

2. GNU Multiple Precision Arithmetic Library. – URL: <http://gmplib.org/>, 2010.

3. Кнут Д. Искусство программирования. Т. 2. Получисленные методы. – М.: Вильямс, 2000. – 832 с.

**И.Н. Лозгачев, А.В. Сенин**

Нижегородский государственный университет им. Н.И. Лобачевского

**ИСПОЛЬЗОВАНИЕ СИМУЛЯТОРА  
ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ GSSIM  
В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ  
ПЛАНИРОВАНИЯ**

Высокопроизводительные вычисления на системах кластерного типа – важный инструмент для решения широкого круга задач в науке и промышленности. Согласно списку Top 500 самых производительных суперкомпьютеров [1] с каждым годом наблюдается рост числа узлов и процессорных элементов в системах – лидерах списка. Пропорционально числу вычислительных элементов и числу пользовательских задач растет сложность управления такими системами. Ключевая роль отводится здесь алгоритмам планирования, распределяющим задания по вычислительным ресурсам. Можно показать, что правильно выбранный алгоритм планирования может существенным образом повысить эффективность использования кластера [2].

Задача планирования на кластерных системах относится к числу NP – трудных, что делает невозможным нахождение точного решения. На практике используются те или иные приближенные алгоритмы, для многих из которых неизвестны теоретические обоснования их эффективности. Поэтому в большинстве современных работ, посвященных алгоритмам планирования на кластерах, исследователи сравнивают новые алгоритмы с существующими, проводя демонстрационные запуски на выбранных конфигурациях и потоках задач. Вполне естественно, что использовать для этих целей реальные высокопроизводительные системы обычно не представляется

возможным, так как доступ к крупнейшим кластерам сильно ограничен. Возможная альтернатива – использовать специальные симуляторы высокопроизводительных систем, позволяющих проверить интересующие аспекты работы алгоритма на произвольных системах без использования дорогостоящего оборудования.

В данной работе рассматривается вопрос интеграции системы управления вычислительной инфраструктурой «Метакластер» [3], разрабатываемой в ННГУ им Н.И. Лобачевского, с симулятором Grid для исследования реализованных алгоритмов планирования.

**Требования к симулятору.** Алгоритмы планирования системы управления «Метакластер» позволяют распределять вычисления на многокластерной системе, учитывая при этом как неоднородности между отдельными кластерами, так и возможную неоднородность внутри каждого кластера. Для проведения адекватных исследований необходимо, чтобы инфраструктура симулятора позволяла симитировать максимально широкое число сценариев использования системы. Это позволяет выдвинуть следующие требования к симулятору:

- возможность гибкого задания вычислительных ресурсов кластера с возможной неоднородностью по таким параметрам, как: скорость процессора, объем оперативной памяти, число вычислительных элементов (ядер);
- возможность задания параметров сети передачи данных: пропускная способность, латентность, топология;
- возможность симуляции нескольких кластеров одновременно.

Сравнение эффективности работы алгоритмов планирования проводится на основании метрик, подсчитанных по результатам планирования. Существуют различные метрики, усредненные значения которых используются для оценки эффективности компонентов планирования. К наиболее распространенным критериям относятся:

- время ожидания запуска (waiting time);

- время отклика (response time) – сумма времени ожидания запуска ( $T_w$ ) и времени работы ( $T_r$ ) задачи  $T_w + T_r$  ;

- время замедления (slowdown) – отношение времени отклика ко времени работы задачи  $\frac{T_w + T_r}{T_r}$  ;

- предельное время замедления (bounded slowdown) – метрика, значение которой определяется следующей формулой  $\max\left(\frac{T_w + T_r}{\max(T_r, \tau)}, 1\right)$ , где  $\tau$  – некоторая граничная константа времени работы задачи (как правило, устанавливается равной 10 с).

К требованиям, предъявляемым к симулятору, также относятся возможность подсчета указанных метрик и предоставление средств визуализации расписания (например, диаграмма Ганта).

**Выбор симулятора.** На данный момент существует ряд проектов по созданию симуляторов высокопроизводительных вычислений: GridSim, OptorSim, Monarch, ChicSim, SimGrid, MicroGrid, GSSIM. Большинство из них ориентированы на Grid-симуляции и созданы для запуска на операционных системах семейства Unix. Ниже приведена сравнительная характеристика наиболее популярных симуляторов:

| Характеристики симулятора                            | Симуляторы  |              |            |             |             |                |            |
|------------------------------------------------------|-------------|--------------|------------|-------------|-------------|----------------|------------|
|                                                      | Grid-Sim[4] | OptorSim [5] | Monarc [6] | ChicSim [7] | SimGrid [8] | Micro-Grid [9] | GSSIM [10] |
| Репликация данных                                    | +           | +            | +          | +           | -           | -              | +          |
| Накладные расходы на операции дискового ввода/вывода | +           | -            | +          | -           | -           | +              | +          |
| Планирование задач пользователя                      | +           | -            | +          | +           | +           | +              | +          |
| Резервация ресурсов                                  | +           | -            | -          | -           | -           | -              | +          |

| Характеристики<br>симулятора        | Симуляторы  |              |            |             |             |                |            |
|-------------------------------------|-------------|--------------|------------|-------------|-------------|----------------|------------|
|                                     | Grid-Sim[4] | OptorSim [5] | Monarc [6] | ChicSim [7] | SimGrid [8] | Micro-Grid [9] | GSSIM [10] |
| Симуляция на основе Workload архива | +           | -            | -          | +           | -           | -              | +          |
| Поддержка различных моделей QoS     | +           | -            | -          | -           | -           | -              | +          |
| Генерация побочного трафика         | +           | +            | -          | -           | +           | +              | +          |
| Язык программирования               | C           | Java         | Java       | Parsec      | C           | Java           | Java       |

Для тестирования разрабатываемых алгоритмов командой разработчиков системы управления «Метакластер» был выбран симулятор GSSIM. В основе сделанного выбора лежит ряд преимуществ симулятора GSSIM: легкость переноса на различные операционные системы, поддержка потоков задач в формате Workload Parallel Archive, резервация ресурсов и др. Ключевым при выборе также стал тот факт, что в отличие от других симуляторов GSSIM изначально проектировался для тестирования именно алгоритмов планирования и содержит ряд специфичных функций, отсутствующих в симуляторах общего назначения: генерация потока задач с заданным распределением параметров, возможности визуализации расписания и другие.

**Особенности Grid-симулятора GSSIM.** В основе любого вычислительного эксперимента лежат параметры вычислительной системы, исследуемый алгоритм планирования и набор задач пользователя. Для описания этой конфигурации GSSIM использует отдельные модули:

- описание вычислительной системы – содержит набор кластеров с характеристиками производительности, параметры коммуникационной среды (учитываются топология, латентность и пропускная способность), набор хранилищ данных, список очередей задач, а также генератор критических ситуаций;

- описание эксперимента – включает в себя набор задач пользователя (содержит требования к ресурсам, время старта и окончания счета, приоритет), планировщик уровня Grid, планировщик уровня кластера, плагин подсчета времени, необходимого для решения конкретной задачи.

В качестве дополнительной возможности реализована поддержка Workload Parallel Archive – базы данных, содержащей историю выполнения вычислительных задач с реальных кластеров и Grid по всему миру. Данная функциональность очень удобна и важна, поскольку решает проблему генерации вычислительных задач и позволяет посмотреть, как будет себя вести исследуемый алгоритм планирования на реальных системах.

После проведения вычислительного эксперимента GSSIM генерирует отчет, содержащий следующую информацию:

- график распределения CPU по задачам пользователя;
- график средней загрузки CPU по каждому кластеру отдельно;
- график выполнения задач;
- график выполненных резерваций ресурсов;
- статистика загрузки сети.

Данная информация может быть использована в дальнейшем для заключения выводов об эффективности того или иного алгоритма планирования.

**Интеграция системы управления «Метакластер» и симулятора GSSIM.** Симулятор GSSIM предоставляет удобную среду для исследования алгоритмов планирования. Однако для использования этих возможностей с алгоритмами «Метакластера» необходимо портировать алгоритмы в среду GSSIM. Существует два решения этой задачи:

- переписывание алгоритмов планирования на язык Java (в настоящее время алгоритмы написаны на C#) и внедрение созданных модулей в симулятор;
- создание некоторого связующего звена между планировщиком и симулятором.

Первое решение неудобно тем, что при модификации алгоритма в системе управления придется переписывать и модули,

внедряемые в симулятор. Поэтому используется второй вариант, а в качестве связующего звена выступает веб-сервис, перенаправляющий запросы от симулятора к планировщику. Выбор веб-сервисов в качестве интерфейса взаимодействия обусловлен в основном соображениями ускорения разработки: C# и Java имеют хорошо развитые средства работы с веб-сервисами.

Основные функции веб-сервиса:

- добавление/удаление вычислительных узлов (AddNode, DeleteNode);
- добавление/удаление задач (AddTask, DeleteTask);
- обработка критических событий с перепланировкой очередей исполнения (ответ на возникшее событие определенного типа).

Для взаимодействия с веб-сервисом реализованы GridSchedulerPlugin и LocalSchedulerPlugin:

- планировщик уровня Grid просто перенаправляет входящие задачи на уровень ниже;
- планировщик уровня кластера подает запросу веб-сервису, вызывая методы AddTask или DeleteTask, и получает на выходе очередь задач. Затем ставит их на исполнение.

Для вычисления приблизительного времени работы программы на системе с конкретным набором ресурсов взят стандартный плагин, входящий в состав GSSIM.

Общение между симулятором и веб-планировщиком осуществляется по протоколу SOAP с использованием автоматически сгенерированного WSDL. Такие сущности, как Task или Node, имеют некоторые различия в наборе характеристик у системы управления «Метакластер» и симулятора GSSIM. Для решения этой проблемы написан код по преобразованию типов.

Работа по интеграции алгоритмов планирования системы управления «Метакластер» с симулятором GSSIM еще не закончена. К настоящему времени проработана схема интеграции, написаны основные функции на стороне веб-сервиса и частично на стороне симулятора. Планируется закончить интеграцию и провести тестирование реализованных в «Метакластере» алгоритмов на различных вычислительных системах и потоках задач.

Работа выполняется в лаборатории Intel-ННГУ «Информационные технологии» (ITLab).

### Список литературы

1. TOP500 Supercomputing Sites [Электронный ресурс]. – URL: <http://www.top500.org>.
2. Гергель В.П., Кустикова В.Д., Сенин А.В. Интеграция системы управления интегрированной средой высокопроизводительных вычислений «Метакластер» с подсистемой планирования Maui // Вестн. ННГУ. Информационные технологии. – 2010.
3. Гергель В.П., Сенин А.В. Разработка системы управления интегрированной средой высокопроизводительных вычислений «Метакластер» // Вестн. ННГУ. Информационные технологии. – 2010.
4. Официальная страница симулятора GridSim [Электронный ресурс]. – URL: <http://www.buyya.com/gridsim>.
5. Официальная страница симулятора OptorSim [Электронный ресурс]. – URL: <http://edg-wp2.web.cern.ch/edg-wp2/optimization/optorsim.html>.
6. Dobre C., Stratan C. Monarc simulation framework // Politehnica University of Bucharest, 2004.
7. Ranganathan K., Foster I. Decoupling computation and data scheduling in distributed data-intensive applications // University of Chicago.
8. Официальная страница симулятора SimGrid [Электронный ресурс]. – URL: <http://simgrid.gforge.inria.fr>.
9. The MicroGrid: a scientific tool for modeling computational grids / Song H.J. [et al.]. – IEEE. 2000.
10. Официальная страница симулятора GSSIM [Электронный ресурс]. – URL: <http://www.gssim.org>.