УДК 681.31:323

ОПТИМАЛЬНАЯ СТРУКТУРА РАСПРЕДЕЛЕННОГО МЕТАПЛАНИРОВЩИКА ГРИД

Д. В. Семенов[1], **В. В. Корнеев**[1]

Рассматривается подход к выбору древовидной структуры метапланировщика, обеспечивающего распределение по ресурсам грид заданного потока заданий.

Ключевые слова: грид, распределенное планирование заданий, иерархическая древовидная структура метапланировщика грид.

1. Введение. В [1] рассмотрен подход к построению децентрализованного распределенного метапланировщика грид, представляющего собой распределенную самовосстанавливающуюся при отказах совокупность взаимодействующих менеджеров с ациклической структурой информационных связей (рис. 1). Каждый менеджер имеет локальную очередь заданий. Менеджеры M_1 нижнего уровня функционируют на управляющих машинах вычислительных систем (ВС) грид, принимают задания пользователей в свою локальную очередь и посылают задания из своей очереди на исполнение в очередь системы пакетной обработки (СПО) этой ВС. В случае оценки загруженности своей ВС как высокой, менеджер первого уровня посылает задания из своей очереди менеджеру второго уровня M_2. Менеджеры M_2 не имеют собственных вычислительных ресурсов. Количество менеджеров M_2 может быть от одного и более в зависимости от требуемых показателей надежности и пропускной способности грид. Менеджеры M_2 при рассмотрении каждого задания своей локальной очереди оценивают состояние загруженности соседних менеджеров и принимают решение о пересылке задания одному из соседних менеджеров или оставлении в своей очереди для его планирования при изменении состояния загруженности соседних менеджеров. Таким образом, локальные СПО всех ВС грид функционируют параллельно и независимо друг от друга, назначая задания на ресурсы управляемых ВС, что обеспечивает необходимую пропускную способность метапланировщика грид. Совокупность локальных очередей всех менеджеров образует общую очередь заданий грид.

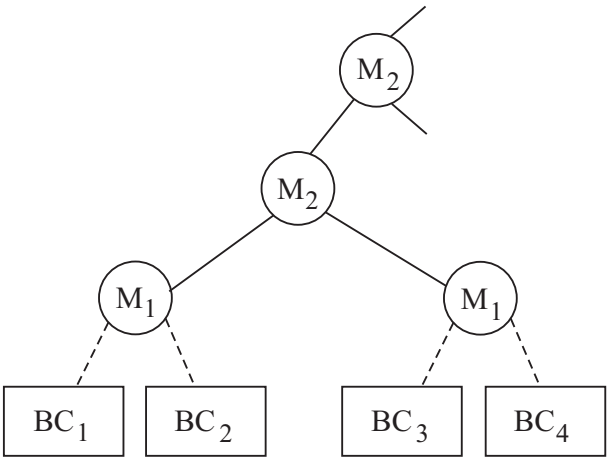

Рис. 1. Структура метапланировщика грид

На основе результатов экспериментов [1] было показано, что за счет изменения параметров алгоритмов работы менеджеров, а также изменения самой стратегии выделения ресурсов пользовательским заданиям можно оказывать значительное влияние на качество планирования пользовательских заданий на ресурсы грид. Децентрализованный распределенный метапланировщик, созданный на основе рассматриваемой модели, используется в грид [2], построенной из вычислительных систем МСЦ РАН (Москва) и его филиалов в других городах России. Предполагается дальнейшее наращивание вычислительных ресурсов и увеличение потоков заданий от пользователей.

Однако в [1] не формулируется обоснование выбора структуры метапланировщика. Авторами не анализируется, будет ли метапланировщик с подобной структурой способен распределять без переполнения очередей менеджеров заданный поток заданий при заданных интенсивностях обработки заданий в менеджерах M_1 и M_2. В настоящее время не существует обоснованного решения выбора оптимальной структуры. В работе [3] рассматривалась проблема выбора оптимальной иерархической структуры метадиспетчера в зависимости от интенсивности поступающего в систему потока пользовательских заданий и доступного числа вычислительных ресурсов, однако авторы остановились лишь на двух уровнях иерархии.

[1] ФГУП “Научно-исследовательский институт “Квант”, 4-й Лихачевский переулок, д. 15, 125438, Москва; Д. В. Семенов, ст. науч. сотр., e-mail: sdvbox@gmail.com; В. В. Корнеев, профессор, зам. директора, e-mail: korv@rdi-kvant.ru

© Научно-исследовательский вычислительный центр МГУ им. М. В. Ломоносова

В настоящей статье, продолжающей [1], рассмотрен подход к выбору структуры метапланировщика с учетом характеристик грид: интенсивность поступающих потоков заданий, количество вычислительных ресурсов в грид и параметров менеджеров. Статья имеет следующую структуру. Во втором разделе представлена модель метапланировщика грид. Третий раздел содержит результаты имитационных экспериментов, подтверждающих состоятельность построенной модели.

2. Модель метапланировщика. Прежде чем приступить к построению модели необходимо ввести ряд пояснений и допущений, аналогичных принятым в [3] и необходимых в методических целях для получения аналитических зависимостей.

Вычислительные ресурсы грид (рис. 2) моделируются множеством вычислительных модулей (ВМ), сгруппированных в кластерные ВС. Каждый кластер состоит из множества вычислительных модулей (ВМ), каждый из которых обрабатывает одно пользовательское задание за время, распределенное по экспоненциальному закону с интенсивностью v. Пусть $N_{\text{ВМ}}$ — число ВМ в ВС и пусть все ВС одинаковы.

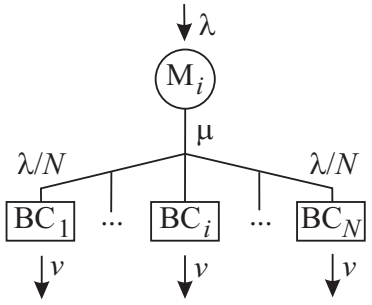

Рис. 2. Модель грид с метапланировщиком

Каждый из кластеров грид может порождать поток пользовательских заданий. Потоки которые имеют распределение Пуассона с параметром $\lambda_i = \dfrac{\lambda_{\text{потока}}}{N}$, где $\lambda_{\text{потока}}$ — интенсивность потока заданий, порождаемого всеми ВС грид, N — количество кластеров в грид.

Менеджеры могут распределять потоки пользовательских заданий по вычислительным ресурсам в соответствии со следующими алгоритмами: 1) случайно; 2) в направлении наименьшей очереди; 3) циклически. Вариант 3 самый простой для моделирования и оценки результатов, так как в этом случае на объекты управления нижестоящего уровня поступает поток одинаковой интенсивности [3]. Этот вариант будем использовать в дальнейшем, в этом случае для менеджеров уровнем ниже интенсивность потока заданий будет равна $\dfrac{\lambda_{\text{потока}}}{k}$, где k — количество объектов управления нижнего уровня.

Каждый менеджер метапланировщика обладает некоторой пропускной способностью, которая зависит от количества объектов управления (связей со смежными менеджерами M_2 или M_1). Чем больше связей, тем меньше пропускная способность μ: $\mu = \dfrac{C}{k_i}$, где C — константа и k_i — количество связей. Будем различать два вида констант: C_1 — константа, характеризующая интенсивность обработки потока заданий менеджером M_1, и C — константа, характеризующая интенсивность обработки потока заданий менеджером M_2.

Сформулируем следующие правила построения метапланировщика грид.

Правило 1. Загруженность ВМ (если весь поток равномерно и без задержек распределяется по ВМ) определяется формулой $\rho = \dfrac{\lambda_{\text{потока}}}{N_{\text{сум}} v}$, где $N_{\text{сум}}$ — суммарное число вычислительных модулей в грид.

Для того чтобы грид была не перегружена, должно выполняться условие $\rho \leqslant 1$. Из этого условия выбираются необходимые параметры $N_{\text{сум}}$ (а возможно, и v, т.е. можно выбирать более или менее производительные ВМ) для того, чтобы метапланировщик справился с обработкой потока заданий с заданной интенсивностью.

Правило 2. Пропускная способность менеджера метапланировщика определяется формулой $\mu = \dfrac{C}{k_i}$, а для менеджера первого уровня — $\mu = \dfrac{C_1}{N_{\text{ВМ}}}$. Можно рассчитать количество ВМ, которое может управляться менеджером первого уровня без перегрузки:

$$\rho = \frac{\lambda_{\text{ВС}} N_{\text{ВМ}}}{C_1} \leqslant 1, \tag{1}$$

где $\lambda_{\text{ВС}}$ — интенсивность потока заданий, поступающего на вход менеджера первого уровня, а $N_{\text{ВМ}}$ — число ВМ в кластере, при этом

$$\lambda_{\text{ВС}} = \frac{\lambda_{\text{потока}} N_{\text{ВМ}}}{N_{\text{сум}}}. \tag{2}$$

Подставив (2) в (1), получим, что для известного C_1 число ВМ в кластере не должно превышать

$$N_{\text{ВМ}} \leqslant \sqrt{\frac{C_1 N_{\text{сум}}}{\lambda_{\text{потока}}}}. \tag{3}$$

Правило 3. Для менеджера M_2 самого высокого уровня иерархии метапланировщика, если одного уровня иерархии недостаточно, должно выполняться условие

$$1 \geqslant \rho = \frac{\lambda}{\mu} = \frac{\lambda_{\text{потока}}\, k}{C}\,, \tag{4}$$

т.е. для заданных значений $\lambda_{\text{потока}}$ и C должно существовать целое $k \geqslant 2$, такое, что выполняется (4).

С учетом перечисленных правил строим метапланировщик распределенными вычислительными ресурсами по следующему алгоритму:

1) для заданной интенсивности потока пользовательских заданий $\lambda_{\text{потока}}$ выбираем необходимое количество $N_{\text{сум}}$ вычислительных модулей с интенсивностью обработки v;

2) разбиваем множество вычислительных модулей на кластеры с учетом правила 2;

3) для управления кластерными системами добавляем на первый уровень иерархии менеджер системы управления с известной интенсивностью обработки паспортов заданий, удовлетворяющей выражению (1);

4) если $\lambda_{\text{потока}} \leqslant \dfrac{C_1}{N_{\text{сум}}}$, то это значит, что для успешного функционирования системы достаточно одного менеджера первого уровня (рис. 2);

5) в ином случае делим ВМ на $k_1 = \dfrac{N_{\text{сум}}}{N_{\text{ВМ}}}$, где k_1 — целое число кластеров под управлением менеджеров первого уровня (рис. 3); $N_{\text{ВМ}}$ получаем из выражения (3);

6) добавляем менеджер метапланировщика второго уровня для управления k_1 менеджерами M_1. В соответствии с правилом 3 имеем

$$k_1 \leqslant \frac{C}{\lambda_{\text{потока}}}\,; \tag{5}$$

в то же время для того чтобы кластер не был перегружен, должно выполняться соотношение

$$1 \geqslant \rho_{\text{ВС}} = \frac{\lambda_{\text{ВС}}}{\mu_{\text{ВС}}} = \frac{\lambda_{\text{потока}} N_{\text{ВМ}}}{k_1 C_1}\,, \quad k_1 \geqslant \frac{\lambda_{\text{потока}} N_{\text{ВМ}}}{C_1}\,. \tag{6}$$

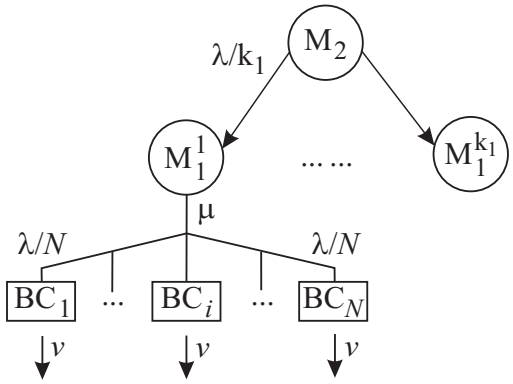

Рис. 3. Модель грид с метапланировщиком с одним менеджером M_2

Сравниваем k_1 в выражениях (5) и (6). Если k_1 удовлетворяет (5) и (6), то менеджер сможет обработать поток пользовательских заданий, и одного уровня иерархии достаточно; если нет, то добавляем еще один уровень, при этом на менеджер нижнего уровня поступает поток с интенсивностью $\lambda = \dfrac{\lambda_{\text{потока}}}{k_1}$. Загруженность менеджера нижнего уровня имеет вид $1 \geqslant \rho = \dfrac{\lambda}{\mu} = \dfrac{\lambda_{\text{потока}} k_2}{k_1 C}$, т.е. при максимальной нагрузке имеем

$$k_2 \leqslant \frac{C k_1}{\lambda_{\text{потока}}} = \left(\frac{C}{\lambda_{\text{потока}}}\right)^2. \tag{7}$$

В то же время для того чтобы кластер не был перегружен, должно выполняться соотношение

$$1 \geqslant \rho_{\text{ВС}} = \frac{\lambda_{\text{ВС}}}{\mu_{\text{ВС}}} = \frac{\lambda_{\text{потока}} N_{\text{ВМ}}}{k_2 C_1}\,, \quad k_2 \geqslant \frac{\lambda_{\text{потока}} N_{\text{ВМ}}}{C_1}\,. \tag{8}$$

Если k_2 удовлетворяет условиям (7) и (8), то метапланировщик сможет обработать поток пользовательских заданий и двух уровней иерархии достаточно. Если нет, то добавляем еще один уровень, при этом на менеджер нижнего уровня поступает поток с интенсивностью $\lambda = \dfrac{\lambda_{\text{потока}}}{k_1 k_2}$. Загруженность менеджера нижнего уровня имеет вид $1 \geqslant \rho = \dfrac{\lambda}{\mu} = \dfrac{\lambda_{\text{потока}} k_2}{k_1 k_2 C}$, т.е. при максимальной нагрузке имеем

$$k_3 \leqslant \frac{C k_1 k_2}{\lambda_{\text{потока}}} = \left(\frac{C}{\lambda_{\text{потока}}}\right)^4. \tag{9}$$

В то же время для того чтобы кластер не был перегружен, должно выполняться соотношение

$$1 \geqslant \rho_{\text{ВС}} = \frac{\lambda_{\text{ВС}}}{\mu_{\text{ВС}}} = \frac{\lambda_{\text{потока}} N_{\text{ВМ}}}{k_3 C_1}\,, \quad k_3 \geqslant \frac{\lambda_{\text{потока}} N_{\text{ВМ}}}{C_1}\,.$$

Из формул (7) и (9) видно, что каждое k_i получается из предыдущего значения возведением в квадрат:

$$k_i \leqslant \frac{Ck_1k_2\dots k_{i-1}}{\lambda_{\text{потока}}} = \left(\frac{Ck_1k_2\dots k_{i-2}}{\lambda_{\text{потока}}}\right)k_{i-1} = (k_{i-1})^2 = \left((k_{i-2})^2\right)^2 = \left(\left((k_{i-3})^2\right)^2\right)^2.$$

Граница снизу не зависит от i и равна $\dfrac{\lambda_{\text{потока}}N_{\text{ВМ}}}{C_1}$. С учетом данного замечания и формулы (5) по индукции получаем следующую систему неравенств:

$$\frac{\lambda_{\text{потока}}N_{\text{ВМ}}}{C_1} \leqslant k_i \leqslant \left(\frac{C}{\lambda_{\text{потока}}}\right)^{2^{i-1}},$$

т.е. нужно найти такое i, при котором бы выполнялось правило 4:

$$\frac{\lambda_{\text{потока}}N_{\text{ВМ}}}{C_1} \leqslant \left(\frac{C}{\lambda_{\text{потока}}}\right)^{2^{i-1}}.$$

Таким образом, зная интенсивность потока пользовательских заданий и интенсивность обработки одного задания ВМ, в соответствии с правилом 1, можно получить минимальное число вычислительных модулей в создаваемой распределенной вычислительной системе. При этом правило 4 позволяет рассчитать необходимое число уровней иерархии метапланировщика, позволяющее равномерно и без задержек распределять задания по вычислительным ресурсам.

3. Экспериментальная проверка предложенной модели. Для того чтобы проверить правильность выводов, сделанных во втором разделе, была создана имитационная модель грид с метапланировщиком, построенным в соответствии со сформулированным подходом. В качестве средства имитационного моделирования использовалась система GPSS World [4]. Ниже представлены некоторые результаты имитационного моделирования. Пусть интенсивность обработки заданий одним ВМ составляет 2 задания в секунду, а интенсивность входного потока 200 заданий в секунду: $v = 2$; $\lambda_{\text{потока}} = 200$. В соответствии с правилом 1, для того чтобы кластерная система справлялась с потоком указанной интенсивности, должно выполняться условие $N_{\text{сум}} \geqslant 100$. В ином случае на входе кластерной системы начнет накапливаться очередь заданий.

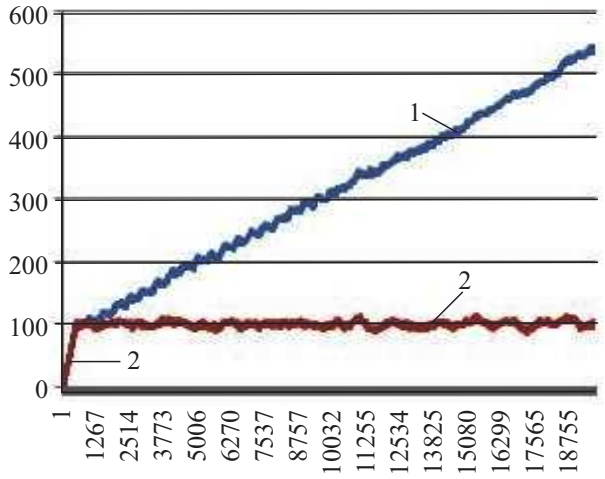

Рис. 4. Зависимости количества заданий в грид от времени: 1) $N = 90$, 2) $N = 110$

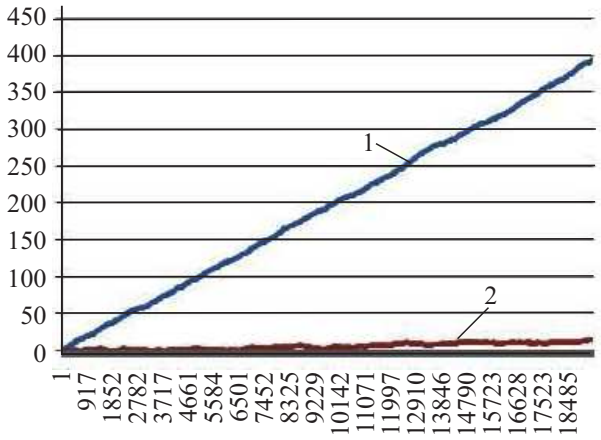

Рис. 5. Зависимости количества заданий в очереди менеджера от времени: 1) $N = 30$, 2) $N = 25$

На рис. 4 представлены графики зависимости количества пользовательских заданий, находящихся в грид (в очереди к ресурсам), от времени при $N_{\text{сум}} = 90$ и $N_{\text{сум}} = 110$ соответственно. Как видно из графиков, количество ВМ меньше 100 приводит к перегрузке системы и бесконечному накоплению очереди. В дальнейших экспериментах будем использовать полученные результаты: для потока с интенсивностью $\lambda_{\text{потока}} = 200$ количество ВМ в распределенной системе будем брать равным $N_{\text{сум}} = 100$.

В предыдущей серии экспериментов был использован идеализированный метапланировщик, который равномерно распределяет задания по ВМ с нулевым временем на обработку (принятие решения, накладные расходы на подготовку данных, пересылку данных и т.д.). Рассмотрим более реальную ситуацию,

когда менеджеру системы управления требуется определенное время на обработку заданий в очереди. Предположим, что интенсивность обработки пользовательских заданий менеджером системы управления первого уровня зависит от количества объектов управления (ВМ) и прямо пропорциональна константе $C_1 = 1250$.

В соответствии с правилом 2: для того чтобы кластерная система под управлением менеджера первого уровня справлялась с потоком указанной интенсивности, должно выполняться условие $N_{\text{ВМ}} = 25$. Отметим, что непосредственно на кластер поступает поток заданий с интенсивностью $\lambda_{\text{ВС}} = \dfrac{\lambda_{\text{потока}}}{N_{\text{сум}}} N_{\text{ВМ}}$. Соответственно при $N_{\text{ВМ}} = 25$ интенсивность потока, поступающего на один кластер, равна $\lambda_{\text{ВС}} = 50$.

На рис. 5 представлены графики зависимости количества пользовательских заданий в очереди менеджера первого уровня от времени при $N_{\text{ВМ}} = 30$ и $N_{\text{ВМ}} = 25$ соответственно. Как видно из графиков, несоблюдение правила 2 приводит к перегрузке системы и бесконечному накоплению очереди менеджера первого уровня.

Объединим множество кластерных систем под управлением менеджеров первого уровня в распределенную среду под управлением менеджеров старшего уровня. Исходные данные будем брать из предыдущих экспериментов: $v = 2$; $\lambda_{\text{потока}} = 200$; $C_1 = 1250$; $N_{\text{сум}} = 100$; $N_{\text{ВМ}} = 25$.

Добавляем в систему менеджер второго уровня, объединяющий в единую грид все кластерные ВС. При этом интенсивность обслуживания пользовательских заданий менеджером второго уровня равна $\mu = \dfrac{C}{k}$, где C — константа и k — количество объектов управления. В соответствии с правилом 4 получаем зависимость количества необходимых уровней иерархии от константы C. Для $C = 800$ получаем $i \geqslant 1$, т.е. при такой интенсивности обслуживания достаточно одного уровня иерархии. Для $C = 400$ необходимо $i \geqslant 2$. Выбор этих значений констант C_1 и C обусловлен тем, что менеджер М_1 должен взаимодействовать только с локальной СПО, а менеджер М_2 должен получать информацию из информационной подсистемы грид или опрашивать соседние менеджеры. Проверим экспериментально вышеприведенные утверждения с использованием имитационной модели.

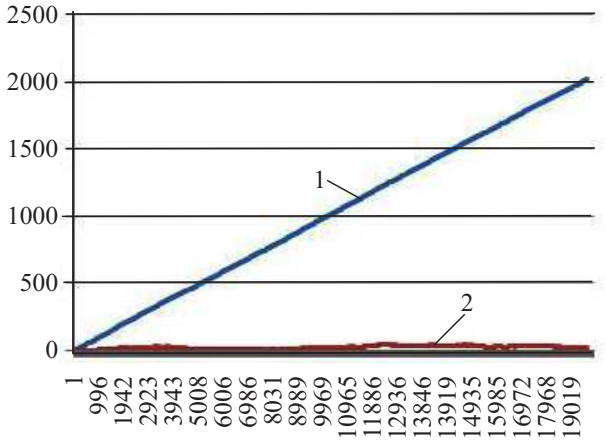

Рис. 6. Зависимости количества заданий в очереди менеджера от времени при $i = 1$: 1) $C = 400$, 2) $C = 800$

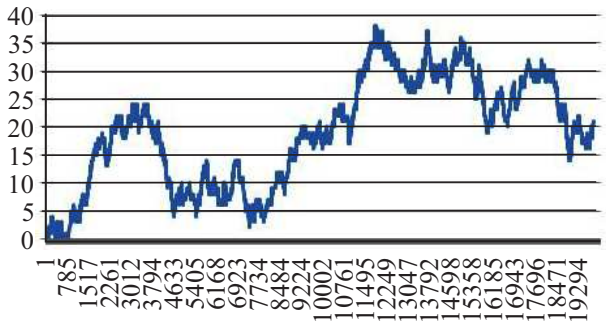

Рис. 7. Зависимость количества заданий в очереди менеджера от времени при $i = 2$, $C = 400$

Из графиков, полученных в результате имитационного моделирования, видно, что при $C = 400$ одного уровня иерархии менеджеров не достаточно (рис. 6). При $C = 400$ и $i = 1$ правило 4 не выполняется. Поэтому в соответствии с правилом 3 добавляем на уровень ниже $k \geqslant 2$ менеджеров первого уровня. Соответственно каждый из них управляет двумя кластерными системами. Проверим функционирование модели грид с двумя уровнями иерархии менеджеров второго уровня и параметром $C = 400$.

Из рис. 7 видно, что при такой архитектуре метапланировщика полученная грид справляется с поступающим потоком пользовательских заданий. При $C = 400$ и $i = 2$ правило 4 выполняется.

С целью демонстрации необходимости использования многоуровневой иерархии метадиспетчера проведем следующий эксперимент на имитационной модели: сравним, какая интенсивность обработки заданий должна быть у менеджера М_2 при двух- и трехуровневой иерархии метадиспетчера.

Итак, пусть для потока пользовательских заданий с интенсивностью $\lambda_{\text{потока}} = 800$ необходимо построить грид, способную обрабатывать данный поток. В соответствии с правилами, сформулированными

выше, получаем, что для этого необходимо как минимум $N_{\text{сум}} = 400$ вычислительных модулей с интенсивностью обработки $v = 2$. Пусть ВМ объединены в кластерные системы под управлением СПО с интенсивностью обработки поступающего потока пользовательских заданий, пропорциональной константе $C_1 = 5000$ и зависящей от количества объединяемых вычислительных ресурсов по закону $\mu = \dfrac{C_1}{N_{\text{ВМ}}}$. Тогда из правила 2 получаем $N_{\text{ВМ}} \leqslant 50$ вычислительных модулей на один кластер.

Получившиеся восемь кластерных систем объединяются в единую распределенную вычислительную среду с помощью метадиспетчера, распределяющего поступающий поток пользовательских заданий по вычислительным ресурсам.

Так как предполагается, что в отличие от кластерных систем взаимодействие между компонентами метадиспетчера, т.е. менеджерами, осуществляется через медленные каналы связи (Интернет), и для принятия управленческих решений менеджерами требуется больше информации и времени, то интенсивность обработки потоков заданий менеджерами должна быть существенно меньше, чем у М_1. Пусть $C = 1600$, тогда получается, что необходимо как минимум три уровня иерархии менеджеров метадиспетчера.

Таким образом, имеем распределенную вычислительную систему со следующими характеристиками: $v = 2$; $\lambda_{\text{потока}} = 800$; $C_1 = 5000$; $N_{\text{сум}} = 400$; $N_{\text{ВМ}} = 50$; $C = 1600$. Следовательно, интенсивность менеджера верхнего уровня равна $\mu = 800$ (такая же, как и интенсивность потока заданий); количество кластеров равно 8; количество уровней иерархии равно 3.

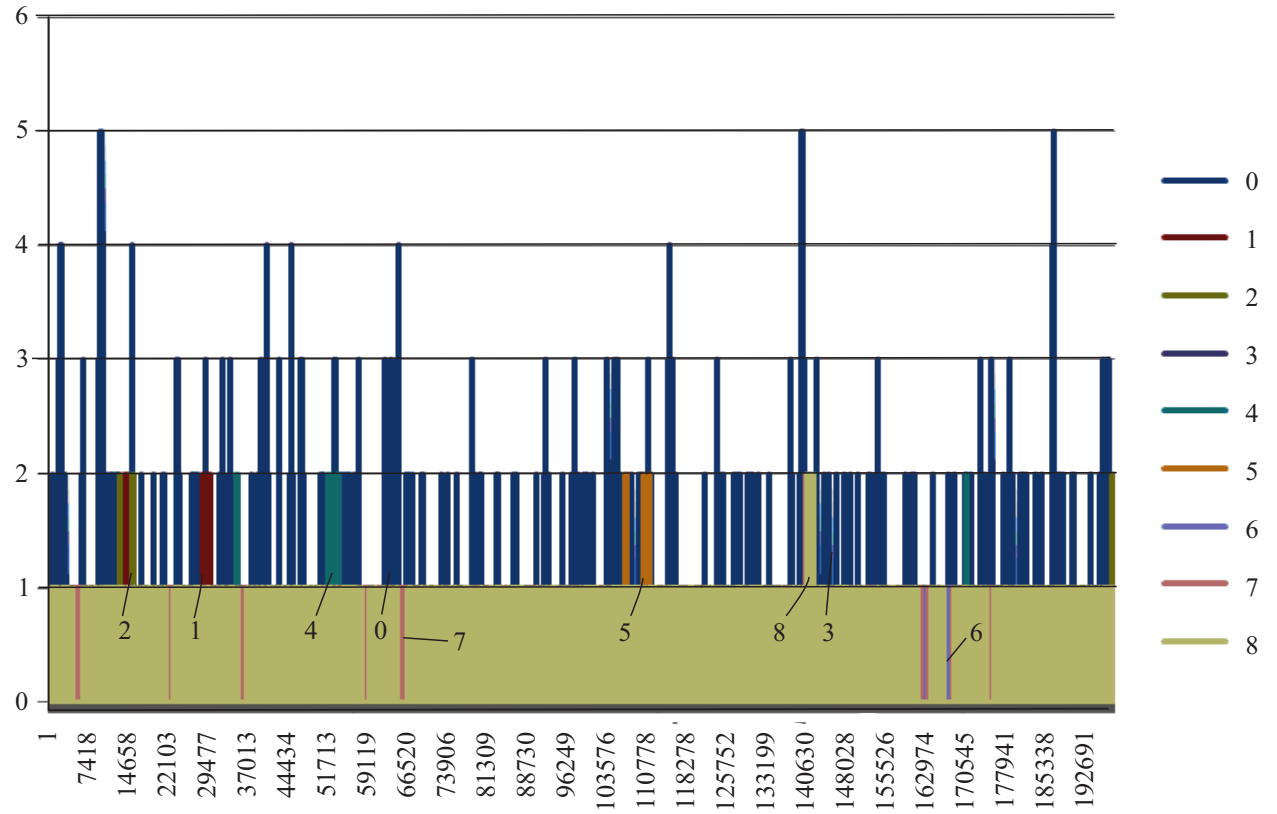

Рис. 8. Зависимости количества заданий в очереди менеджера и СПО каждой ВС от времени: 0) менеджер М_2, 1) СПО1, 2) СПО2, 3) СПО3, 4) СПО4, 5) СПО5, 6) СПО6, 7) СПО7, 8) СПО8

В результате имитационного эксперимента получаем следующую информацию о состоянии очередей СПО и менеджера самого высокого уровня. Как видно из рис. 8, поток поступающих заданий был обработан без отказов и переполнения очередей. Теперь предположим, что всеми кластерами управляет один единственный менеджер М_2. Соответственно для такого метадиспетчера будем иметь пропускную способность $\mu = \dfrac{C}{N_{\text{кластеров}}} = \dfrac{C}{8}$. В силу того, что $\rho = \dfrac{\lambda_{\text{потока}}}{\mu} \leqslant 1$, получаем $C \geqslant 8\lambda_{\text{потока}}$.

На рис. 9 представлен результат имитационного эксперимента, в котором метадиспетчер имеет интенсивность обработки $C < 8\lambda_{\text{потока}}$. Видно, что очередь к центральному менеджеру монотонно растет. По данным, выдаваемым системой моделирования, за время эксперимента в систему поступило 15 358 заданий, из них в конце эксперимента в очереди менеджера оказалось 2862 задания. Изменим параметр

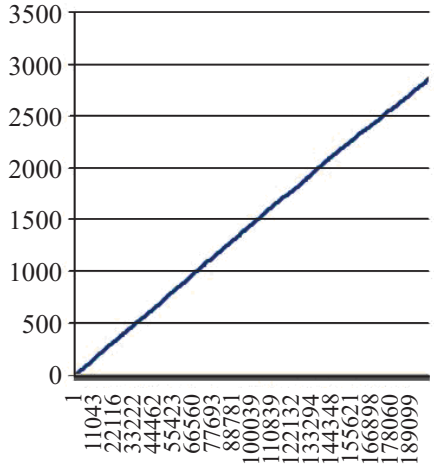

Рис. 9. Зависимость количества заданий в очереди менеджера от времени при $C = 6\lambda_{\text{потока}}$

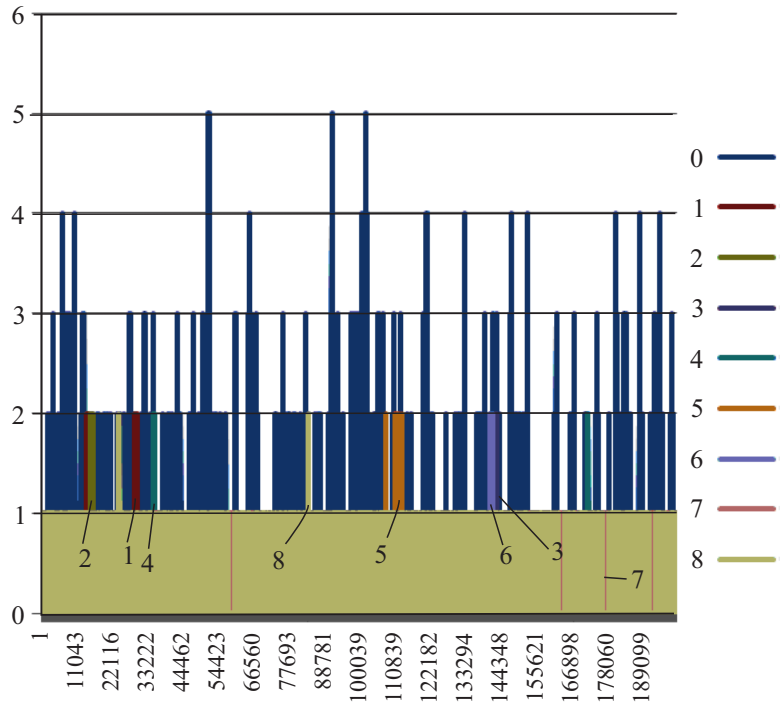

Рис. 10. Зависимости количества заданий в очереди менеджера и СПО каждой ВС от времени при $C = 8\lambda_{\text{потока}}$: 0) менеджер M_2, 1) СПО1, 2) СПО2, 3) СПО3, 4) СПО4, 5) СПО5, 6) СПО6, 7) СПО7, 8) СПО8

интенсивность обработки менеджером $C = 8\lambda_{\text{потока}}$. Результаты эксперимента показаны на рис. 10. По данным, выдаваемым системой моделирования, имеем: за время эксперимента в систему поступило 15 358 заданий, из них в конце эксперимента в очереди менеджера оказалось два задания. Таким образом, все поступившие за время моделирования задания были либо обработаны, либо распределены по вычислительным ресурсам без переполнения очередей СПО.

4. Заключение. В ходе всех экспериментов интенсивность потока пользовательских заданий оставалась неизменной. Поэтому отсутствие очередей у менеджеров говорит об адекватном выборе архитектуры метапланировщика для заданных интенсивности поступления заданий и интенсивностей их обработки в менеджерах метапланировщика и в ВМ. Таким образом, сформулированные в статье правила позволяют построить грид под управлением распределенного метапланировщика для любого потока пользовательских заданий с заданной интенсивностью.

СПИСОК ЛИТЕРАТУРЫ

1. *Корнеев В.В., Семенов Д.В.* Распределенный метапланировщик грид // Вычислительные методы и программирование. 2010. **11**, № 2. 214–221.
2. *Савин Г.И., Корнеев В.В., Шабанов Б.М., Семенов Д.В. и др.* Инфраструктура грид для суперкомпьютерных приложений // Изв. вузов. Электроника. 2011. № 1. 51–56.
3. *Palmer J., Mitrani I.* Optimal tree structures for large-scale grids. Technical Report NE1 7RU. School of Computing Science. University of Newcastle. Newcastle, 2004.
4. http://www.minutemansoftware.com/general_information.htm

Поступила в редакцию
05.05.2012