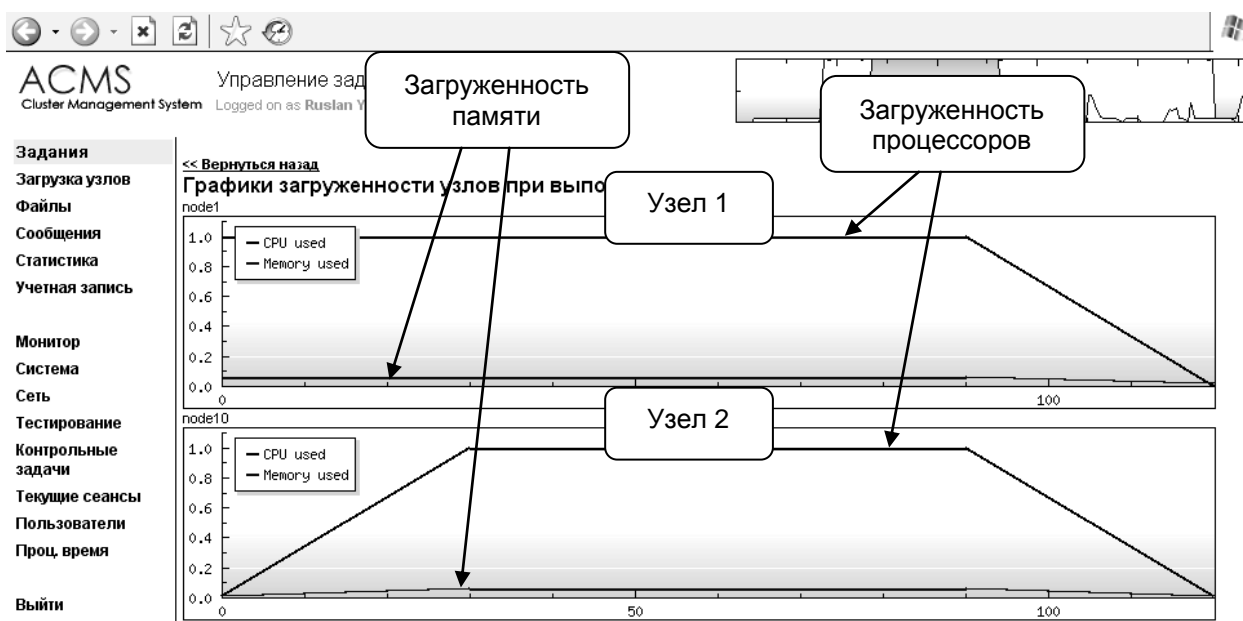


Рис. 6. График загрузки узлов при выполнении параллельной программы

### Управление оборудованием

В систему ACMS встроены модули для контроля состояния оборудования, а также тестирования и управления узлами кластера. Системному администратору комплекса доступна панель управления оборудованием (рис.7). Он может видеть состояние узлов, загрузку процессора, объем свободной памяти, температуру процессоров, скорости вращения вентиляторов в системе охлаждения, а также значение критических напряжений.

Управлять оборудованием можно на уровне операционной системы, а также на более низком уровне интерфейса IPMI, чтобы получить доступ к аппаратным функциям узлов кластера по сети. IPMI позволяет удаленно включать, выключать, сбрасывать компьютер, интерактивно управлять компьютером через текстовую и графическую KVM-консоль. В этом режиме по сети передается изображение экрана на клиентский компьютер, а обратно – коды нажатых клавиш и движения мыши.

Интеллектуальный интерфейс управления оборудованием (IPMI) – это спецификация, состоящая из наборов интерфейсов, используя которые программные службы могут получить доступ к аппаратным функциям компьютера. IPMI функционирует независимо от операционной системы и позволяет управлять компьютером даже тогда, когда операционная система не загружена, например при выключенном компьютере.

Спецификация IPMI определяет только интерфейсы, а реализация отличается у разных разработчиков. На комплексах «Инпарком» IPMI – это дополнительная плата, подключаемая в специальный слот в компьютере. Эта плата получает питание даже тогда, когда компьютер выключен. С внешней стороны такой платы доступен порт сети Ethernet, через который осуществляется удаленное управление.

Интерфейс IPMI предусматривает генерацию сигналов предупреждения в момент возникновения проблемных для оборудования ситуаций, например при перегреве. Просмотреть все сигналы предупреждения можно через веб-интерфейс в меню «Система» в разделе «Предупреждения».

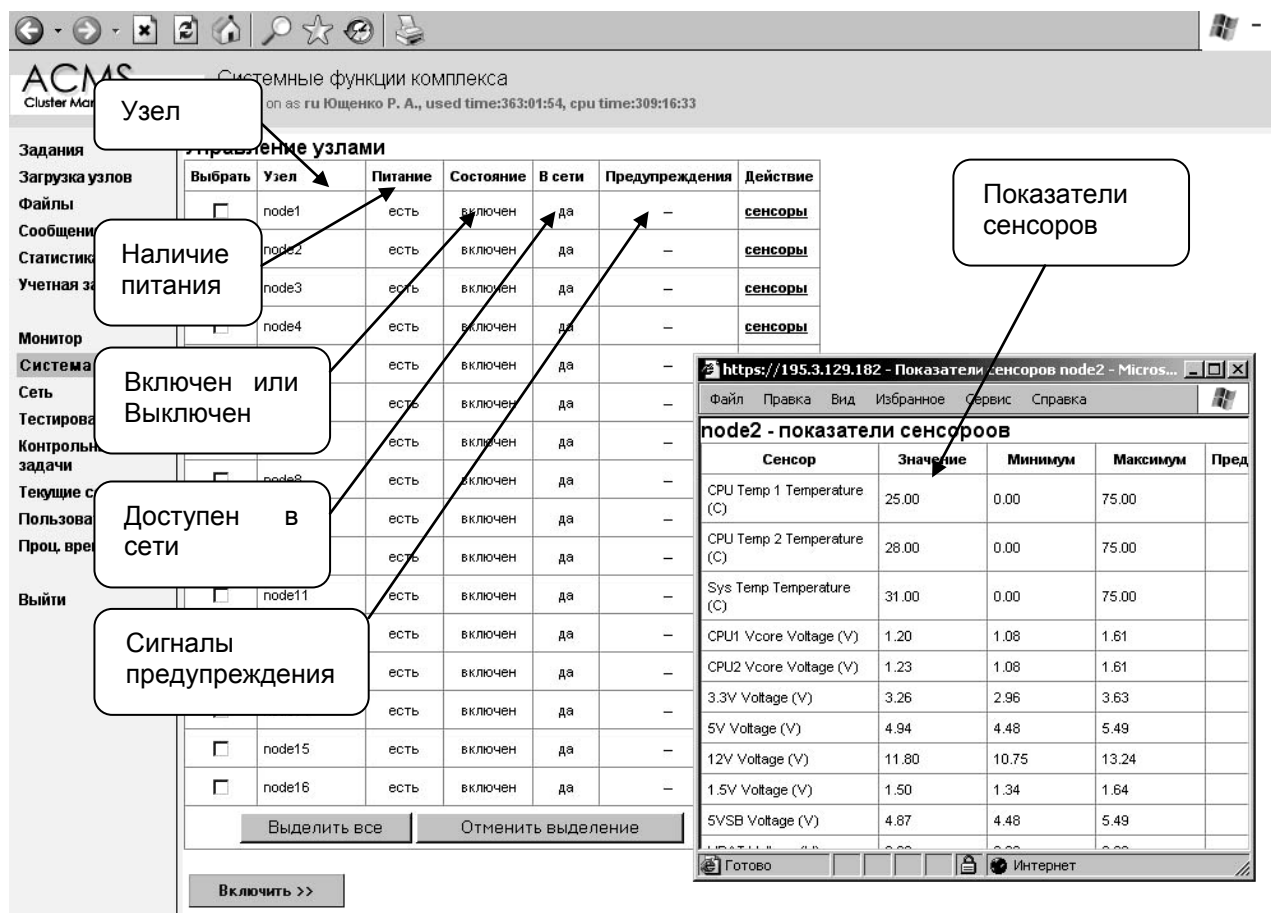


Рис. 7. Мониторинг состояния и управление оборудованием

### Автоматическое включение и выключение узлов

Поскольку, интерфейс IPMI позволяет включать и выключать питание узлов программно, система управления кластером может контролировать этот процесс без вмешательства человека. Выключая узлы, которые простаивают без полезной работы, экономится электроэнергия, потребляемая этими узлами и кондиционеров, охлаждающих помещение. Кроме того, автоматически выключать узлы можно в аварийных случаях, например при отказе систем охлаждения.

Система управления кластером ACMS позволяет настраивать политику автоматического включения и выключения узлов. Приведем правила, действующие на комплексе «Инпарком-256».

1. Выключить все компьютеры, если температура в комнате превышает 36°C.
2. Принудительно завершить выполняющиеся задания, если температура в комнате превышает 32°C<sup>1</sup>.
3. Если температура в комнате превышает 28°C, то новые задания можно принимать в очередь, но нельзя отправлять на выполнение.
4. Если комнатная температура ниже 28°C, не все узлы включены и в очереди есть задания, для выполнения которых недостаточно свободных узлов, необходимо включить нужное число узлов. При этом счетчики простоя для свободных узлов нужно обнулить.
5. Выключить узел, если он неактивен в течение 30 минут.

#### Примечание.

- Правила указаны в порядке их приоритета. В случае противоречия двух и более правил, выполняется верхнее по списку.
- Узел считается неактивным некоторое время, если на нем не выполняются зарегистрированные в менеджере ресурсов задания и загрузка процессора не превышает 20 %.
- Между включением/выключением узлов и последующим анализом делается перерыв в 5 минут, чтобы вступило в силу изменение состояние всей системы.

Пример графиков загруженности комплекса при автоматическом включении и выключении всех узлов показан на рис. 8.

<sup>1</sup> При интенсивных вычислениях потребляемая электроэнергия (и соответственно теплота) возрастает на 50 – 100 %. Если система охлаждения не справляется с задачей, принудительного завершения часто достаточно для нормализации работы комплекса.

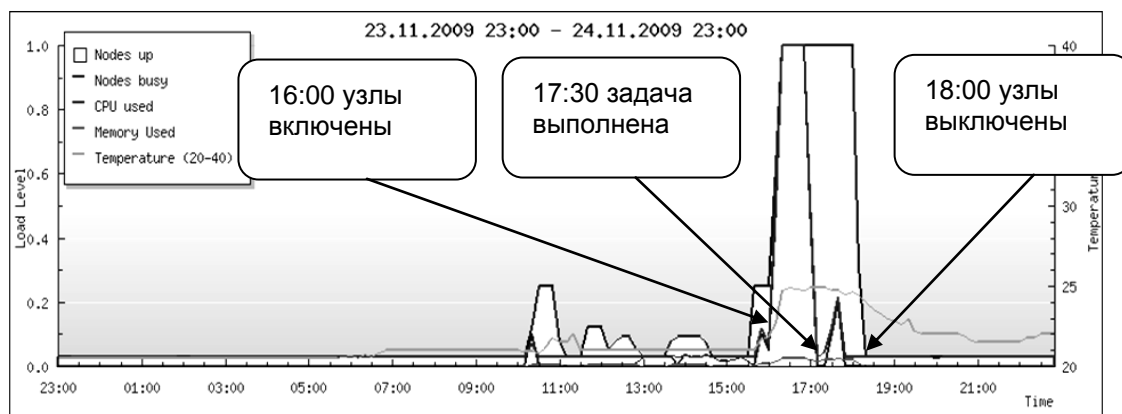


Рис. 8. Графики загрузки при автоматическом включении и выключении

### Прогнозирование эффективности

Данные о выполненных на кластере заданиях можно использовать для прогнозирования времени выполнения этих заданий для другого количества процессоров. Для этого используется метрика К.Флатта (МКФ), которая позволяет экспериментально оценить долю накладных затрат и последовательной части программы<sup>1</sup> для разного количества процессоров ( $N$ ) [8]. При идеальном параллелизме эта составляющая равна нулю, а при отсутствии накладных затрат кривая МКФ параллельна оси X. Однако на практике кривая МКФ возрастает<sup>2</sup> с увеличением  $N$ . ACMS позволяет анализировать времена выполнения параллельных программ прямо в пользовательском интерфейсе. Минимальные данные, необходимые для получения прогноза, – это время выполнения при  $N=1$  и хотя бы в двух точках при  $N>1$ . Для оценки используется линейная аппроксимация МКФ. На рис. 9 показан пример прогнозирования прикладной программы, выполняемой на комплексе Инпарк-256.

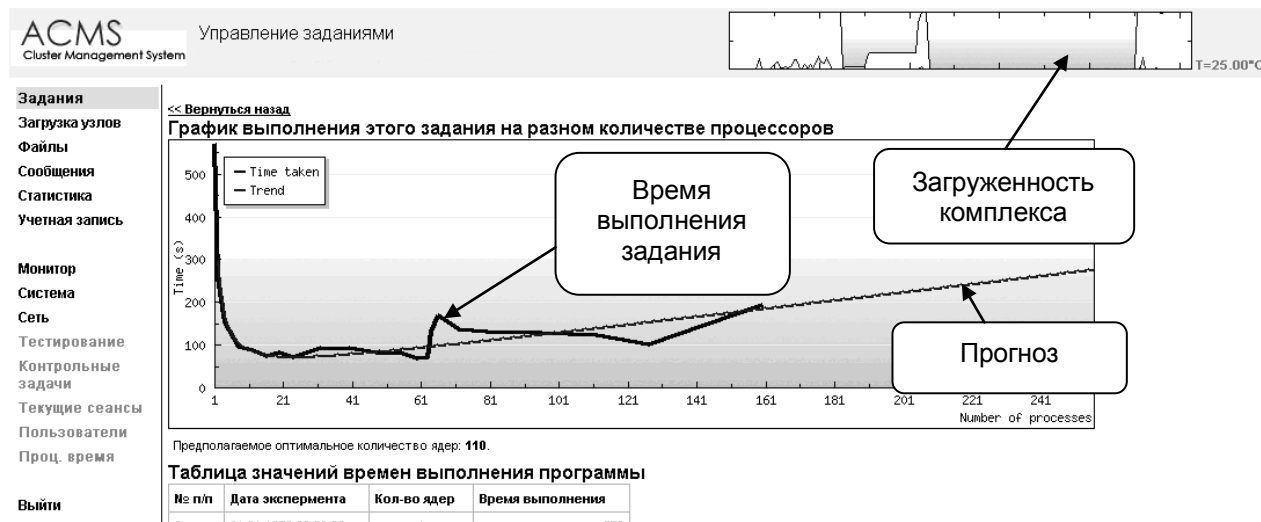


Рис. 9. Прогнозирование времени расчета на комплексе «Инпарк-256»

Имея прогнозные и экспериментальные данные о времени выполнения программы при разном  $N$ , легко построить графики оценки накладных расходов (МКФ), ускорения и эффективности.

Ускорение определяется как  $S_N = \frac{T_1}{T_N}$ , а эффективность  $E_N = \frac{S_N}{N} = \frac{T_1}{NT_N}$ , где  $T_1$  – время выполнения

на одном ядре,  $T_N$  – время выполнения на  $N$  ядрах.

На рис. 10. показан пример таких графиков. Подробнее о методах исследования эффективности параллельных программ см. в работах [9–11].

<sup>1</sup> В литературе *накладными* (или *коммуникационными*) *затратами* при фиксированном размере задачи называют долю времени, которая увеличивается с ростом  $N$ , а *последовательной частью программы* – долю времени, которая с ростом  $N$  остается постоянной.

<sup>2</sup> Иногда кривая МКФ может убывать, причем Карп и Флатт объясняли это наличием сверхлинейного ускорения или кэш-эффектов. На самом деле кривая МКФ может убывать, если с ростом  $N$  накладные затраты растут логарифмически, например, в параллельных алгоритмах быстрого преобразования Фурье [9].



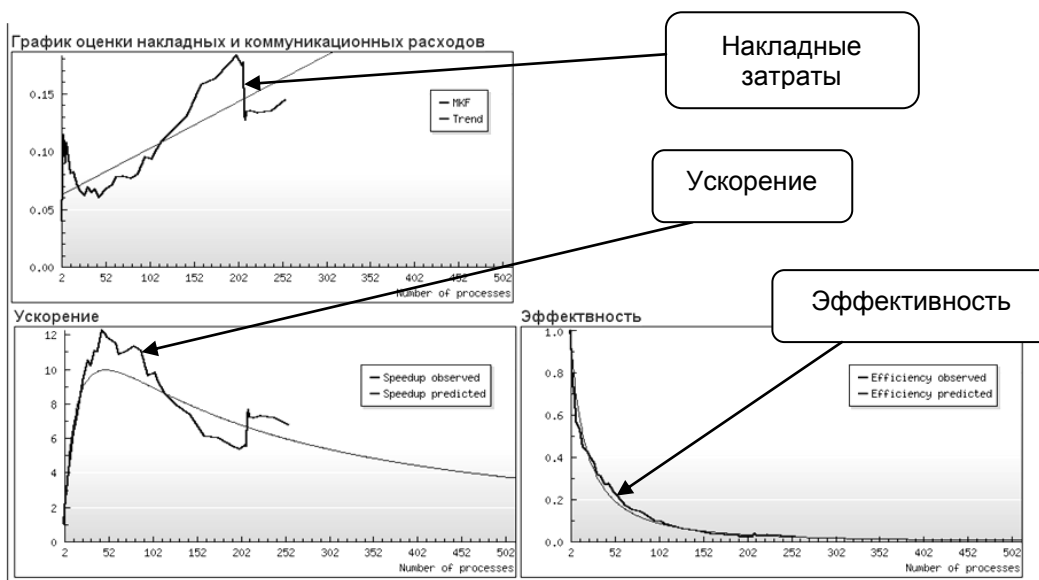


Рис. 10. Графики накладных расходов, ускорения и эффективности

## Заключение

Повышение интеллектуальности параллельных вычислительных систем и инструментов разработки – приоритетное направление компьютерной индустрии, поскольку параллелизм – единственный путь наращивания производительности компьютеров. Обычные компьютеры стали многоядерными, а суперкомпьютеры все чаще создаются в рамках смешанной архитектуры, в которой кластерный комплекс состоит из многоядерных процессоров и компьютеров с графическими процессорами (GPU). Но сегодня операционные системы не предоставляют достаточных средств для управления кластерным комплексом как единым целым. Разработанная система управления кластером ACMS выполняет интегрирующую функцию, координируя компоненты комплекса, разработанные независимыми группами. За счет инкапсуляции системной и аппаратной специфики в службу низкоуровневого доступа удалось добиться хорошей мобильности ПО и интероперабельности с ПО других разработчиков. (Система управления кластеров ACMS успешно работает на разных комплексах семейства, имеющих различную архитектуру).

1. Heroux M., Raghavan P., Simson H. Parallel Processing for Scientific Computing. – Philadelphia: SIAM, 2006. – 396 p.
2. <http://www.beowulf.org/>
3. Численное программное обеспечение MIMD-компьютера Инпарк. / А.Н. Химич, И.Н. Молчанов В.И. Мова и др. – Киев: Наук. думка, 2007. – 221 с.
4. Молчанов И.Н., Перевозчикова О.Л., Химич А.Н. Опыт разработки семейства кластерных комплексов Инпарк // Кибернетика и системный анализ. – 2009.– № 6. – С. 88–96.
5. <http://www.linux.com/archive/feature/57073>
6. <http://www.clusterresources.com/products/torque-resource-manager.php>
7. <http://www.rce-cast.com>
8. Karp A. H., Flatt H. P. Measuring Parallel Processor Performance // Communication of the ACM. – 33, N. 5. – 1990. – P. 412–422.
9. Kumar V., Gupta A. Analyzing Scalability of Parallel Algorithms and Architectures // J. of Parallel and Distributed Computing. – 1994. – 22. – P. 379–391.
10. Перевозчикова О.Л., Тульчинский В.Г., Ющенко Р.А. Построение и оптимизация параллельных компьютеров для обработки больших объемов данных // Кибернетика и системный анализ. – 2006. – № 4. – С. 117–129.
11. Ющенко Р.А. Измерение производительности параллельных компьютеров с распределенной памятью // Кибернетика и системный анализ. – 2009.– № 6. – С. 106–117.