УДК 004.942, 519.876.55

Исследование управляющих процессов в суперкомпьютерных системах на основе мультиагентного моделирования

Д.В. Винс, Б.М. Глинский, А.С. Родионов

В работе представлен подход к исследованию управляющих алгоритмов в плане их применимости для вычислительных систем будущего поколения. Для этого создаются мультиагентные модели управляющих алгоритмов, ориентированных на миллионы вычислительных ядер. Производится анализ эффективности этих управляющих алгоритмов, и при необходимости даются рекомендации по их модификации.

Ключевые слова: мультиагентное моделирование, управление большими системами.

1. Введение

На общем собрании Национальной суперкомпьютерной технологической платформы (НСТП) 28 сентября 2011 г. утверждено решение Правления о Концепции по развитию технологий высокопроизводительных вычислений на базе суперЭВМ экзафлопного класса на 2012–2020 годы. Планируется создание в 2014–2015 годах суперЭВМ производительностью 10 – 15 PFlops, в 2017–2018 годах – 100 PFlops, и в 2020 году – запуск суперкомпьютера экзафлоспной производительности. Основу вычислительной мощности экзафлопсного суперкомпьютера составят разрабатываемые ускорители, ориентированные на определённый класс стратегически важных задач. При этом нельзя быть уверенным, что современное программное обеспечение будет работать на таких вычислительных системах (ВС) так же эффективно.

Говоря о программном обеспечении, прежде всего, предполагаются алгоритмы управления процессом вычислений и вычислительные параллельные алгоритмы для решения сложных задач. Алгоритмы управления вычислительной системой должны удовлетворять требованиям по обеспечению качества обслуживания: загруженности, гарантированному времени выполнения поступающих ресурсных запросов, отказоустойчивости и эффективности работы. От того, насколько эти алгоритмы удовлетворяют этим требованиям, зависит их эффективность.

Большемасштабные системы, к которым, в частности, относятся современные и будущие ВС, характеризуются сложным характером алгоритмов управления и стохастической природой входящих потоков данных и задач. Это требует реализации распределённой системы управления, что включает в себя разработку алгоритмов синхронизации объектов (или процессов), функционирующих на различных узлах вычислительной системы, и распределённых алгоритмов обеспечения отказоустойчивости. Эффективность реализации, в свою очередь, зависит от равномерности распределения (балансировки) вычислительной нагрузки по узлам ВС во время функционирования распределённой программной системы [1, 2], каковой является, в частности, распределённая система управления [3]. Исследование и модифи-

кация алгоритмов управления делает необходимым проведение имитационного моделирования подобных систем.

В задачах управления потоком заданий и распределения ресурсов в вычислительных системах метод имитационного моделирования используется или использовался многими научными коллективами: например, в INRIA (Франция) при исследовании мультикластерных VLIW архитектур, в Мичиганском технологическом институте для управления внутренним трафиком, в Технологическом университете г. Делфт, Нидерланды, для сравнительного анализа алгоритмов динамического управления загрузкой и др.

В ИВМиМГ СО РАН развивается мультиагентный подход, который применим как для задачи имитации вычисленных процессов, так и для имитации работы распределённой системы управления. В качестве атомарной, независимой частицы в имитационных моделях выбран вычислительный узел и исполняемый на нём код алгоритма. В зависимости от исследуемой модели каждый функциональный агент эмулирует поведение вычислительного узла системы и программу вычислений, работающую на этом узле. Поведение вычислительного узла представляется набором алгоритмов планирования и балансировки вычислительной нагрузки. Вычисления представляются в виде набора примитивных операций (вычисление на ядре; запись/чтение данных в память; парный обмен данными; синхронизация данных между вычислителями) и временны́х характеристик каждой операции [5].

2. Система моделирования AGNES (AGent NEtwork Simulator)

Первоначально разработанная для моделирования телекоммуникационных и информационных сетей [6,7], система AGNES показала свою эффективность и для моделирования работы распределенной системы управления [8] и исполнения высокопроизводительных параллельных программ [9, 10].

Пакет AGNES базируется на Java Agent Development Framework (JADE) [4]. JADE – это мощный инструмент для создание мультиагентных систем, и он состоит из 3-х частей: среда исполнения агентов; библиотека базовых классов, необходимых для разработки агентной системы; набор утилит, позволяющих наблюдать и администрировать МАС.

JADE приложение обладает двумя важными свойствами для агентных систем:

- *распределённость*, JADE позволяет создавать приложения, запускаемые на локальной сети;
- *универсальность*, поддержка стандарта FIPA обеспечивает лёгкость взаимодействия агентов JADE с другими программными, аппаратными или комплексными агентами, поддерживающими этот стандарт.

Благодаря этому инструменту разработчик имеет ряд готовых средств для управления агентами:

- создание, удаление, регистрация и миграция между вычислителями агентов;
- регистрация функций агентов в единой базе;
- взаимодействие между агентами;
- отслеживание сообщений внутри системы;
- графическая поддержка отладки во время разработки агентов.

AGNES использует все эти возможности, а также расширяет мультиагентную систему до системы моделирования.

Среда моделирования AGNES состоит из двух типов агентов:

- управляющие агенты (УА), которые создают среду моделирования;
- функциональные агенты (ФА), которые образуют модель, работающую в среде моделирования.

Приложение AGNES – это распределённая МАС, называемая платформой. Платформа AGNES состоит из системы контейнеров, распределённых в сети (рис. 1). Обычно на каждом

хосте находится по одному контейнеру (но при необходимости их может быть несколько). Агенты существуют внутри контейнеров. В системе может быть только один главный контейнер, который представляет собой точку начальной загрузки платформы. Главный контейнер создаётся первым, все созданные позже контейнеры должны быть зарегистрированы в главном контейнере. На главном контейнере обязательно работают управляющие агенты AMS (Agent Management System) и DF (Directory Facilitator) [7].

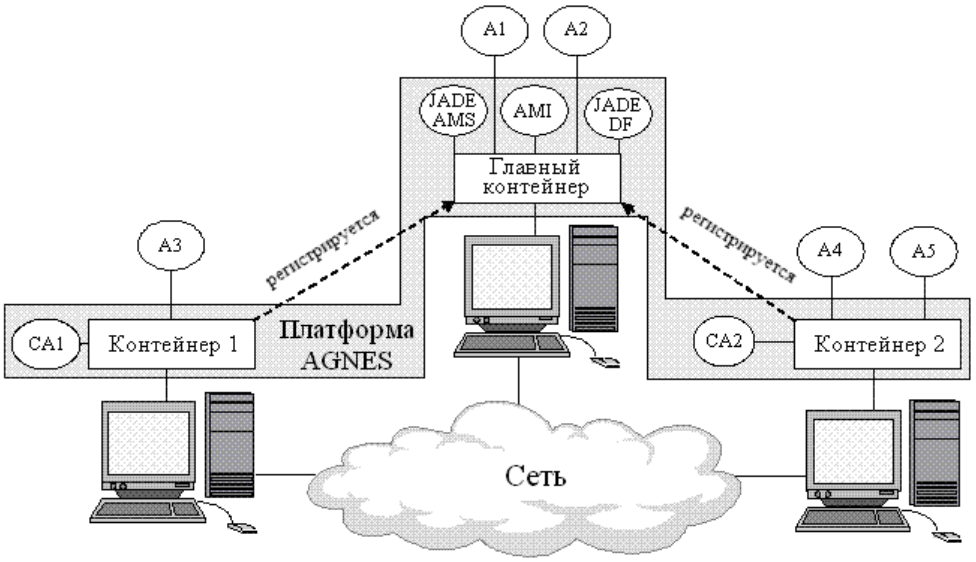


Рис. 1. Платформа AGNES

В настоящее время пользователь может:

- контролировать состояние платформы, т.е. видеть количество контейнеров, список агентов на каждом из контейнеров, создавать или удалять агенты, подключать или отключать контейнеры;
- обмениваться сообщениями с агентами, т.е. отправлять любые сообщения любому агенту, а также получать ответные сообщения от агента как реакцию на свой запрос;
- наблюдать структуру сети, если модель можно представить в виде графа, узлами которого являются агенты, а рёбрами – информационные связи между ним;
- имеет механизмы глобального управления моделированием (приостановка, возобновление, преждевременное прекращение моделирования, получение промежуточных результатов).

Следующие методы должны быть реализованы у AGNES-агента независимо от модели:

- `SaveBackup(agent Sender, time BackupMoment, array BackupData)` – сохранение резервных данных другого агента в своем хранилище;
- `SendBackup(agent Receiver, time BackupMoment, agent BackupAgent)` – отправка резервных данных из своего хранилища указанному агенту;
- `SetTime(time ModelTime)` – синхронизация модельного времени у агента;
- `GetAgentTime()` – получить данные о внутреннем времени агента;
- `SendLog()` – широковещательная рассылка своего состояния всем подписанным на это событие агентам;
- `Ping()` – команда для проверки работоспособности агента.

Чтобы обеспечить высокий уровень отказоустойчивости, AGNES реализует несколько механизмов:

- отсутствие централизованного хранения данных для восстановления;
- хранение необходимой информации ведётся подобно peer-to-peer сетям, т.е. она располагается частями на разных агентах среды моделирования и хранится с избытком для гарантии её восстановления;

– динамическое изменение хранилищ информации во время работы среды моделирования.

То есть основные принципы улучшенной отказоустойчивости среды моделирования – это децентрализация хранилищ и избыточность информации.

Преимуществами AGNES являются:

– малый объём кода имитационных программ;
– балансировка нагрузки при исполнении;
– доступность проблемно-ориентированных библиотек;
– возможность динамического изменения модели в ходе эксперимента.

3. Моделирование системы управления нагрузкой вычислительных систем

Для исследования управляющих процессов современных и будущих вычислительных систем создана и исследуется модель распределённой системы управления. Основные требования к системе: возможность гарантированного обслуживания центром части поступающих заданий, основанная на конкурентном доступе к разделяемым ресурсам и субъективной пользовательской оценке важности результатов выполнения задач, а также обеспечение отказоустойчивой работы.

Разработанная модель системы управления вычислительными системами ЦКП (рис. 2) включает в себя программные агенты, реализующие модели: внешних источников задач, распределителей и контроллеров ресурсов, вычислительных систем. Модель внешней среды источников заданий представлена однотипными программными агентами, имитирующими пользователей, отправляющих задания на ЦКП. Статистика, собранная о характеристиках заданий, выполненных пользователями, а также о выделенных для них ресурсах хранится в агенте сбора и анализа статистики. Этот агент подбирает однотипные задания каждого пользователя и выделяет оценки реальных параметров *T*, *Rcpu* и *E*, (*T* – максимальное время выполнения задания, *Rcpu* – объём необходимой оперативной памяти, *E* – количество параллельных ветвей, которые используются при планировании).

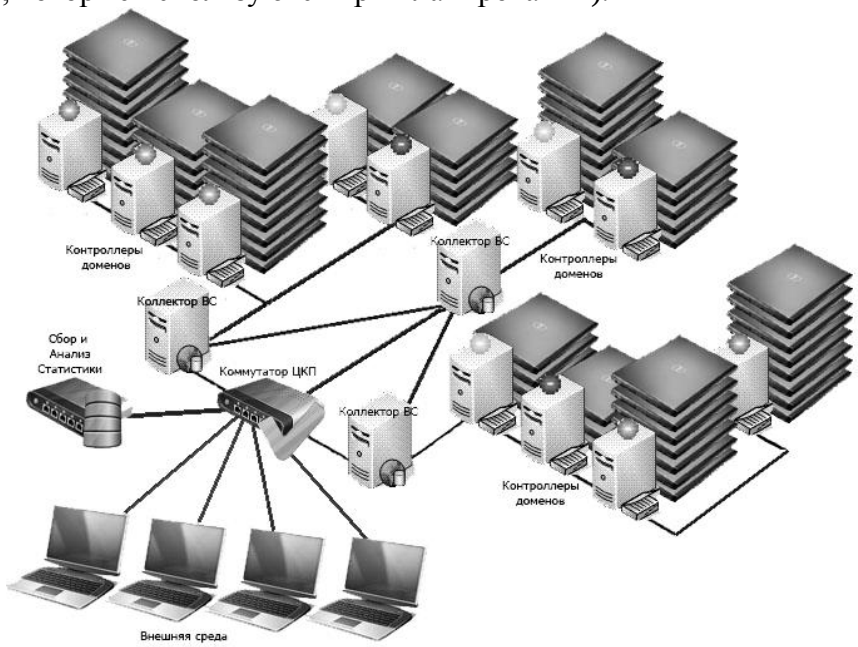


Рис. 2. Мультиагентная модель управления. Общий вид

Агент-коммутатор периодически получает сообщения от коллекторов ВС о состоянии их очереди заданий (количество заданий, ожидающих планирования), наличии вычислительных

ресурсов и памяти. При поступлении пользовательского задания агент-коммутатор отправляет агенту сбора и анализа статистики информацию о характеристиках задания и пользователе. В ответ он получает средние значения зафиксированных характеристик решённых заданий для данного пользователя и имена ВС, на которых они чаще решались – основная и альтернативная. Оценивая информацию об этих ВС, коммутатор передаёт задания наиболее подходящей из них.

Агент-коллектор ВС принимает информацию от контроллеров доменов о получении задании и об его завершении. При получении задания агент-коллектор просматривает реализующуюся сейчас очередь заданий алгоритмом BackFill. Если не удаётся включить задание в эту очередь, оно отправляется ожидать формирования новой очереди. При получении сигнала о завершении задания с помощью всё того же алгоритма оценивается возможность занять освободившиеся ресурсы заданием из текущей очереди или из ожидающих заданий. Когда текущая очередь подходит к концу, начинается формирование следующей очереди с помощью генетического алгоритма.

Первоначальная популяция тут создаётся алгоритмом ортогональной упаковки прямоугольников Best-Fit-Decreasing-Height (BFDH) [11]. Оператором кроссинговера является функция частичного обмена родительских особей задачами. Оператором мутации выступает функция изменение набора выделяемых ресурсов для заданного набора задач. Признак остановки – стабилизация минимума суммарного времени решения $T(S)$ и суммарного штрафа $R(S)$ за задержку решения.

Контроллеры доменов следят за физическим состоянием всех ресурсов ВС, а также за выделением, освобождением этих ресурсов для выполнения заданий. Выделены следующие виды контроллеров доменов:

1. Контроллер резерва ВУ хранит информацию обо всех ресурсах, отправленных в «холодный» или «горячий» резерв. Из резерва ресурсы выдаются по запросу из других доменов. Здесь также хранится информация о ресурсах выведенных из эксплуатации узлов для проведения профилактики или ремонта. При восстановлении физических параметров ресурса он возвращается в домен свободных ВУ.

2. Контроллер свободных ВУ хранит характеристики всех готовых к работе ресурсов ВС. Здесь реализуется выделение положенного числа ресурсов и их освобождение.

3. Контроллер работающих ВУ хранит информацию о ресурсах, выданных для решения данной задачи, а также запасных (перспективных) ресурсах. Осуществляется контроль за исполнением параллельного задания и анализ необходимости перераспределения. В этом случае выявляется порция вычислительной нагрузки (ветвь параллельной программы) на определённом ВУ из этой области, которую необходимо передать другим узлам, чтобы избежать дисбаланса и определить целевой узел, на который следует перенести эту нагрузку. Целевой узел отыскивается из числа ВУ, находящихся в данном домене. Если же таковых нет, то контроллер, взаимодействуя с контроллерами соседних доменов работающих ВУ, доменов свободных ВУ и резерва, узнаёт адрес вычислительного узла, способного принять дополнительную нагрузку. Если и тут происходит неудача, то с этой целью он обращается к коллектору ВС.

Необходимо отметить, что представленная модель легко включаема в процесс оперативного управления реальной вычислительной системой. Для этого на датчики соответствующих агентов подаётся реальный поток заданий пользователей и реальные данные о физическом состоянии ВУ и о выполняющихся на них вычислениях. На начальном этапе интеграции модели можно подключить её в качестве консультативной поддержки реальной системы. В этом случае модель подключается следующим образом: модель работает параллельно с центром управления ВС (ЦУ) и принимает решения, обладая теми же данными. Тут модель может не только «обучаться» (т.е. анализируя управленческие решения, принятые моделью и ЦУ при необходимости корректировать собственное поведение), но и «подсказывать» ЦУ. В вычислительной системе, как в «живой» системе, любое

изменение (приход новой задачи, завершение вычисления задачи, изменение структуры ресурсов, например, отказ одного из узлов, и т.д.) приводит к инициализации принятия нового решения об управлении. Перед принятием собственного решения модель может параллельно посчитать несколько вариантов развития событий (различные варианты приоритетизации задач в очереди и распределения ресурсов по задачам) в долгосрочной перспективе, а затем на их основе выбрать более предпочтительное решение. Затем это решение на управление выдаётся как альтернативное консультативное для ЦУ.

4 Экспериментальное исследование модели распределенной системы управления ВС

Начальные (тестовые) запуски модели были призваны подобрать субоптимальные параметры для анализа статистики и алгоритмов планирования. Дальнейшее исследование модели уже проводится над реальными данными существующей системы. Наиболее доступными были данные Центра коллективного пользования Сибирского суперкомпьютерного центра СО РАН. Была воссоздана коммуникационная среда кластера НКС-30Т+GPU, который состоит из неоднородных узлов: G5; G6; G7; SL390, SMP. На вход агента-коммутатора пускались задания, зарегистрированные системой управления кластера (PBS Pro) за 2011–2013 гг. (рис. 3). Было создано от 124 (2011 г.) до 189 (2013 г.) агентов-пользователей, которые отправляли такие же задания и в то же время коммутатору, что и реальные пользователи в указанный период.

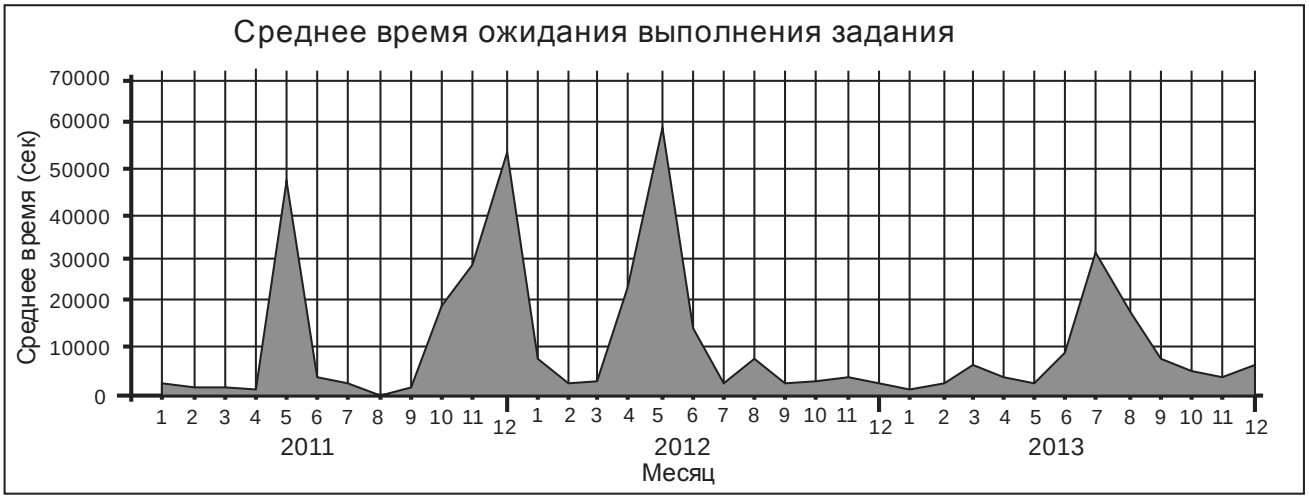


Рис. 3. Статистика нагрузки кластеров ССКЦ СО РАН за 2011 – 2013 годы

Для демонстрации работы имитационной модели системы управления потоками заданий и подобранных алгоритмов планирования на основе данных статистики были произведены 3 опыта:

1. моделирование загрузки ЦКП без агента-коммутатора (задания отправлялись на те же очереди, что и в рассматриваемый период), без использования генетического алгоритма формирования расписаний;

2. те же условия, но в работу включился генетический алгоритм;

3. те же условия, но подключён агент-коммутатор (модель системы сама принимала решения, в какую очередь отправить задание).

Результаты данных запусков приведены на рис. 4 (см. на следующей странице).

Как видно из рис. 4А, самые плохие результаты (увеличение среднего времени ожидания выполнения относительно данных статистики) показало моделирование без использования алгоритмов формирования очереди заданий. Использование генетического алгоритма для этих целей привело к незначительному изменению среднего времени ожидания. Включение же в принятие решения о планировании агента-коммутатора позволило практически по всем месяцам уменьшить среднее время ожидания выполнения. Это объясняется тем, что многие задания были отправлены на менее загруженные в данный момент очереди, что видно на рис. 4Б – нагрузка активно перераспределяется между очередями заданий.

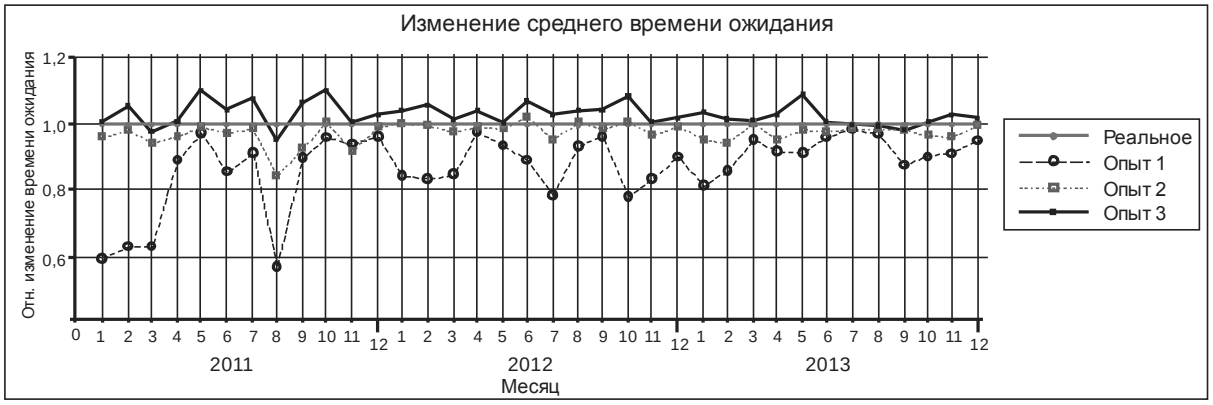
Также производился запуск модели для проверки её отказоустойчивости. Для этого при моделировании были принудительно «выведены из строя» 5% элементарных машин. В результате, все вышедшие из строя ЭМ были заменены вовремя, а нагрузка перераспределена между другими машинами.

5. Заключение

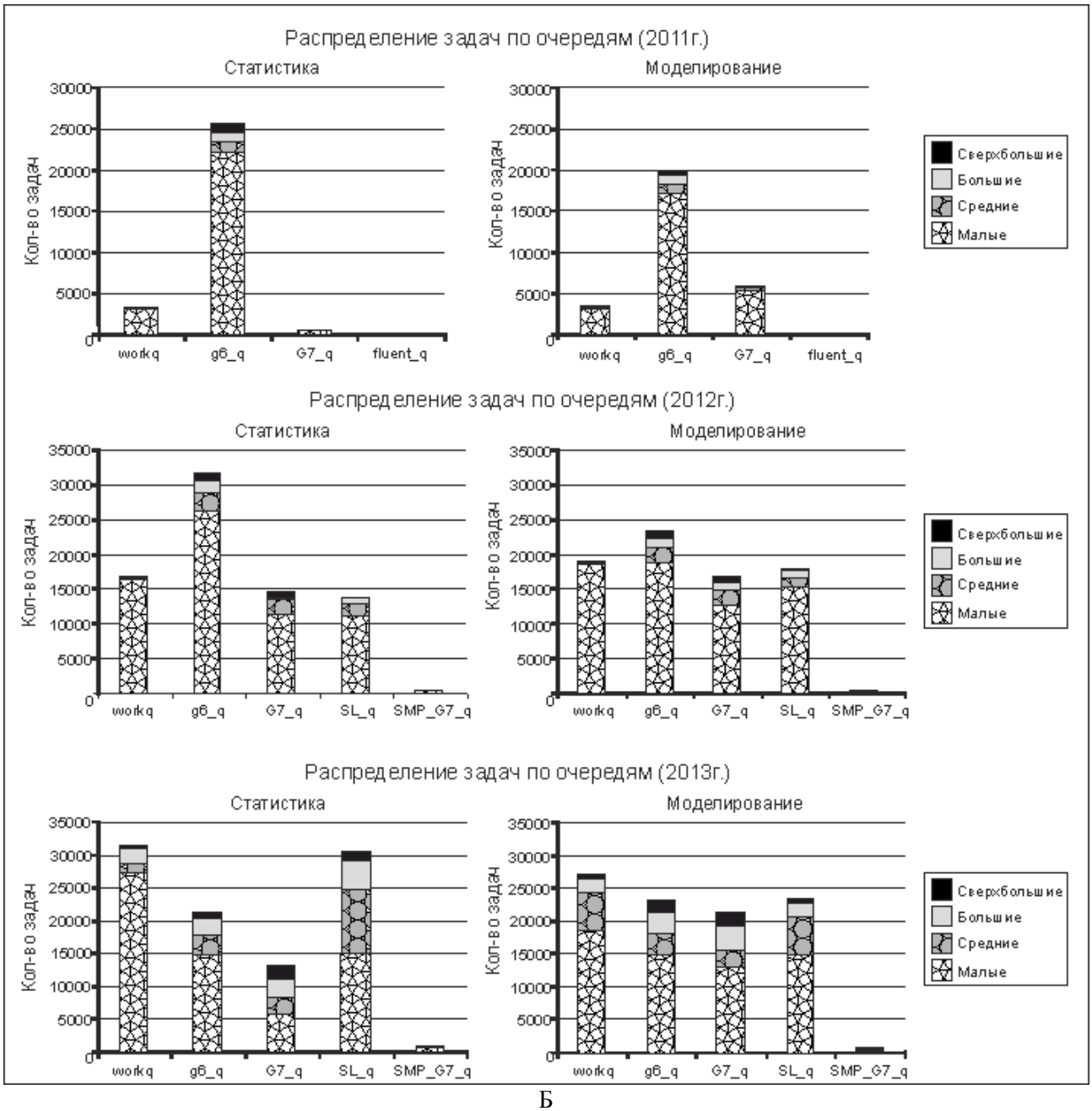
В работе представлена мультиагентная система имитационного моделирования AGNES. Система может быть использована как для исследования управляющих алгоритмов и процессов в современных и перспективных вычислительных системах, так и для исследования масштабируемости различных алгоритмов для суперкомпьютеров с учётом особенностей архитектуры суперкомпьютера.

При тестировании модели распределённой системы управлении вычислительной системой были подобраны оптимальные параметры для алгоритмов планирования. Запуски модели, где моделируемый поток заданий был заменён на реальный (статистика работы ЦКП ССКЦ СО РАН за 2011–2013 годы), показали, что принимаемые управленческие решения позволяют сократить среднее время ожидания начала выполнения задания на 1–10%. Также задания располагаются в вычислительной системе более рационально, т.к. при постановке их на выполнение учитывается карта сети связи.

Таким образом, с помощью AGNES можно проводить исследования, связанные с разработкой алгоритмов для суперкомпьютеров экзафлопсного класса с учётом различных предполагаемых архитектур данных суперкомпьютеров. Приведённые тесты имитационного моделирования хорошо соотносятся с реальными данными, полученными при запуске исследуемых программ и алгоритмов на кластерах ЦКП ССКЦ СО РАН.



А



Б

Рис. 4. Результаты запуска модели ЦКП на основе статистики 2011 года.
А – изменение среднего времени ожидания в очереди заданий для различных опытов.
Б – распределение заданий по очередям ЦКП во время опыта 3 (работа агента-коммутатора)

Литература

1. *Копысов С.П.* Динамическая балансировка нагрузки для параллельного распределённого МДО//Труды Первой Всероссийской научной конференции «Методы и средства обработки информации». М.: Изд-во МГУ, 2003 г., сс. 222-228.
2. *Курилов Л.С.* Прогностическая стратегия балансировки загрузки для невыделенных кластерных систем//Труды Первой Всероссийской научной конференции «Методы и средства обработки информации». М.:Изд-во МГУ, 2003 г., сс. 413-418.
3. *Linda F. Wilson and Wei Shen* Experiments In Load Migration And Dynamic Load Balancing In Speedes. Proceedings of the 1998 Winter Simulation Conference. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds, pp. 483-490.
4. *Michael T. Goodrich*: Simulating Parallel Algorithms in the MapReduce Framework with Applications to Parallel Computational Geometry CoRR, URL: `http://dblp.uni-trier.de/db/journals/corr/corr1004.html\#abs-1004-4708` (дата обращения 9.9.2014)
5. *Глинский Б.М., Родионов А.С., Марченко М.А., Подкорытов Д.И., Винс Д.В.* Агентно-ориентированный подход к имитационному моделированию суперЭВМ экзафлопсной производительности в приложении к распределённому статистическому моделированию // Вестник ЮУрГУ, 2012. № 18 (277), Вып. 12., с. 94-99.
6. *Podkorytov, D., Rodionov, A., Sokolova, O., Yurgenson, A.* Using Agent-Oriented Simulation System AGNES for Evaluation of Sensor Networks. LNCS, vol. 6235, pp. 247--250, Springer, Heidelberg (2010).
7. *Podkorytov, D., Rodionov, A., Choo, H.*: Agent-based simulation system AGNES for networks modeling: review and researching. Proc. of the 6th Int. Conference on Ubiquitous Information Management and Communication (ACM ICUIMC 2012), ISBN 978-1-4503-1172-4, Paper 115, 4 pages, ACM (2012).
8. *Винс Д.В*. Анализ эффективности системы управления потоком заданий для ЦКП в мультиагентной имитационной модели// Вестник НГУ, 2014. т.12. вып. 2, с. 33-41.
9. *Glinsky, B., Rodionov, A., Marchenko, M., Podkorytov, D., and Weins, D.* Scaling the Distributed Stochastic Simulation to Exaflop Supercomputers. High Performance Computing and Communication \& 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on, Proceedings, pp. 1131–1136, IEEE (2012).
10. *Глинский Б.М., Марченко М.А., Михайленко Б.Г., Родионов А.С., Черных И.Г., Караваев Д.А., Подкорытов Д.И., Винс Д.В.* Отображения параллельных алгоритмов для суперкомпьютеров экзафлопсной производительности на основе имитационного моделирования // Информационные технологии и вычислительные системы, 2013, № 4, с. 3-14.
11. *Coffman E. G.* Performance bounds for level-oriented two-dimensional packing algorithms / E.G. Coffman, M.R. Garey, D.S. Johnson [et al.] // SIAM Journal on Computing. – 1980. Vol. 9. –P. 808-826.

Статья поступила в редакцию 01.09.2014.

Глинский Борис Михайлович

д.т.н., заведующий лабораторией Института вычислительной математики и математической геофизики СО РАН (630090, Новосибирск, пр-т ак. Лаврентьева, д. 6), тел. (383) 3-306-279, e-mail: gbm@sscc.ru

Родионов Алексей Сергеевич

д.т.н., заведующий лабораторией Института вычислительной математики и математической геофизики СО РАН (630090, Новосибирск, пр-т ак. Лаврентьева, д. 6), тел. (383) 3-326-949, e-mail: alrod@sscc.ru

Винс Дмитрий Владимирович

м.н.с., Институт вычислительной математики и математической геофизики СО РАН (630090, Новосибирск, пр-т Ак.Лаврентьева, д. 6), e-mail: wns.dmitry@gmail.com

Investigation of computing and operating processes of computing systems using multi-agent simulation

D. Weins, B. Glinskiy, A. Rodionov

The approach to investigating control algorithms in terms of their applicability for computing systems of the future generation is presented in this paper. Multi-agent models of control algorithms have being created for the purpose that are oriented to simulation of computer systems with millions cores. Analysis of efficiency of modeled algorithms is being done, and recommendations about their modification are proposed.

Keywords: multi-agent simulation, job scheduling, resource management, efficiency of operating algorithms.