

© 2012 г. Б.М. ГЛИНСКИЙ, д.т.н.,
 А.С. РОДИОНОВ, д.т.н.,
 М.А. МАРЧЕНКО, к.ф.-м.н.,
 Д.И. ПОДКОРЫТОВ,
 Д.В. ВИНС

(Институт вычислительной математики и математической геофизики
СО РАН, Новосибирск)

МУЛЬТИ-АГЕНТНОЕ МОДЕЛИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ ПРОГНОЗА МАСШТАБИРУЕМОСТИ АЛГОРИТМОВ И НАСТРОЙКИ ИХ ПАРАМЕТРОВ

Производительность вычислительной техники возрастает стремительными темпами, и если в недавнем прошлом параллельные программы исполнялись, быть может, на нескольких десятках вычислительных ядер, то сейчас (говоря о массовых вычислениях) используется до нескольких десятков тысяч ядер, а в ближайшем будущем счёт пойдёт на многие миллионы. При этом вычислительные алгоритмы, как правило, являются более консервативными по сравнению с развитием средств вычислительной техники и желательно заранее знать, насколько эффективным будет их исполнение при доступе к вычислительным комплексам будущего. Имитационное моделирование позволяет выявить узкие места в алгоритмах, понять пути их модификации, подобрать параметры в зависимости от количества ядер. В работе [1] показана возможность применения агенто-ориентированной системы имитационного моделирования для решения некоторых проблем, возникающих при создании суперЭВМ экзафлопсной производительности.

В данной работе рассматриваются особенности поведения различных вычислительных алгоритмов при их реализации на гибридном кластере Сибирского суперкомпьютерного центра (ССКЦ) и оценка их масштабируемости, проведённая с помощью мульти-агентной системы моделирования AGNES (AGent NEtwork Simulator) [2].

1. Краткие сведения о системе моделирования AGNES

Мульти-агентная система (МАС) моделирования AGNES реализована на Java с применением свободно распространяемой агентной платформы JADE [3], что обеспечивает её переносимость. Кроме того, использование системы JADE в качестве платформы избавляет разработчика моделей от необходимости заботиться о базовых функциях МАС: средствах взаимодействия, регистрации агентов, распределённого запуска приложения, совместимости с другими МАС.

Система состоит из агентов, которых можно разделить на две группы: управляющие агенты (УА), которые создают среду моделирования, и функциональные агенты (ФА), которые образуют модель, работающую в среде моделирования. Основная задача функциональных агентов – это имитирование работы исследуемой системы. Управляющие агенты занимаются контролем хода эксперимента и сохранением результатов. Основные задачи УА:

- Инициализация и запуск модели;
- Сбор и хранение информации о ходе моделирования;
- Синхронизация модельного времени;
- Балансировка нагрузки между вычислительными узлами, участвующими в моделировании;
- Взаимодействие с пользователем (вывод отчетов, предоставление возможности влиять на ход моделирования);
- Обеспечение отказоустойчивости, восстановление модели.

При запуске модели, все ФА разделяются на виртуальные кластеры, над каждым таким кластером назначается контролирующий агент (КА), который является разновидностью УА. Основные функции контролирующих агентов:

- Идентификация отказа агентов в среде моделирования;
- Пересылка всех управляющих команд функциональным агентам;
- Хранение информации для восстановления агентов при сбоях и/или откатах;
- Восстановление модели при сбое и /или откате.

Внутри AGNES циркулируют два типа сообщений: управляющие команды и информационные сообщения внутри модели. Для моделирования важно собирать и хранить информационные сообщения, т.е. все обмены данными внутри самой модели. Для решения этой задачи служат агенты логгеры. Они подписываются на все сообщения определенного типа и получают их копии. В зависимости от специфики модели существует необходимость сбора сообщений определенного типа либо от конкретного агента, для этого у логгера можно настроить фильтр и собирать только значимую для исследователя информацию. Эта возможность реализована за счет структуры FIPA сообщений, а также удобных механизмов классификации и фильтрации сообщений.

Также при моделировании важно иметь информацию о состоянии функциональных агентов. Благодаря сервису «желтых страниц» JADE, УА могут найти всех интересующих агентов образующих модель, опрашивать их, сохранять у себя нужную информацию о состоянии ФА.

2. Исследование масштабируемости вычислительных алгоритмов

В качестве алгоритмов для исследования возможности применения системы AGNES для исследования масштабируемости вычислительных алгоритмов были выбраны методы статистического моделирования (методы Монте-Карло) и сеточные методы. Атомарной, независимой частицы в модели вычислений выбран вычислительный узел и исполняемый на нем код алгоритма. Каждый функциональный агент эмулирует поведение вычислительного узла кластера, и программу вычислений, работающую на этом узле. Вычисления в модели представляются в виде набора примитивных операций (вычисление на ядре; запись/чтение данных в память; парный обмен данными; синхронизация данных между вычислителями) и временных характеристик каждой операции. На данном этапе разработки рассматривалась единственная гибридная архитектура вычислительного кластера, соответствующая архитектуре имеющегося в распоряжении авторов кластера НКС-30Т Сибирского суперкомпьютерного центра коллективного пользования (ССКЦ КП) СО РАН. Для калибровки моделей использовались данные профилирования реальных вычислительных программ, исполняемых на этом кластере с использованием до нескольких десятков тысяч вычислительных ядер, что позволяет надеяться на учёт в задержках на обмены системной составляющей.

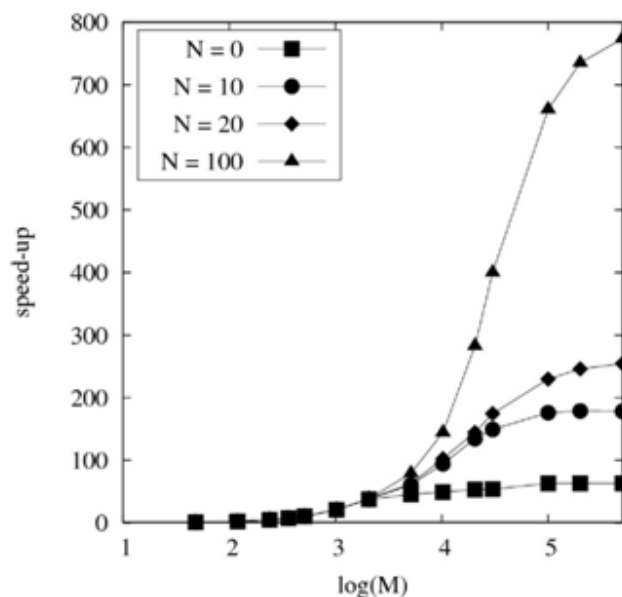


Рис. 1. Сравнение ускорения распределенного статистического моделирования для разных вариантов организации обмена данными для числа ядер M до 500 000 (горизонтальная ось – в логарифмическом масштабе). N – число промежуточных сборщиков; M – число ядер.

2.1. Модель вычислений по методу Монте-Карло

Методы статистического моделирования Монте-Карло широко применяются при решении задач газовой динамики. Метод основан на представлении газа множеством дискретных частиц (каждая из которых представляет собой большое количество реальных молекул), для которых задан стохастический процесс их столкновения друг с другом. Эволюция множества частиц описывается как равномерное прямолинейное движение, прерываемое в случайные моменты времени мгновенными актами парных столкновений. Для упрощения алгоритма и существенного ускорения счёта, фазы перемещения и столкновения частиц разделены между собой и чередуются, а столкновительные партнеры выбираются только в пределах той же самой ячейки (без учёта взаимного расположения). Схема работы параллельной реализации метода основана на очевидном разделении выборки на части с последующим «усреднении средних». При этом объём вычислений на ядре не определяется заранее, ядро генерирует подвыборку порциями равного объёма, вычисление завершается после получения процессом «сборщиком» заданного количества частичных средних. Кажется очевидным снижение времени вычислений обратно пропорционально числу задействованных ядер, что подтверждают численные эксперименты при использовании до 968 ядер. Однако моделирование показало, что при большом количестве ядер сборщик перестаёт справляться с потоком входящих средних и ускорение замедляется. Выход заключается в каскадном усреднении. Моделирование позволило подобрать количество промежуточных сборщиков (см. рис. 1).

2.2. Модель сеточного метода вычислений

Сеточные методы рассматривались применительно решению задачи механики сплошных сред.

Для имитации сеточных методов реализованы два класса функциональных агентов:

DataBus. Центральный вычислитель, собирающий итоговую информацию о вычислениях и агрегирующий её. Агент регистрирует свой сервис, все вычислители находят его. Далее они в течение своей жизни отправляют ему данные о вычислениях. Агент *DataBus* агрегирует полученную информацию и с заданной периодичностью сохраняет её на диск.

Grid. Вычислитель, имитирующий расчет сеточных методов. Моделируются вычисления, когда область режется вдоль одной оси. Таким образом, получается, что у каждого вычислителя есть пересечение по данным максимум с 2-ми вычислителями (“крайние” вычислители обмениваются только с одним соседом) Каждый вычислитель, после завершения итерации расчетов, должен обмениваться данными о граничных областях со своими соседними агентами.

Таблица 1. Модель супер-вычислений должна обязательно содержать агента *DataBus*, и множество агентов вычислителей, возможно с различными параметрами. Критериями остановки модели является: либо общее время работы, либо количество итераций вычислений, которое должен смоделировать каждый вычислитель. В качестве выходных данных собираются отчеты аналогичные методам Монте-Карло.

Количество ядер	Реальное время счета	Модельное время счета
256	10743	
500	5537	
800	3411	
1200	2317	2269
2400	1397	1431
3600	1023	1011
4800	779	752
6000		705
9600		642
19200		570
38400		503

В таблице представлены результаты вычислений собранные на реальных данных, и смоделированные данные. Моделирование не проводилось на слишком маленьких объемах вычислителей, поэтому пересечение реальных и модельных данных видно только в средней части таблицы. Данные показывают, что модель ведёт себя правдоподобно.

Заметное снижение эффекта от добавления новых ядер связано с возрастанием количества обмениваемых данных между полосами области вычислений. При проведении вычислительных экспериментов на разных моделях моделировалось исполнение алгоритмов на архитектурах, содержащих до полутора миллионов ядер. При этом соответствующее количество агентов размещалось на количестве ядер до 144 (это требовало использования нескольких виртуальных Java-машин) при использовании кластера НКС-30Т Сибирского суперкомпьютерного центра, что показывает высокую эффективность моделирования.

3. Заключение

В своём исследовании мы показали, как предварительное моделирование алгоритма может помочь при оценивании его эффективности на разном количестве вычислительных ядер (на конкретной архитектуре вычислительного кластера). Модели вычислений выбирались из соображений возможности качественной и количественной (на количестве ядер до нескольких десятков тысяч) проверки адекватности модели. Поведение алгоритмов вполне предсказуемо, однако даже в этом случае модели оказались полезными для подбора конкретных значений параметров алгоритмов, обеспечивающих их наибольшую производительность. Исследование проводилось в рамках выполнения Государственного контракта № 07.514.11.4016 “Исследования и разработка методов имитационного моделирования функционирования гибридных экзафлопсных вычислительных систем” в рамках ФЦП “Исследования и разработки по приоритетным направлениям развития научно-технического комплекса России на 2007-2013 годы”.

СПИСОК ЛИТЕРАТУРЫ

1. *Б.М. Глинский, А.С. Родионов, М.А. Марченко и др.* Агентно-ориентированный подход к имитационному моделированию суперЭВМ экзафлопсной производительности в приложении к распределенному статистическому моделированию // Вестник ЮУрГУ, 2012. № 18 (277), Вып. 12., с. 93-106. 2012. Р. 93-106.
2. *Д.И. Поджорытов.* Агентно-ориентированная среда моделирования сетевых систем AGNES // Ползуновский вестник, 2012. № 2/1, с. 94-99. 2012. Р. 93-106.
3. *Bellifemine, F. L.; Caire, G. & Greenwood, D.* Developing Multi-Agent Systems with JADE //Wiley, 2007.