

**АГЕНТНАЯ ПАРАДИГМА КОМПЬЮТЕРНОЙ ИМИТАЦИИ В ЗАДАЧАХ
МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ GRID-ИНФРАСТРУКТУР****Е. О. Кринецкий, Ю. А. Шебеко (Москва)**

Непрерывное развитие науки и повышение сложности решаемых задач приводит к тому, что для своевременного решения сложных задач становится недостаточно стандартных вычислительных мощностей. В связи с этим большой интерес вызывает применение Grid-технологии[1]. Под Grid-инфраструктурой понимается программно-аппаратная среда, которая построена из вычислительных установок, пространственно распределенных в открытой сетевой среде, позволяющей дистанционно использовать любое количество ресурсов этих установок для выполнения широкого класса приложений. Под ресурсами понимаются процессоры, оперативная и постоянная память, программы и данные. Отметим наиболее значимые свойства Grid:

- Автономность ресурсов. Ресурсная база Grid формируется из независимых друг от друга вычислительных центров или отдельных компьютеров. Ресурсы обслуживаются и администрируются владельцами, которые имеют право проводить собственную политику доступа к ним;
- Гетерогенность ресурсов. Ресурсы, вовлеченные в Grid, имеют различные характеристики аппаратного и программного обеспечения, поэтому следует подбирать только те ресурсы, характеристики которых отвечают необходимым требованиям;
- Глобальная распределенность. Составляющие Grid вычислительные установки распределены в открытой глобальной сетевой среде, в которой велики коммуникационные издержки и предъявляются повышенные требования к безопасности;
- Коллективный режим работы. Ресурсы Grid используются в коллективном режиме, поэтому необходимо гибкое и скоординированное их распределение между пользователями, решающими разные задачи;
- Динамичность среды. Как состав ресурсов, так и состав пользователей может меняться, причем это является правилом, а не исключением.

Набор функций, которые требуются в Grid-системах от программного обеспечения – это обеспечение безопасности, надежности, мониторинга заданий и устройств, учета и протоколирования заданий. Ключевое место в этом ряду занимает функция диспетчеризации заданий. Эта функция обеспечивает распределение ресурсов из общего ресурсного пула Grid между заданиями, доставку программ и данных, реализуя одну из важнейших концепций Grid, а именно виртуализацию ресурсов. Этот процесс распределения называется задачей аллокации.

Служба диспетчеризации[2, 3] выполняет автоматическую обработку множества заданий, включая распределение заданий по доступным ресурсам – планирование, доставку необходимых входных файлов на ресурс, запуск, управление и мониторинг выполнения задания, доставку выходных файлов.

Независимо от того, как планируется расписание выполнения заданий, до запуска либо в реальном времени, для алгоритма планирования всегда можно выделить время планирования процессов на многопроцессорной системе. Это время определяется моментом между получением планировщиком информации об освобождении процессора и моментом непосредственного назначения работы на исполнение. Важной задачей является минимизация этого времени. Обычно расписание, близкое к оптимальному, требует большого времени планирования, а быстро сделанное расписание будет далеко от оптимального. Поэтому приходится искать компромисс.

Самым простым методом разделения пространства ресурсов между заданиями является метод FCFS [4]. Механизм работы данного алгоритма заключается в том, что

задание, поступившее в очередь раньше других, имеет самый высокий приоритет и должно быть запущено первым. Если для запуска такого задания оказывается недостаточно ресурсов, то ожидается момент времени, когда накопится нужный объем требуемых ресурсов и задание будет запущено. Порядок заданий в очереди не может быть нарушен. Данный метод гарантирует запуск задания, но неэффективно расходует ресурсы, так как в период накопления ресурсов для самого приоритетного задания часть ресурсов простаивает. Это основной недостаток метода, поэтому у него существуют различные модификации, призванные повысить эффективность распределения ресурсов.

Более интересный способ решения предлагает метод обратного заполнения BackFill [5,6]. В отличие от рассмотренного ранее FCFS, он требует от пользователей ориентировочную оценку времени выполнения их заданий. Это позволяет выделить ресурсы для заданий заблаговременно. Для этого строится расписание запусков заданий.

В процессе построения расписания ресурсы выдаются заданиям в порядке их приоритетов. Алгоритм Backfill работает по следующему принципу: размещая наиболее приоритетное задание, он определяет момент времени, когда освободится достаточное количество ресурсов, и резервирует эти ресурсы. Задание с меньшим приоритетом может быть запущено вне очереди только в том случае, если оно не будет мешать запуску более приоритетных заданий, например, если свободных ресурсов недостаточно для запуска задания с высоким приоритетом. Таким образом, данный метод позволяет распределять задания по ресурсам с нарушением порядка очереди, что способствует более эффективной загрузке ресурсов. Кроме того, алгоритм обладает следующими достоинствами:

- работая на основе приоритетов, он позволяет предотвратить зависание задания и гарантирует его запуск;
- эффективно загружает ресурсы, не допуская их фрагментации;
- работает с множеством гетерогенных ресурсов;
- достаточно быстро, даже при работе на большом количестве вычислительных узлов.

Основной целью работы является создание прототипа имитационной модели (ИМ) Grid-инфраструктуры. ИМ позволит адекватно оценивать поведение Grid-инфраструктуры при изменяющихся условиях, а также оценивать стратегию управления потоками задач и при необходимости корректировать ее. В качестве выбранных стратегий распределения ресурсов в работе реализуются рассмотренные ранее алгоритмы, а также их модификации [7, 8, 9].

При разработке прототипа ИМ был использован продукт AnyLogic 6.5. Разработанная ИМ представляет собой агентную модель, в которой взаимодействуют два различных типа агентов – агенты-ресурсы и агенты-задания.

Механизм взаимодействия этих двух видов агентов может изменяться в зависимости от используемого алгоритма распределения. В ИМ предполагается, что производительность ресурсов характеризуется временем выполнения на них стандартного задания. Стандартное задание – то, которое требует T секунд процессорного времени одного ресурса со стандартной производительностью V операций в секунду. При этом задания отличаются своей сложностью и требованиями к количеству процессоров, необходимых для их решения. Под сложностью определено время выполнения задания на стандартном ресурсе в течение времени $1T$, $2T$, $3T$ и т.д. Требование к количеству процессоров мы назовем размерностью задания D . D – количество процессоров одинаковой производительности, необходимых для выполнения задания.

Для прототипа ИМ разработан интерактивный интерфейс, который позволяет следить за ходом распределения заданий по ресурсам, отслеживать изменение основных показателей работы модели, воздействовать на управляющие параметры. Кроме

того реализован механизм, позволяющий перенастраивать конфигурацию моделируемой Grid архитектуры. Модель дает возможность анализировать ход выполнения эксперимента по следующим критериям:

- количество используемых и доступных процессорных узлов;
- загруженность кластеров в процентном соотношении по каждому часу и дню;
- количество задач, выполняющихся в данный момент времени, и ждущих выполнения;
- количество отказов различных узлов инфраструктуры;
- среднее использование процессорных элементов кластеров за каждый час.

Разработан прототип ИМ, позволяющей конфигурировать различные Grid-инфраструктуры. Был проведен ряд экспериментов для выбранной Grid-инфраструктуры с заранее определенными заданиями. Результаты экспериментов сравнивались с результатами других симуляторов [8, 9] для тех же начальных условий. Сравнительный анализ данных результатов показал неполное совпадение. Этот факт объясняется тем, что в разработанной модели были учтены не все возможные факторы, но при устранении этих недостатков созданная модель может применяться для моделирования реальной Grid-инфраструктуры. Агентный подход является одним из эффективных средств при решении задачи аллокации для Grid-инфраструктур.

Литература

1. **Грушин Д. А., Поспелов А. И.** Система моделирования Grid: реализация и возможности применения. Институт системного программирования РАН (ИСП РАН). М.
2. **Garey M. R. and Johnson D. S.** Computers and Intractability; a Guide to the Theory of Np-Completeness. W. H. Freeman & Co., 1990. P. 338, ISBN: 0716710455.
3. **Cera Marcia C., Pezzi Guilherme P., Mathias Elton N., Maillard Nicolas, Navaux Phillipe O. A.** Improving the Dynamic Creation of Processes in MPI-2 // Lecture Notes in Computer Science LNCS 4192 Recent Advantages in Parallel Virtual Machine and Message Passing Interface. 2006. Vol. 4192. P. 247–255. ISBN-10: 3-540-39110-X ISBN-13: 978-3-540-39110-4.
4. **Schwiegelshohn U. and Yahyapour R.** Analysis of First-Come-First-Serve Parallel Job Scheduling // In Proceedings of the 9th SIAM Symposium on Discrete Algorithms, 1998. P. 629–638. http://www-ds.e-technik.uni-dortmund.de/~yahya/papers/cei_soda98.pdf
5. **Коваленко В. Н., Корягин Д. А.** Организация ресурсов грид. Препринт ИПМ РАН, № 63, стр. 1-25, Москва, 2004.
6. **Коваленко В. Н., Коваленко Е. И., Корягин Д. А., Любимский Э. З.** Метод опережающего планирования для грид. Препринт ИПМ РАН, М., 2005.
7. **Коваленко В. Н., Коваленко Е. И., Корягин Д. А., Семячкин Д. А.** Управление параллельными заданиями в гриде с неотчуждаемыми ресурсами ИПМ им. М.В.Келдыша РАН. М., 2007.
8. GridSim: A Grid Simulation Toolkit for Resource Modelling and Application Scheduling for Parallel and Distributed Computing. <http://www.buyya.com/gridsim/>
9. SimGrid: a Generic Framework for Large-Scale Distributed Experimentations. <http://simgrid.gforge.inria.fr/publis.html>