УДК 004.942, 519.876.55

Д. В. Винс

Институт вычислительной математики
и математической геофизики СО РАН
пр. Акад. Лаврентьева, 6, Новосибирск, 630090, Россия

E-mail: wns.dmitry@gmail.com

АНАЛИЗ ЭФФЕКТИВНОСТИ СИСТЕМЫ УПРАВЛЕНИЯ ПОТОКОМ ЗАДАНИЙ ДЛЯ ЦКП В МУЛЬТИАГЕНТНОЙ ИМИТАЦИОННОЙ МОДЕЛИ

Представлена мультиагентная модель для имитации и исследования работы системы управления потоком заданий на центры коллективного пользования. Создается модель для исследования алгоритмов планирования распределения ресурсов для потока параллельных заданий на суперкомпьютерные центры. Продемонстрированы результаты экспериментального исследования модели на статистике реальной нагрузки ССКЦ СО РАН.

Ключевые слова: планирование заданий, управление ресурсами, мультиагентные модели.

Введение: задача создания системы управления потоком заданий

Центр коллективного пользования (ЦКП) – это объединенная вычислительная среда, предназначенная для обслуживания ресурсных запросов пользователей, состоящая из разнородных вычислительных систем (ВС), администрируемых независимо друг от друга и предоставляющих неотчуждаемые ресурсы для общего пользования [1. С. 3–5]. К таким центрам предъявляются требования по обеспечению качества обслуживания – загруженности, гарантированному времени выполнения поступающих ресурсных запросов, отказоустойчивости и эффективности работы.

Рассматриваемые в работе ресурсные запросы – параллельные задания пользователей (назовем их заданиями) помимо ограничений по времени обслуживания, выраженных в виде характеристических функций потери ценности решения, обладают случайным временем выполнения. В качестве исходного условия принимается априорное превышение количества заданий над возможностями обслуживающей системы.

Таким образом, встает вопрос о реализации распределенной системы управления потоком заданий для ЦКП. Реализация распределенной системы управления требует разработки алгоритмов синхронизации объектов (или процессов), функционирующих на различных узлах ЦКП. Эффективность реализации, в свою очередь, зависит от равномерности распределения (балансировки) вычислительной нагрузки по узлам ЦКП во время функционирования распределенной программной системы [2; 3], каковой является, в частности, распределенная система управления [4].

Проблема балансировки вычислительной нагрузки распределенной системы управления возникает, если:

- структура заданий неоднородна, различные их части требуют различных вычислительных мощностей;
- структура вычислительного центра также неоднородна, т. е. разные вычислительные узлы обладают разной производительностью;

Винс Д. В. Анализ эффективности системы управления потоком заданий для ЦКП в мультиагентной имитационной модели // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2014. Т. 12, вып. 2. С. 33–41.

ISSN 1818-7900. Вестник НГУ. Серия: Информационные технологии. 2014. Том 12, выпуск 2

- структура межузлового взаимодействия неоднородна, так как линии связи, соединяющие узлы, могут иметь различные характеристики пропускной способности [5. С. 4].

В работе рассматривается возможность гарантированного обслуживания центром части поступающих заданий, основанная на конкурентном доступе к разделяемым ресурсам и субъективной пользовательской оценке важности результатов выполнения задач.

Многие известные ученые и научные коллективы, работающие в области исследования и проектирования информационных и вычислительных систем и сетей, используют или использовали метод имитационного моделирования в качестве одного из основных инструментов исследования. Непосредственно в задачах управления потоком заданий и распределения ресурсов в мультикластерных системах метод имитационного моделирования применялся, например, в INRIA (Франция) при исследовании мультикластерных VLIW-архитектур, Мичиганском технологическом институте для управления внутренним трафиком, Технологическом университете г. Делфт, Нидерланды, для сравнительного анализа алгоритмов динамического управления загрузкой и др. Также в ИВМиМГ СО РАН создан программный комплекс, который решает задачу объединения вычислительных систем на основе стандартов MPI (NumGRID [6; 7]), однако в данном проекте не рассматривается вопрос балансировки вычислительной нагрузки. В работе предлагается новый метод оптимизации загрузки неоднородных кластеров на основе мультиагентной модели, отличающийся возможностью оперативного принятия управленческих решений, в зависимости от состояния вычислительной среды и загруженности кластера задачами.

Постановка задачи распределения ресурсов вычислительной системы. С точки зрения наличия вычислительных ресурсов вычислительную систему можно представить множеством из *n* вычислителей, множеством *C* количества памяти вычислительных узлов и иерархической моделью коммуникационной сети между ресурсами системы (рис. 1).

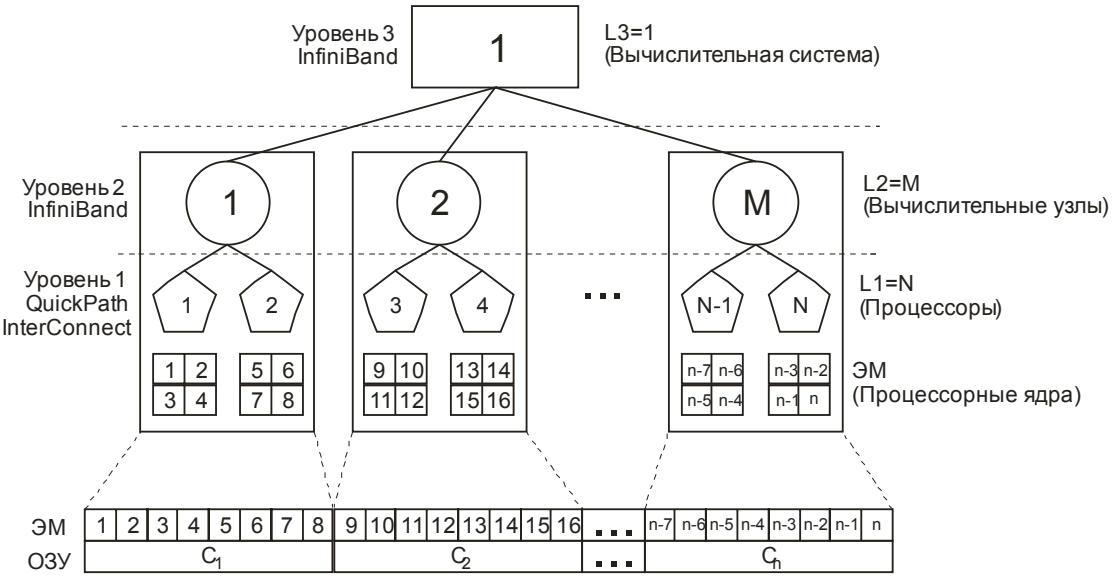


Рис. 1. Иерархическая модель коммуникационной сети между ресурсами системы

Коммуникационная среда системы может быть представлена в виде дерева, содержащего L уровней, $l \in \{1, 2, \ldots, L\}$. Каждый уровень системы образован отдельным видом структурных элементов системы (например, телекоммуникационные шкафы, вычислительные узлы и т. п.), которые объединены каналами связи своего уровня. На уровне L размещено Ln элементов. Для каждого элемента на уровне L задано количество Z_k, $k \in \{1, 2, \ldots, Ln\}$, его прямых дочерних узлов. Дополнительно определим функцию $g(l.k_1.k_2)$ – номер уровня, на котором находится элемент, являющийся ближайшим общим предком для элементов $k_1, k_2 \in \{1, 2, \ldots, Ln\}$ уровня l. Например, общим предком для элементов 3, 5 уровня 1 является элемент 1 уровня 1, т. е. $g(3, 3, 5) = 1$. Для каждого уровня коммуникационной среды извест-

ны значения показателей производительности каналов связи на нем. Пусть b_l – значение пропускной способности каналов связи на уровне l ($\lceil b_l \rceil$ = байт/с).

Рассмотрим систему с точки зрения поступающих ресурсных запросов. При отправке на выполнение пользователь запрашивает определенное количество параллельных ветвей задания – E, объем оперативной памяти *Rcpu*, а также время *Tmax*, за которое задание выполнится. Таким образом, ресурсный запрос – задание $G = \langle E, Rcpu, T\max \rangle$, причем если объем оперативной памяти пользователь может рассчитать точно (хотя обычно он этого не делает), то время выполнения он точно рассчитать не может, поэтому оба эти параметра обычно указываются с запасом. Таким образом, значение этих параметров лишь их оценка сверху. Для некоторых заданий может понадобиться запуск дополнительных ветвей, что также необходимо учитывать при обработке запроса, т. е. принимать E как оценку реального значения снизу [8].

Вторым способом представления ресурсного запроса, является информационный граф задания (рис. 2).

В информационном графе $G = (V, E)$, $V = \{1, 2, \ldots, M\}$ – множество параллельных ветвей программы; $E \subseteq V \times V$ – множество информационно-логических связей между ее параллельными ветвями (обмены информацией). Обозначим за d_{ij} вес ребра $(i, j) \in E$, отражающий объем данных, передаваемый по нему за время выполнения программы ($\lceil d_{ij} \rceil$ = байт).

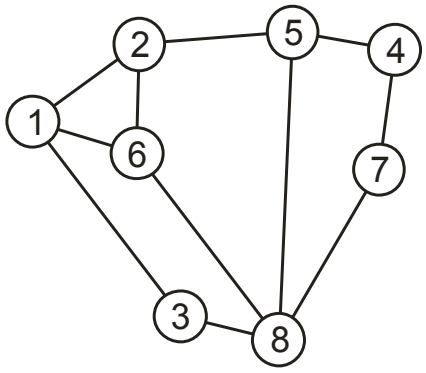


Рис. 2. Информационный граф задания

Существенный вопрос при обработке заданий – время их выполнения. Время выполнения параллельной программы зачастую зависит от эффективности ее вложения в коммуникационную сеть вычислительной системы, т. е. необходимо наложить информационный граф задания на иерархическую модель коммуникационной сети вычислительной системы так, чтобы для всех $i, j \in E$ и расположенных на уровне l, $g(l, i, j)$ был минимальным и $d_{ij} \le b_l$.

Однако зачастую анализ графа программы до начала ее выполнения невозможен, что не позволяет оптимально разместить задачу по ресурсам сразу. Следовательно, вопрос оптимального вложения информационного графа задания необходимо решать во время выполнения задания, когда все связи между вершинами информационного графа становятся известны [9].

Но выше мы рассмотрели проблемы распределения ресурсов для одного задания, а на ЦКП поступает поток разнородных заданий. Для планирования потока заданий необходимо решить следующую задачу: имеется поток из J задач, необходимо составить расписание S такое, чтобы каждая $j \in \{1 \ldots J\}$ задача получила E_j элементарных машин производительностью EP_j, $Rcpu_j$ оперативной памяти и началась через время TS_j. Время выполнения расписания – $T(S) = \max\{TS_j + EP_j \cdot T\max_j\}$ и штраф за простой в очереди $-R(S) = \max\{TS_j\}$ должны быть минимальными.

Имитационная модель системы управления ЦКП

Принципы мультиагентного моделирования. В качестве методологии исследования выбрано имитационное моделирование, которое является одним из основных методов исследования сложных систем вообще и вычислительных систем в частности. Как уже упоминалось, многие известные ученые и научные коллективы, работающие в области исследования и проектирования информационных и вычислительных систем и сетей, используют метод имитационного моделирования в качестве одного из основных инструментов исследования. В современном имитационном моделировании весьма популярен мультиагентный подход, т. е. моделирование с применением мультиагентных систем (МАС). Агент – это сущность, живущая в среде обитания, обладающая сенсорами для восприятия среды и исполнительными механизмами для воздействия на среду обитания.

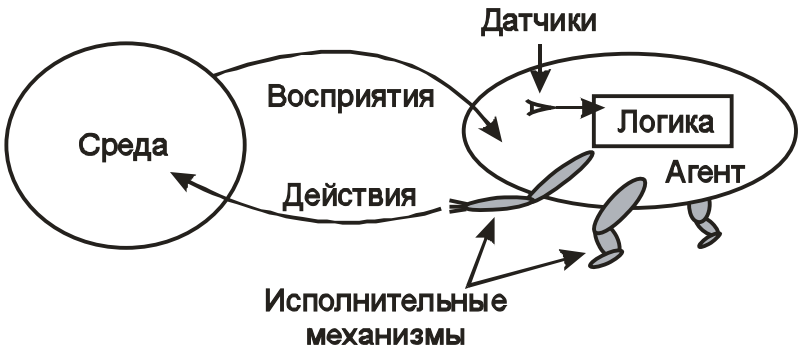


Рис. 3. Интеллектуальный агент

Мультиагентная система (МАС) – это система, образованная несколькими взаимодействующими программными агентами в единой среде. Применение программных агентов для представления компонентов модели обладает рядом преимуществ, наиболее существенными из которых являются:

1) предоставление естественных возможностей интеллектуализации процесса моделирования;

2) существование стандартов взаимодействия программных агентов, позволяющее интегрировать новые модели с существующими агентными моделями и системами, разработанными третьими лицами;

3) существование открытых платформ разработки МАС.

Дополнительным преимуществом применения мультиагентного подхода является возможность перехода от моделирования ЦКП непосредственно к управлению. Стандартизация механизмов взаимодействия между агентами, позволяет заменять программных агентов реальными [10].

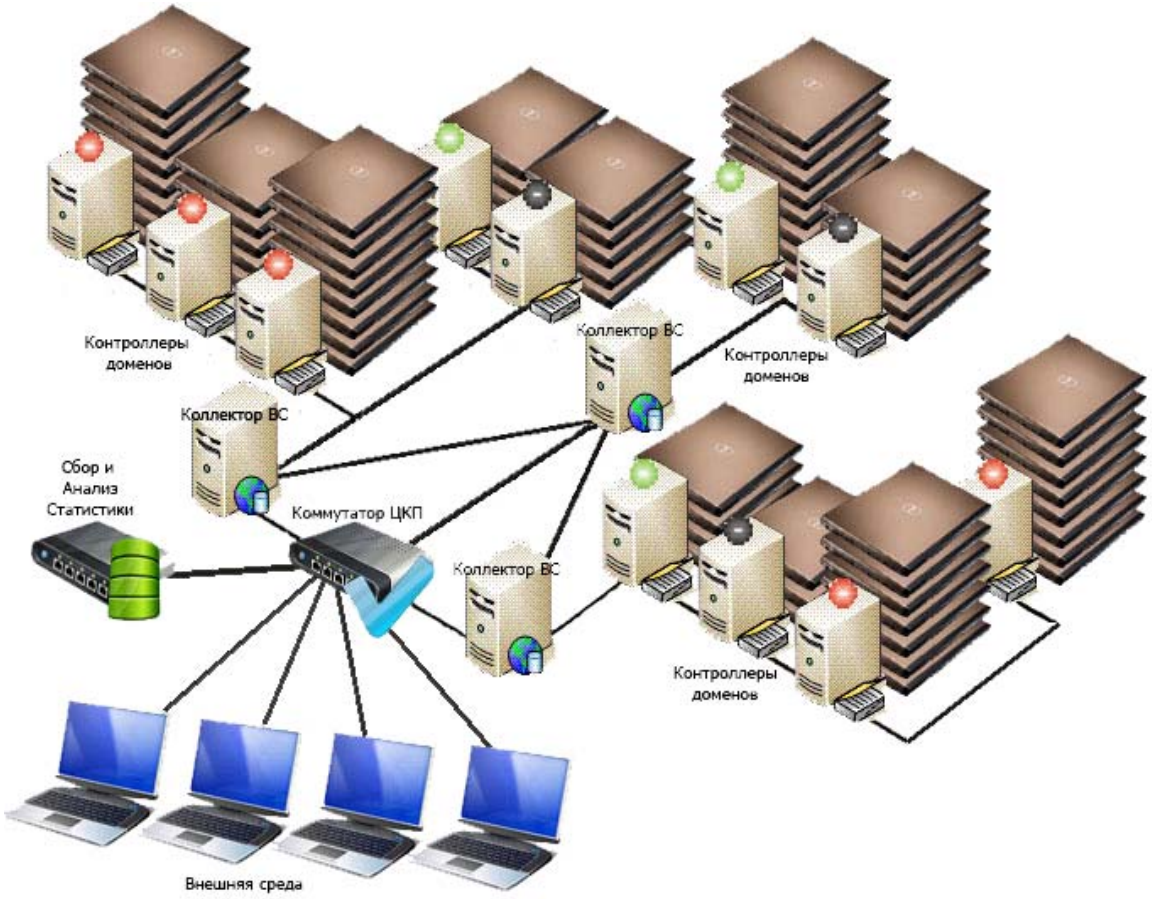


Рис. 4. Мультиагентная модель ЦКП. Общий вид

Мультиагентная модель. Разработанная модель системы управления потоком задач (рис. 4) включает 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, $Rcpu$ – объем необходимой оперативной памяти, E – количество параллельных ветвей), которые используются при планировании.

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 решенных заданий для данного пользователя и имена ВС, на которых они чаще решались, – основная и альтернативная. Оценивая информацию об этих ВС, коммутатор передает задания наиболее подходящей.

Агент-коллектор ВС принимает информацию от контроллеров доменов о получении задании и о его завершении. При получении задания агент-коллектор просматривает реализующуюся сейчас очередь заданий алгоритмом BackFill. Если не удается включить задание в эту очередь, оно отправляется ожидать формирования новой очереди. При получении сигнала о завершении задания с помощью все 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 создается алгоритмом прямоугольной ортогональной упаковки BFDH (Best-Fit-Decreasing-Height) [11]. Оператором кроссинговера является функция частичного обмена родительских особей задачами. Оператором мутации выступает функция изменения 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 доменов.

1. Резерв ВУ хранит информацию о 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 свободный домен.

2. Свободные ВУ хранят характеристики всех ресурсов ВС. Здесь реализуется выделение положенного числа ресурсов и их освобождение.

3. Работающие ВУ хранят информацию о ресурсах, выданных для решения данной задачи, а также запасных (перспективных) ресурсах. Осуществляется контроль за исполнением параллельного задания и анализ необходимости перераспределения.

Экспериментальное исследование модели системы управления потоком заданий ЦКП

Начальные (тестовые) запуски модели были призваны подобрать субоптимальные параметры для анализа статистики и алгоритмов планирования. Дальнейшее исследование модели уже проводится над реальными данными, реальной системы. Наиболее доступными были данные ЦКП ССКЦ СО РАН. Была воссоздана коммуникационная среда кластера НКС-30Т+GPU, который состоит из неоднородных узлов: G5; G6; G7; SL390, SMP.

Запуск модели с данными реальной статистики. На вход агента коммутатора пускались задания, зарегистрированные системой управления кластера (PBS Pro) за 2011–2013 гг. (рис. 5). 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. 5. Статистика нагрузки кластеров ССКЦ СО РАН за 2011–2013 гг.

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;

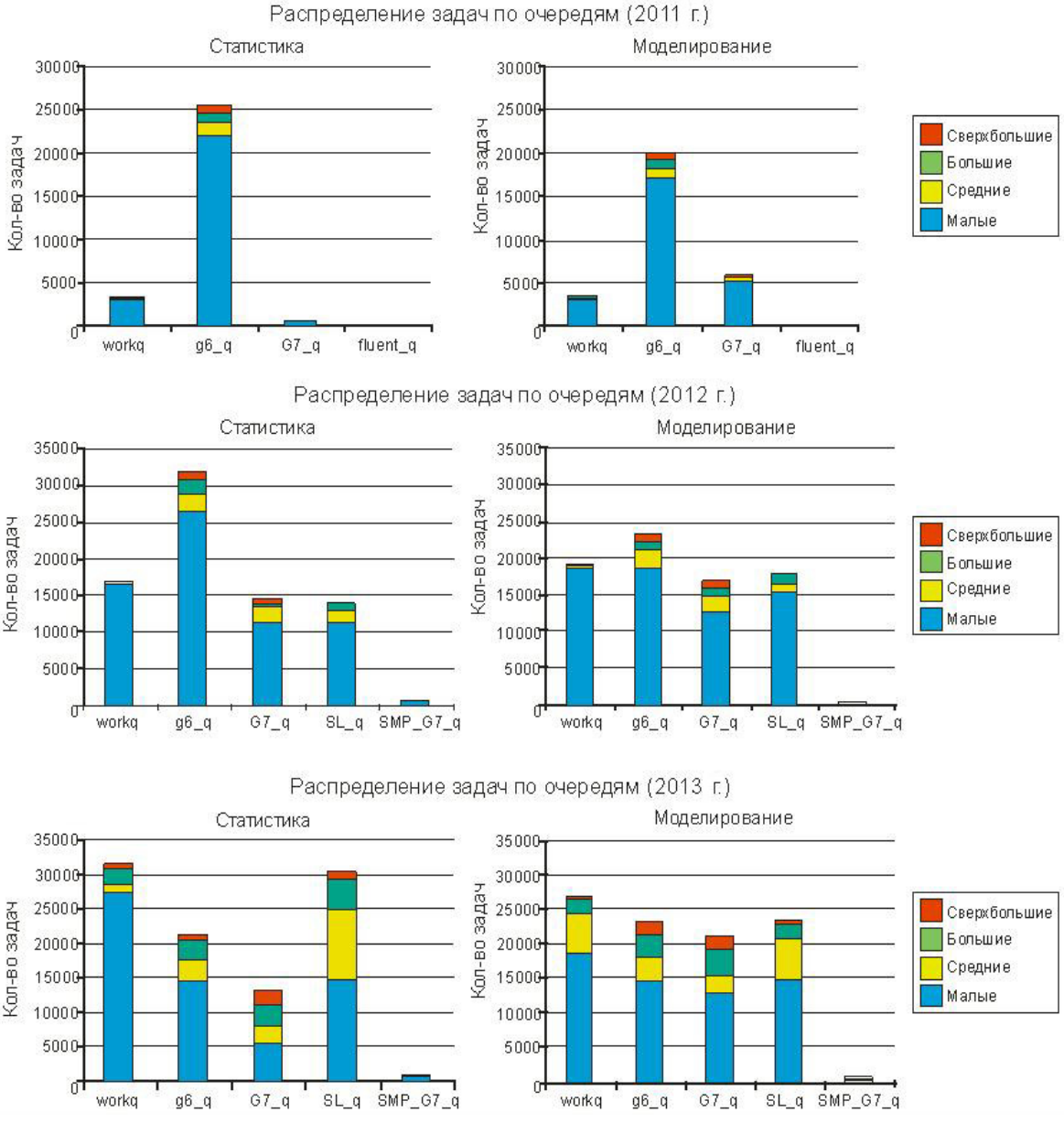
2) те же условия, но в работу включился генетический алгоритм;

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

Результаты данных запусков приведены на рис. 6.

Как видно из рис. 6, *а*, самые плохие результаты (увеличение среднего времени ожидания выполнения относительно данных статистики) показало моделирование без использования алгоритмов формирования очереди заданий. Применение 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. 6, *б* – нагрузка активно перераспределяется между очередями заданий.

а



б

Рис. 6. Результаты запуска модели ЦКП на основе статистики 2011 г.: *а* – изменение среднего времени ожидания в очереди заданий для различных опытов; *б* – распределение заданий по очередям ЦКП во время опыта 3 (работа агента коммутатора)

Также производился запуск модели для проверки ее 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.

Заключение

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

Имея рабочую модель системы, можно включать ее в процесс управления. Получая задачи от пользователей ЦУ (центр управления ЦКП), система помещает их в очередь, и когда возникает задача оптимальной загрузки кластера задачами, то ЦУ запускает несколько имитационных моделей с различными вариантами приоритетов задач в очереди и различными вариантами распределения ресурсов по задачам. Расчет моделей занимает существенно меньшее время, что позволяет сделать множество прогнозов развития событий при том или ином планировании. Собираются данные из моделей, и на их основе принимается решение о том, какой вариант управления более предпочтительный. Так как кластер – это «живая» система, то любое изменение в системе (приход новой задачи, завершение вычисления задачи, изменение структуры ресурсов, например, отказ одного из узлов, и т. д.) может приводить к инициализации принятия нового решения об управлении, т. е. запуску имитационных моделей.

В дальнейшем планируется испытание модели управления ЦКП в вопросе экономии электроэнергии. Вычислительные узлы, которые по прогнозам не будут принимать участие в вычислениях в ближайшее время, переводятся в «горячий» или «холодный» резерв. Загрузка узлов будет производиться не максимально, а оптимально по потреблению электричества.

Также планируется включение в модель системы новых современных архитектур, таких как CPU + GPU, CPU + MIC и др.

Список литературы

1. *Голосов П. Е.* Планирование заданий с временной функцией потери ценности решения в сетевой среде распределенных вычислений: Автореф. дис. … канд. техн. наук. М., ИКСИИ, 2010. URL: http://sovet.mitme.ru/blurb/2010/golosov.pdf

2. *Копысов С. П.* Динамическая балансировка нагрузки для параллельного распределенного МДО // Тр. I Всерос. науч. конф. «Методы и средства обработки информации». М.: Изд-во МГУ, 2003. С. 222–228.

3. *Курилов Л. С.* Прогностическая стратегия балансировки загрузки для невыделенных кластерных систем // Тр. I Всерос. науч. конф. «Методы и средства обработки информации». М.: Изд-во МГУ, 2003. С. 413–418.

4. *Wilson L. F. and Wei Shen.* Experiments In Load Migration And Dynamic Load Balancing In Speedes // Proc. of the 1998 Winter Simulation Conference / D. J. Medeiros, E. F. Watson, J. S. Carson and M. S. Manivannan (Eds.), 1998. P. 483–490.

5. *Миков А. И., Замятина Е. Б., Осмехин К. А.* Метод динамической балансировки процессов имитационного моделирования. Пермь: Изд-во ПГУ, 2004.

6. *Городничев М. А.* Объединение вычислительных кластеров для крупномасштабного численного моделирования в проекте NumGRID // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2012. Т. 10, вып. 4. С. 63–73.

7. *Gorodnichev M., Kireev S., Malyshkin V.* Optimization of Intercluster Communications in the NumGRID // Methods and Tools of Parallel Programming Multicomputer – Second Russia-Taiwan Symposium, MTPP 2010 / Vladivostok, Russia, May 16–19, 2010, Revised Selected Papers. Lecture Notes in Computer Science 6083. Springer, 2010. P. 78–85.

8. *Хорошевский В. Г., Курносов М. Г.* Моделирование алгоритмов вложения параллельных программ в структуры распределенных вычислительных систем / В. Г. Хорошевский, // Тр.

Междунар. науч. конф. «Моделирование-2008» (Simulation-2008). Киев: Изд-во ИПМЭ им. Г. Е. Пухова, 2008. Т. 2. С. 435-440.

9. *Agarwal T. et al.* Topology-Aware Task Mapping for Reducing Communication Contention on Large Parallel Machines // Parallel and Distributed Processing Symposium. 2006. P. 10.

10. *Bellifemine F. L., Caire G., Greenwood D.* Developing Multi-Agent Systems with JADE. Wiley, 2007.

11. *Coffman E. G., Garey M. R., Johnson D. S. et al.* Performance Bounds for Level-Oriented Two-Dimensional Packing Algorithms // SIAM Journal on Computing. 1980. Vol. 9. P. 808–826.

Материал поступил в редколлегию 19.05.2014

D. V. Vins

ANALYSIS OF EFFECTIVENESS OF JOB STREAM MANAGEMENT SYSTEM FOR THE CENTERS OF COLLECTIVE USE IN MULTI-AGENT SIMULATION MODEL

The paper presents multi-agent model for simulation and research work of job stream management system for The Centers Of Collective Use. Is created simulation model for research resource scheduling algorithms for a thread parallel jobs heading on supercomputer centers. Demonstrated the results of experimental research of model on statistic of real load of SSCC SB RAS.

Keywords: job sheduling, resource management, multi-agent simulation.

References

1. Golosov P. E. Scheduling jobs with temporary loss of function of the value of network solutions in the distributed computing environment. Abstract of the thesis. Moscow, IKSI 2010, p. 3–5. URL: http://sovet.mitme.ru/blurb/2010/golosov.pdf

2. Kosipov S. P. Dynamic balancing for parallel distributed MDO. *Proceedings of First Russian Conference «Methods and means for processing the information»*, Moscow, MSU, 2003, p. 222–228.

3. Kurilov L. S. Predictive balancing strategy for unselected cluster systems. *Proceedings of First Russian Conference «Methods and means for processing the information»*, Moscow, MSU, 2003, p. 413–418.

4. Wilson L. F. 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.), 1998, p. 483–490.

5. Mikov A. I., Zamyatina E. B., Osmechin K. A. Method for dynamic balancing process simulation. Perm, PSU, 2004.

6. Gorodnichev M. A. Ob'edinenie vychislitel'nyh klasterov dlya krupnomasshtabnogo chislennogo modelirovaniya v proekte NumGRID [Joining computing clusters for large scale numerical simulation in the NumGRID project]. *Vestnik of Novosibirsk State University. Series*: *Information Technology*, 2012, vol. 10, iss. 4. p. 63–73.

7. Gorodnichev M., Kireev S., Malyshkin V. Optimization of Intercluster Communications in the NumGRID // Methods and Tools of Parallel Programming Multicomputer – Second Russia-Taiwan Symposium, MTPP 2010 / Vladivostok, Russia, May 16–19, 2010, Revised Selected Papers. Lecture Notes in Computer Science 6083. Springer, 2010. P. 78–85.

8. Khoroshevsky V. G., Kurnosov M. G. Simulation of algorithms for Assigning Parallel Program Branches to structure of space-distributed Computer Systems. *Proceedings of International Conference «Simulation-2008»*, Kiev, 2008, vol. 2, p. 435–440.

9. Agarwal, T. et al.Topology-aware task mapping for reducing communication contention on large parallel machines. *Parallel and Distributed Processing Symposium*, 2006, p. 10.

10. BellifemineF. L., Caire G., Greenwood D. Developing Multi-Agent Systems with JADE. Wiley, 2007.

11. Coffman E. G. Garey M. R., Johnson D. S. et al. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 1980, vol. 9, p. 808–826.