

## Комплексный подход к обслуживанию вычислительных кластеров

С. А. Жуматий\*

*В статье приводится описание программного комплекса ParCop, предназначенного для помощи пользователям и администраторам в эффективном использовании вычислительных кластеров. Приведены примеры часто возникающих задач, успешно решаемых с помощью ParCop.*

### 1. Введение

Вычислительные кластеры являются мощным инструментом для проведения сложных расчётов, использование которого сопряжено с рядом трудностей как для администраторов, так и для самих пользователей. Взять хотя бы поддержку работоспособности и мониторинг текущего состояния большого количества постоянно включенного оборудования — крайне непростая задача. В данной работе мы обсудим, что может быть необходимо для обеспечения эффективной работы и пользователей, и их задач на кластере.

Пользовательская программа может обладать специфическими свойствами, которые значительно снизят её производительность на кластере, причем компилятор, библиотеки вряд ли помогут исправить ситуацию. Что является причиной замедления — несоответствие размера оперативной памяти узла данным задачи, необходимость постоянно пользоваться медленными сетевыми дисками, несогласованность работы параллельных процессов или что-то ещё? Умение найти ответы на подобные вопросы может значительно облегчить работу пользователя.

---

\* Научно-исследовательский вычислительный центр МГУ

Администратору кластера такая информация тоже будет очень полезна, так как он должен иметь информацию о характере использования кластера. В детали работы каждой задачи ему вникать, может, и не всегда надо, но иметь сводную статистику по задачам, пользователям и использованию ресурсов кластера просто необходимо.

Существуют различные средства, позволяющие помочь в решении перечисленных выше вопросов. К примеру, система Ganglia позволяет собирать данные о работе процессоров, использовании памяти и другие характеристики работы вычислительных узлов. Однако, привязать эти данные к задачам пользователей или получить комплексные характеристики о работе кластера невозможно. Другие программные комплексы, насколько нам известно, также решают лишь очень узкие задачи и не способны служить серьёзным подспорьем ни пользователю, ни администратору кластера.

## 2. Постановка задачи

Требуется создать программную среду для поддержки выполнения параллельных приложений на кластерных установках, позволяющую:

- управлять прохождением задач,
- автоматически распределять свободные процессоры между задачами,
- гибко управлять политикой использования кластера различными пользователями,
- предоставлять детальную информацию о работе задач и пользователей на кластере,
- предоставлять детальную статистику использования ресурсов кластера задачей как во время работы задачи, так и после её завершения,
- предоставлять суммарные отчёты о работе задач, пользователей, использовании ресурсов кластера,

- оказывать минимальное влияние на работу пользовательских задач.

### 3. Архитектура предложенного решения

Для решения поставленной задачи был предложен подход, предусматривающий создание двух программных компонентов, способных работать как независимо друг от друга, так и в режиме взаимодействия. Оба компонента в комплексе представляют собой интегрированную среду, получившую название ParCon.

Первый компонент — это система управления заданиями Cleo. Она была разработана в НИВЦ МГУ специально для управления использованием ресурсов вычислительных кластеров. Система ориентирована на работу с параллельными приложениями и поддерживает многие параллельные среды. Под её управлением в течении 5 лет успешно работали 4 разных кластера суперкомпьютерного комплекса НИВЦ МГУ, объединявшие в общей сложности 153 вычислительных узла (306 процессоров).

Ключевые особенности Cleo:

- изначальная ориентированность на вычислительные кластеры,
- гибкость политик управления ресурсами,
- поддержка большого числа параллельных сред,
- расширяемость за счёт дополнительных планировщиков и модулей.

Для работы с Cleo пользователю практически не нужно дополнительной подготовки, поскольку программа `mpirun` автоматически ставит задачу в очередь. Для просмотра очереди можно использовать web-интерфейс или консольные команды.

Описание системы Cleo доступно по адресу в интернете:  
<http://parcon.parallel.ru/cleo.html>.

Второй компонент среды ParCon — это комплекс мониторинга AntMon. Именно он позволяет собирать детальную информацию о

состоянии вычислительных узлов и передать эту информацию администратору. AntMon изначально разрабатывался таким образом, чтобы предоставить возможность сбора статистики с большого числа узлов кластера с высокой степенью детализации и низкой нагрузкой на узлы. Этот программный пакет спроектирован по модульному принципу: в любой момент к нему можно добавить модуль для сбора той информации, которая существенна для данного кластера.

Сам AntMon способен не только собирать данные, но и определять критические ситуации на вычислительных узлах (если собранная информация позволяет это сделать), оповещать об этом администратора и/или выполнять набор экстренных команд. Однако для того, чтобы дать возможность провести содержательный анализ работы кластера или вычислительных программ, требуется иное средство.

Таким средством является аналитический модуль ParCon, который сохраняет данные о задачах и полученную с узлов информацию в базе данных, предоставляя возможность для исследования динамических характеристик задач и узлов. Например, с его помощью можно получить ответ на такие вопросы как «какова была средняя загрузка процессора во время работы задачи?» или «насколько согласована работа вычислительных узлов на данной задаче?».

Описание комплекса ParCon представлено на странице в интернете: <http://parcon.parallel.ru>.

#### **4. Система управления заданиями**

Сleo предоставляет пользователям прозрачную схему запуска программ, для чего используется привычный скрипт `mpirun`. В поставку системы входят программы для просмотра состояния очереди, удаления задач и изменения приоритета задач. Разработан web-интерфейс, позволяющий просматривать состояние узлов кластера, задач и очередей.

Администратору кластера предоставлены возможности гибкого управления политикой запуска программ пользователей. Среди таких возможностей стоит отметить:

- выделение процессорного времени для особых задач,
- временная блокировка указанных задач,
- ограничение количества одновременно занятых пользователем процессоров,
- ограничение количества запущенных пользователем задач,
- ограничение количества задач пользователя в очереди,
- управление приоритетами пользователей.

Система автоматически отслеживает состояние вычислительных узлов кластера и блокирует их в случае неполадок. Состояние задач на узлах также контролируется, и в случае некорректного завершения задания его процессы на узлах принудительно завершаются.

В систему встроена возможность автоматического определения работоспособности узлов, и в случае остановки работы вычислительного узла система автоматически блокирует его и снимает все задачи, работавшие на нём.

Администратор может в любое время блокировать и разблокировать задачи и вычислительные узлы. Реализована возможность «отложенной» блокировки узлов, когда блокировка вступает в силу только после окончания работы задач, распределённых на эти узлы.

В системе поддерживается иерархическая организация очередей, что позволяет создавать несколько разделов в рамках одного кластера, сохраняя возможность использовать его полностью, или, наоборот, объединять несколько кластеров под единой системой управления.

Cleo спроектирована расширяемой за счёт дополнительных планировщиков и внешних модулей, выполняющихся по заданным условиям. В случае, если модуль или планировщик содержит ошибку и вызывает сбой в работе, то Cleo просто блокирует его и продолжает работать без данного модуля или со встроенным стандартным планировщиком.

Немаловажной особенностью Cleo является поддержка иерархии очередей в рамках одного кластера. Она позволяет использовать

часть процессоров для выделенных задач (например, для отладки). При этом остаётся возможность использования всех процессоров, как выделенных, так и остальных, одновременно для одной задачи. Для такого использования не требуется переконфигурация или приостановка системы очередей.

## 5. Система мониторинга

Система AntMon была разработана для сбора данных с большого числа узлов и реагирования на аномальные значения полученных данных. Она спроектирована таким образом, чтобы сбор необходимых сведений проводился с минимальной нагрузкой на сеть и процессоры, а сбой головного сервера не приводил бы к отказу всей системы.

Система имеет распределённую структуру. На узлах, где необходимо собирать данные мониторинга, работают агенты, а анализ собранной информации выполняют головные серверы.

Информация для мониторинга собирается агентами при помощи подключаемых модулей. Модули запускаются один раз при получении первого запроса на информацию от них. После запуска модули остаются в рабочем состоянии и активируются при очередном запросе. Сами запросы обслуживаются агентами системы, они же отвечают за запуск модулей. На каждом узле, где необходимо собирать информацию, запускается один агент.

Модульный принцип построения AntMon позволяет отслеживать не только те параметры, которые предусмотрены в системе изначально, но и любые другие параметры необходимые пользователю, для чего достаточно написать модуль, который будет получать эти данные и передавать в систему. Интерфейс системы с модулями прост и полностью документирован. Модуль может быть написан практически на любом языке программирования и для его написания не требуется высокой квалификации программиста.

Для повышения надежности системы в архитектуре AntMon предусмотрено использование нескольких головных серверов. В штатном режиме они работают независимо друг от друга и могут распределять между собой общую нагрузку: каждый сервер будет

обслуживать свой набор агентов или отдельных параметров составных частей кластера. Для каждого параметра указывается список головных серверов, которые могут его обслуживать.

Сначала параметр обслуживается первым в списке сервером. Если связь с этим сервером обрывается, то параметр переходит под опеку следующего в списке сервера. В случае сбоя одного из головных серверов, остальные серверы перераспределяют ответственность за приписанные ему параметры и производят реагирование на сбой, запуская последовательность предопределенных команд. Когда сбойный головной сервер возвращается в рабочее состояние, остальные серверы повторно перераспределяют ответственность за параметры, а также выполняют соответствующий набор команд, например уведомление администратору вычислительной системы.

## 6. Аналитический модуль

Модуль выполняет задачу объединения данных от Cleo и AntMon в единую базу данных и предоставления аналитических данных пользователю и администратору. С его помощью осуществляется анализ собранных данных с привязкой к конкретным задачам на кластере. Сохранение информации в базе данных позволяет проводить анализ различной степени сложности. Например, узнать средний процент загрузки процессора на задачах определённого пользователя за последние несколько месяцев.

В качестве примера приведём небольшое исследование работы параллельных задач на кластере Ant НИВЦ МГУ. На рис. 1, 2, 3 и 4 представлен пример анализа времени счёта задач за месяц. По вертикали отложено число задач, по горизонтали — время счёта. Видно, что большое число задач завершается, не отработав даже секунды. Это говорит о том, что проводилось большое число запусков неотлаженных программ. С помощью аналогичного анализа с разбивкой по пользователям удалось быстро выявить пользователя, у которого регулярно возникали такие проблемы.

Яркие «пики» в 30 и 6000 минут соответствуют лимитам времени работы задач в очередях. «Пик» в 4300 минут соответствует лимиту времени, установленному одним из пользователей кластера.



Рис. 1. Распределение времён счёта задач менее 5 минут



Рис. 2. Распределение времён счёта задач менее часа

Другой интересный пример анализа конкретной задачи можно увидеть на рис 5. Здесь приведены графики минимальной, максимальной и усреднённой доли простоя всех процессоров, занятых задачами. Видно, что в начальный момент времени часть процессоров простаивает на 70%. Через какое-то время доля простоя процессоров сводится к нулю.





Рис. 3. Распределение времён счёта задач до двух дней



Рис. 4. Анализ времени счёта задач

Если сопоставить этот график с графиком свободной памяти (рис. 6), то становится видна их корреляция. Обусловлена она тем, что на некоторых узлах под данные задачи не хватило памяти и начался процесс вытеснения страниц задачи на диск. В момент, когда задаче выделяется достаточное место в памяти и происходит «обнуление» доли простоя процессора.

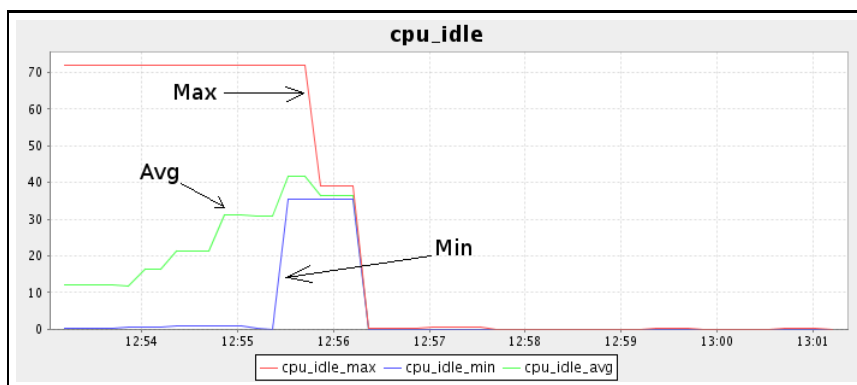


Рис. 5. Доля простоя процессора

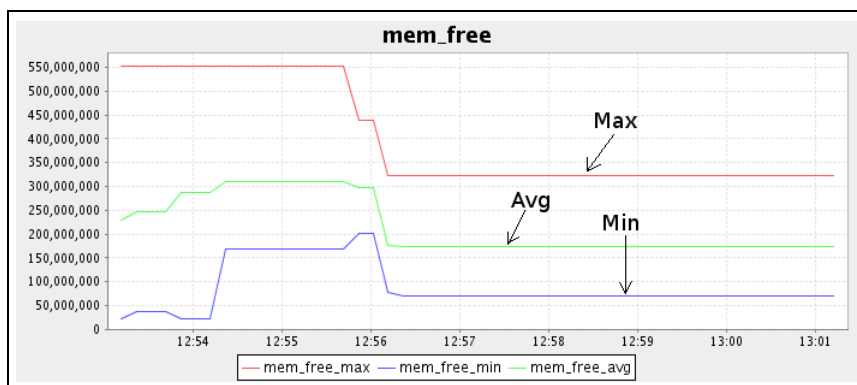


Рис. 6. Уровень загрузки памяти

## 7. Заключение

Обслуживание вычислительных кластеров — это большая и комплексная задача. Исключительное значение в её решении имеет информация о кластере, пользователях и их задачах и инструментальные средства для выполнения типовых действий. Чем больше администратор кластера знает о работе установки, тем продуктивнее бу-

дет её работа. Чем полнее представление пользователя о характере работы его программы на кластере, тем больше у него возможностей повысить эффективность программы.

ParCon способен предоставить пользователю и администратору средства для получения информации о работе как кластера в целом, так и отдельных задачах, пользователях, использовании ресурсов. ParCon продолжает развиваться. В первоочередные планы его развития входит исследование масштабируемости и расширение аналитических возможностей.

### Список литературы

- [1] Жуматий С.А. Система ParCon — ассистент в работе пользователя и администратора вычислительного кластера // Методы и средства обработки информации. Труды Всероссийской научной конференции. 2005. Изд-во отд. факультета вычислительной математики и кибернетики МГУ им. М.В.Ломоносова. С. 252-255.
- [2] Воеводин Вл.В., Жуматий С.А. Вычислительное дело и кластерные системы. — М.: Изд-во МГУ, 2007. — 150 с.
- [3] Жуматий С.А., Князев А.С. Исследование характеристик потока задач на вычислительном кластере // Научный сервис в сети Интернет: многоядерный мир. 15 лет РФФИ: Труды Всероссийской научной конференции. 2007. М.: Изд-во МГУ. С. 139.

