Вестник УГАТУ
Уфа : УГАТУ, 2011 Т. 15, № 2 (42). С. 99–105
УПРАВЛЕНИЕ, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАТИКА

УДК 004.7+004.051

А. В. Юлдашев

МИНИМИЗАЦИЯ ВРЕМЕНИ ВЫПОЛНЕНИЯ MPI-ПРОГРАММ С УЧЕТОМ КОНКУРЕНЦИИ ЗА КАНАЛЫ ПЕРЕДАЧИ ДАННЫХ КОММУНИКАЦИОННОЙ СРЕДЫ КЛАСТЕРНОЙ СИСТЕМЫ

В данной работе исследуется влияние конкуренции за каналы передачи данных коммуникационной среды на время выполнения MPI-программ на кластерных системах, узлы которых построены на основе многоядерных процессоров. Предлагается модель конкурентного использования каналов передачи данных. Описывается разработанный метод назначения задач (MPI-программ) на узлы кластерной системы, позволяющий сократить время выполнения программ за счет минимизации задержек, возникающих при конкурентном использовании каналов передачи данных. Представлена апробация разработанного метода назначения задач на кластерной системе УГАТУ. *Кластерная система*; *коммуникационная среда*; *многоядерный процессор*; *MPI*; *оценка времени коммуникаций*; *метод назначения задач*

ВВЕДЕНИЕ

Узлы современных кластерных вычислительных систем строятся на основе многоядерных процессоров. При выполнении параллельных программ на узлах может находиться множество процессов (потоков), конкурирующих за общие ресурсы: кэш и оперативную память, дисковую систему, а также каналы передачи данных коммуникационной среды, что негативно сказывается на эффективности выполнения программ. Для того чтобы минимизировать задержки, возникающие при использовании общих ресурсов, необходимо комплексно учитывать характеристики программ и архитектуру высокопроизводительного кластера при назначении задач на узлы вычислительной системы. Однако в существующие системы пакетной обработки и планировщики не вложены модели и алгоритмы планирования, необходимые для оптимального использования общих ресурсов многоядерных узлов.

В данной работе исследуется конкуренция за каналы передачи данных коммуникационной среды, возникающая при выполнении параллельных программ, использующих интерфейс передачи сообщений MPI (Message Passing Interface).

Проведенные экспериментальные исследования производительности коммуникационной среды Infiniband с помощью доступных тестовых программ mpi-bench-suite и OSU microbenchmarks, а также собственных тестов, позволили разработать модель конкурентного использования канала передачи данных коммуникационной среды кластерной системы, базирующуюся на модели Хокни и результатах теории массового обслуживания. Предложенная модель обеспечивает возможность оценки времени коммуникации с учетом конкуренции за каналы передачи данных при наличии известных характеристик процессов MPI-программ (количества пересылок, суммарного числа передаваемых сетевых пакетов и времени вычислений на одной итерации), а также характеристик коммуникационной среды (латентности и пиковой пропускной способности).

В целях сокращения времени выполнения MPI-программ на узлах кластерной системы с многоядерными процессорами, разработан метод назначения задач (MPI-программ), в котором учитывается загруженность коммуникационной среды и минимизируются задержки, возникающие при конкурентном использовании каналов передачи данных. Проведено экспериментальное сравнение предложенного метода с некоторыми известными методами – Best Fit и Least Utilized Node First [1], которое показало, что его использование позволяет сократить время выполнения программ на 20%.

1. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ КОНКУРЕНЦИИ ЗА КАНАЛЫ ПЕРЕДАЧИ ДАННЫХ

Кратко рассмотрим результаты проведенных экспериментов, позволивших определить набор входных параметров для модели конкурентного использования канала передачи данных коммуникационной среды Infiniband.

Эксперименты проводились на кластерной системе УГАТУ, состоящей из 266 двухпроцессорных вычислительных узлов сверхплотной компоновки IBM BladeCenter HS21 Blade Server

Контактная информация: +7(917)7741081

на базе четырехядерных процессоров Intel Xeon 5300 (Clovertown), объединенных высокопроизводительной вычислительной сетью Infiniband SDR 4x 10-Gbps.

С помощью теста коммуникационной среды transfer из пакета mpi-bench-suite были определены латентность и пропускная способность каналов передачи данных в отсутствие конкуренции при пересылке сообщений размером от 16 B до 16 MB [2]. Проведено сравнение экспериментальных данных с оценками времени коммуникации, полученными из моделей Хокни, LogP, LogGP, pLogP. Получено, что наиболее простая из рассмотренных, модель Хокни, может использоваться для оценки времени коммуникации при пересылке больших сообщений (от 128 KB) с погрешностью менее 10%.

С помощью теста osu_mbw_mr из пакета OSU microbenchmarks были определены суммарная и средняя (на одну коммуникацию) пропускные способности каналов передачи данных при однонаправленных пересылках сообщений различного размера в случае нескольких конкурирующих коммуникаций. Было показано, что средняя пропускная способность уменьшается с увеличением числа коммуникаций, и наибольшее ее снижение наблюдается при пересылке больших сообщений.

На практике в параллельных программах осуществляются как коммуникации, так и вычисления. Для того чтобы исследовать влияние конкуренции на время коммуникации при наличии в программе вычислительной составляющей, была разработана тестовая MPI-программа, процессы которой итерационно выполняли передачу сообщения определенного размера и имитацию вычислительной работы в течение заданного времени [3]. Таким образом, было исследовано влияние конкуренции на время коммуникации при передаче больших сообщений (от 128 KB до 4 MB) в зависимости от числа конкурентов и доли коммуникаций, наблюдавшейся в программе в отсутствие конкуренции.

В результате проведенных экспериментов было получено, что время коммуникации при наличии конкуренции за каналы передачи данных существенно зависит от таких параметров, как число конкурирующих процессов, количество передаваемых пакетов и время вычислений на одной итерации параллельной программы.

2. МОДЕЛЬ КОНКУРЕНТНОГО ИСПОЛЬЗОВАНИЯ КАНАЛА ПЕРЕДАЧИ ДАННЫХ

Пусть многоядерный узел кластерной системы содержит *nc* вычислительных ядер, и на нем выполняются $k = \overline{2, nc}$ независимых MPI-процессов, итерационно осуществляющих коммуникации с процессами, расположенными на других узлах кластера и вычисления. Пусть t^i_{comm} – время коммуникаций, а t^i_{sol} – известное время вычислений на одной итерации *i*-го процесса, где $i = \overline{1,k}$. Необходимо оценить время коммуникаций в условиях конкуренции за общий канал передачи данных.

Возьмем за основу модель Хокни [4], которая позволяет оценить время коммуникации без учета конкуренции по формуле

$$t_c = L + \frac{m}{B_{peak}},$$

где L – латентность, B_{peak} – пиковая пропускная способность, m – размер передаваемого сообщения.

Приведенное соотношение удобно представить в виде

$$t_c = L + \frac{m}{w} \cdot \frac{w}{B_{\text{peak}}} = L + n \cdot t_{\mu},$$

где w – размер сетевого пакета, n – число пакетов в сообщении, t_{μ} – время передачи одного пакета.

Тогда без учета конкуренции суммарное время коммуникаций на одной итерации *i*-го процесса можно оценить по формуле

$$t^i_{\text{comm}} = r^i L + N^i t_{\mu},$$

где N^i – суммарное число передаваемых пакетов, r^i – количество пересылок на одной итерации *i*-го процесса.

На практике при выполнении на узле нескольких процессов, разделяющих общий канал передачи данных, во время отправки сетевых пакетов возникают дополнительные временные задержки, среднее значение которых обозначим как t_Q. Учитывая это, оценим суммарное время коммуникаций на одной итерации *i*-го процесса по формуле

$$\tilde{t}^i_{\text{comm}} = r^i L + N^i (t_{\mu} + t_Q). \quad (1)$$

Тогда время выполнения одной итерации *i*-го процесса составит

$$t^i_{\text{iter}} = \tilde{t}^i_{\text{comm}} + t^i_{\text{sol}} = r^i L + N^i (t_{\mu} + t_Q) + t^i_{\text{sol}}. \quad (2)$$

Для нахождения t_Q процесс передачи сетевых пакетов представляется возможным моделировать с помощью открытой одноканальной системы массового обслуживания (СМО) типа M/D/1 (рис. 1) с простейшим входящим потоком заявок [5].

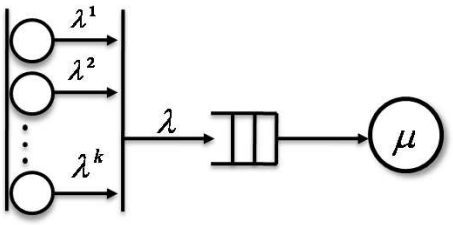

Рис. 1. Одноканальная СМО

Предполагается, что СМО содержит один обслуживающий прибор, заявки представляют собой сетевые пакеты, а источниками заявок являются процессы MPI-программ, выполняющиеся на многоядерном узле. Также предполагается, что перед прибором имеется накопитель неограниченной емкости (буфер), что означает отсутствие отказов поступающим заявкам при их постановке в очередь.

Пусть заявки поступают от *i*-го процесса с интенсивностью

$$\lambda^i = \frac{N^i}{t^i_{\text{iter}}}.$$

Введем обозначение

$$\alpha^i = \frac{r^i L + t^i_{\text{sol}}}{N^i} + t_\mu,$$

Тогда интенсивность поступления заявок от *i*-го процесса определяется как

$$\lambda^i = \frac{1}{\alpha^i + t_Q}.$$

Интенсивность входящего в СМО потока заявок λ складывается из интенсивностей образующих его потоков, следовательно

$$\lambda = \sum_{i=1}^{k} \frac{1}{\alpha^i + t_Q}. \qquad (3)$$

Предположим, что поступающие заявки обслуживаются в системе с интенсивностью $\mu = \frac{1}{t_\mu}$. Тогда из теории массового обслуживания при соблюдении условия стационарности $\frac{\lambda}{\mu} < 1$ среднее время ожидания заявки в очереди можно найти по формуле

$$t_Q = \frac{\lambda}{2\mu(\mu - \lambda)}. \qquad (4)$$

Таким образом, для нахождения среднего времени задержки t_Q требуется решить систему, образованную уравнениями (3) и (4). В свою очередь, при известном t_Q времена коммуникаций каждого процесса в условиях конкуренции за канал передачи могут быть вычислены по формуле (1).

Решение системы уравнений (3) и (4) при различных α^i не удается выписать аналитически, тем не менее, оно может быть найдено численно.

Отметим, что для нахождения времен коммуникаций в условиях конкуренции за канал передачи с помощью предложенной модели необходимо иметь характеристики процессов MPI-программ (количество пересылок, суммарное число передаваемых сетевых пакетов и время вычислений на одной итерации), а также характеристики коммуникационной среды (латентность и пиковую пропускную способность).

3. НОВЫЙ МЕТОД НАЗНАЧЕНИЯ ЗАДАЧ НА МНОГОЯДЕРНЫЕ УЗЛЫ

Анализ производительности MPI-версий ряда пакетов численного моделирования (Eclipse, Tempest More, NGT BOS, Fire Dynamics Simulator) на кластерной системе УГАТУ показал, что для достижения минимального времени выполнения программ, число процессов, распределенных на узел (*ppn*), не должно быть более двух (на рис. 2 изображено распределение 4 процессов MPI-программы с *ppn* = 1 на некотором кластере с четырехядерными узлами). Иначе возникают задержки при конкурентном доступе процессов к общим ресурсам многоядерных узлов, приводящие к увеличению времени выполнения программ.

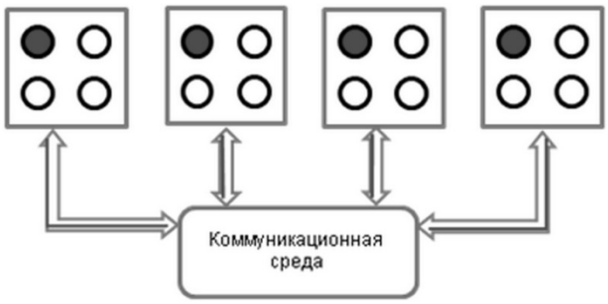

Рис.2. Распределение процессов с *ppn* = 1

Однако при таком способе распределения процессов большинство ядер на узлах простаивают. Тем не менее, по мере поступления задач на кластер можно также распределять процессы новых программ на группе ранее задействованных узлов, увеличивая количество загруженных ядер.

Тестирование показало, что даже при полной загрузке всех имеющихся на группе узлов ядер (в нашем примере для этого необходимо распределить с *ppn* = 1 процессы 4 программ), указанный способ распределения дает преиму-

щество по времени выполнения программ относительно распределения с *ppn=nc* (рис. 3).

Рис. 3. Распределение процессов с *ppn = nc*

В то же время распределение на группе узлов процессов нескольких программ с *ppn* = 1 приводит к тому, что процессы различных MPI-программ, выполняющиеся на одном узле, разделяют общий канал передачи данных и могут конкурировать при выполнении коммуникаций (рис. 4).

Рис. 4. Распределение процессов
двух программ на одной группе узлов

В целях сокращения времени выполнения MPI-программ на узлах кластерной системы с многоядерными процессорами, разработан метод назначения задач (MPI-программ), в котором учитывается загруженность коммуникационной среды и минимизируются задержки, возникающие при конкурентном использовании каналов передачи данных.

Пусть имеются MPI-программы с фиксированным количеством процессов (p) и известными характеристиками, которые требуется назначить на многоядерные узлы кластерной системы. Разобьем множество узлов на одинаковые группы, состоящие из p узлов, и получим G пронумерованных групп. При назначении MPI-программы на узлы выбирается некоторая группа, и процессы программы распределяются на каждый узел из выбранной группы с *ppn* = 1.

Необходимо определить, на какую группу узлов назначать новую программу для минимизации задержек при передаче данных, если на каждой группе уже запущено как минимум по одной программе. Для этого рассмотрим некоторую группу узлов с номером j, на которой выполняются $k^j \geq 1$ программ. Считаем, что все процессы, принадлежащие определенной программе, выполняются итерационно и имеют одинаковые известные характеристики: количество пересылок $r^{i,j}$, суммарное число передаваемых сетевых пакетов $N^{i,j}$ и время вычислений $t_{\text{sol}}^{i,j}$ на одной итерации. Здесь $i = \overline{1,k^j}$ – порядковый номер программы, запущенной на группе с номером j.

Так как характеристики процессов, принадлежащих определенной программе, одинаковы, и на всех вычислительных узлах имеется по одному каналу передачи данных, характеристики которых также одинаковы, средние времена задержки при передаче сетевых пакетов на каждом узле j-й группы будут равны. В связи с этим приведенные далее выкладки верны для любого узла из рассматриваемой группы.

Рассмотрим выполнение k^j процессов на одном из узлов выбранной группы в течение времени $T >> t_{\text{iter}}^{i,j}, \forall i$. Предполагаем, что за время T процессы выполняют $u^{i,j}$ итераций. С учетом (2) можно представить время T как

$$\begin{aligned} T &= u^{i,j}(t_{\text{sol}}^{i,j} + r^{i,j}L + N^{i,j}(t_{\mu} + t_Q)) = \\ &= u^{i,j}t_{sol}^{i,j} + u^{i,j}(r^{i,j}L + N^{i,j}t_{\mu}) + u^{i,j}N^{i,j}t_Q = \\ &= T_{\text{sol}}^{i,j} + T_{\text{transf}}^{i,j} + T_Q^{i,j}, \end{aligned}$$

где $T_{\text{sol}}^{i,\text{j}}$ – суммарное время вычислений, $T_{\text{transf}}^{i,j}$ – суммарное время, затрачиваемое непосредственно на коммуникации, а $T_Q^{i,j}$ $T_{\text{sol}}^{i,\text{j}}$ – суммарное время задержек при коммуникациях, возникающих в результате использования общего канала передачи данных процессами различных программ.

Так как на каждом узле выполняется k^j процессов и группа содержит p узлов, верно

$$k^j pT = \sum_{i=1}^{k^j} p(T_{\text{sol}}^{i,j} + T_{\text{transf}}^{i,j} + T_Q^{i,j}), \forall j = \overline{1,j}\,.$$

Складывая приведенное соотношение по всем G группам, получим

$$\sum_{j=1}^{G} k^j pT = \sum_{j=1}^{G} p\left(\sum_{i=1}^{k^j} T_{\text{sol}}^{i,j} + \sum_{i=1}^{k^j} T_{\text{transf}}^{i,j} + \sum_{i=1}^{k^j} T_Q^{i,j}\right),$$

откуда следует

$$\frac{\sum_{j=1}^{G}\sum_{i=1}^{k^j} T_{\text{sol}}^{i,j}}{\sum_{j=1}^{G} k^j T} + \frac{\sum_{j=1}^{G}\sum_{i=1}^{k^j} T_{\text{transf}}^{i,j}}{\sum_{j=1}^{G} k^j T} + \frac{\sum_{j=1}^{G}\sum_{i=1}^{k^j} T_Q^{i,j}}{\sum_{j=1}^{G} k^j T} = 1\,.$$

Слагаемые в левой части равенства могут принимать значения от 0 до 1 в зависимости от характеристик MPI-программ. Последнее слагаемое, которое обозначим f_Q и будем называть долей задержек, показывает, какую часть времени задействованные ядра кластерной системы простаивают из-за конкуренции за каналы передачи данных. Таким образом, величина доли задержек влияет на эффективность загрузки кластерной системы: чем она меньше, тем эффективнее используются вычислительные ресурсы.

Доля задержек может быть выражена следующим образом:

$$f_Q = \frac{1}{\sum_{j=1}^{G} k^j} \cdot \sum_{j=1}^{G} \sum_{i=1}^{k^j} \frac{t_Q^j}{\alpha^{i,j} + t_Q^j},$$

где $\alpha^{i,j} = \frac{r^{i,j} L + t_{sol}^{i,j}}{N^{i,j}} + t_\mu$, а t_Q^j – среднее время задержки при передаче сетевых пакетов на j-й группе. Далее, учитывая (3), получим

$$f_Q = \frac{1}{\sum_{j=1}^{G} k^j} \cdot \sum_{j=1}^{G} \lambda^j t_Q^j,$$

где λ^j – интенсивность потока сетевых пакетов на вычислительных узлах j-й группы.

Отметим, что в полученном представлении f_Q отсутствуют $u^{i,j}$, следовательно, для нахождения доли задержек достаточно владеть характеристиками коммуникационной среды и процессов MPI-программ на одной итерации.

При назначении новой программы на группу с номером l будем иметь новую долю задержек:

$$\tilde{f}_Q = \frac{1}{\sum_{j=1}^{G} k^j + 1} \left(\sum_{\substack{j=1 \\ j \neq l}}^{G} \lambda^j t_Q^j + \tilde{\lambda}^l \tilde{t}_Q^l \right).$$

С целью повышения эффективности использования кластера необходимо определить, при каком l будет минимальна новая доля задержек или, что эквивалентно, изменение доли задержек:

$$\tilde{f}_Q - f_Q = \frac{-\sum_{j=1}^{G} \lambda^j t_Q^j}{\left(\sum_{j=1}^{G} k_j + 1 \right) \sum_{j=1}^{G} k_j} + \frac{\left(\tilde{\lambda}^l \tilde{t}_Q^l - \lambda^l t_Q^l \right)}{\sum_{j=1}^{G} k_j + 1}. \quad (5)$$

Отсюда следует, что оптимально назначить новую программу на группу с номером l, для которой

$$\delta^l = \left(\tilde{\lambda}^l \tilde{t}_Q^l - \lambda^l t_Q^l \right) \to \min_l. \quad (6)$$

Причем для нахождения λ^l и t_Q^l необходимо решить систему уравнений

$$\begin{cases} \lambda^l = \sum_{i=1}^{k_l} \frac{1}{\alpha^{i,l} + t_Q^l} \\ t_Q^l = \frac{\lambda^l}{2\mu(\mu - \lambda^l)}, \end{cases} \quad (7)$$

а для нахождения $\tilde{\lambda}^l$ и $\tilde{t}_Q^l$ – систему

$$\begin{cases} \tilde{\lambda}^l = \sum_{i=1}^{k_l} \frac{1}{\alpha^{i,l} + \tilde{t}_Q^l} + \frac{1}{\alpha^{K_l+1,l} + \tilde{t}_Q^l} \\ \tilde{t}_Q^l = \frac{\tilde{\lambda}^l}{2\mu(\mu - \tilde{\lambda}^l)}. \end{cases} \quad (8)$$

Таким образом, для выбора оптимальной группы, необходимо G раз решить системы (7), (8) и выбрать такую группу, для которой δ^l минимальна.

Отметим, что если до назначения новой программы на группе узлов выполнялась только одна программа, конкуренция за каналы передачи данных на узлах данной группы отсутствовала, следовательно, можно положить f_Q равной нулю.

4. ВЫБОР ГРУППЫ УЗЛОВ ДЛЯ НАЗНАЧЕНИЯ НОВОЙ ЗАДАЧИ В УПРОЩЕННОЙ ПОСТАНОВКЕ

Решение поставленной задачи (6) в общем случае может быть найдено только численно. Тем не менее, в упрощенной постановке – при условии, что параметр $\alpha^{i,j}$, $\forall\ i, j$ принимает не более двух различных значений, минимизируемая функция из (6) может быть выражена аналитически.

Случай 1. Характеристики процессов всех программ таковы, что $\alpha^{i,j} = \alpha$, $\forall\ i, j$.

В этом случае системы уравнений (7) и (8) могут быть сведены к квадратным уравнениям, откуда выписываются единственные корни $\lambda^l = \lambda^l(\mu, \alpha, k^l)$ и $\tilde{\lambda}^l = \tilde{\lambda}^l(\mu, \alpha, k^l)$, удовлетворяющие условию стационарности.

На рис. 5 представлен график зависимости функции $\delta^l = \delta^l(\mu, \alpha, k^l)$ от α на отрезке [0,38 × × 10^{-5}…3,8 · 10^{-5}] при $\mu = 2^{19}$ и ряде фиксированных $k^l = \overline{1,7}$.

Видно, что при любом фиксированном значении α, минимальное значение δ^l достигается при минимальном значении k^l. Другими словами, если на узлы кластера назначаются программы, процессы которых характеризуются одинаковыми значениями параметра α, то при выборе группы узлов для назначения новой

программы можно руководствоваться только значением параметра k^l. Таким образом, с точки зрения загрузки каналов передачи данных, оптимально будет назначить новую программу на группу, где выполняется наименьшее число программ. Также из рис. 5 следует, что при фиксированном значении k^l минимальное значение δ^l достигается при максимальном значении α.

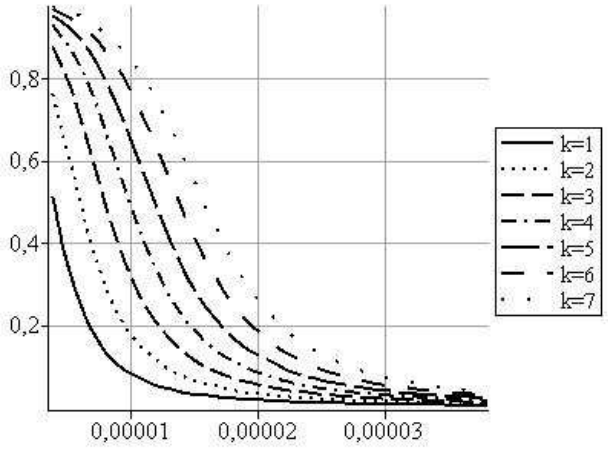

Рис. 5. График зависимости изменения доли задержки от α

Случай 2. Характеристики процессов всех программ до назначения новой программы таковы, что $\alpha^{i,j} = \alpha_1$, $\forall\, i, j$, характеристики процессов новой программы описываются параметром α_2.

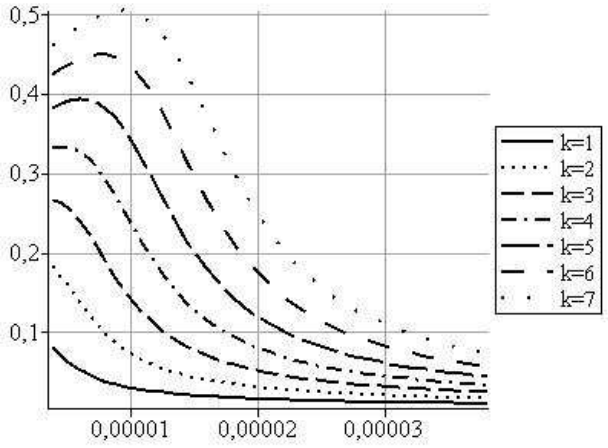

Рис. 6. График зависимости изменения доли задержки от α_1

В этом случае система (7) сводится к квадратному уравнению, а (8) – к кубическому, откуда явно выписываются единственные корни $\lambda^l = \lambda^l(\mu, \alpha_1, k^l)$ и $\tilde{\lambda}^l = \tilde{\lambda}^l(\mu, \alpha_1, \alpha_2, k^l)$, удовлетворяющие условию стационарности.

На рис. 6 представлен график зависимости функции $\delta^l = \delta^l(\mu, \alpha_1, \alpha_2, k^l)$ от α_1 на отрезке $[0{,}38 \cdot 10^{-5} \ldots 3{,}8 \cdot 10^{-5}]$ при $\alpha_2 = 2{,}08 \cdot 10^{-5}$, $\mu = 2^{19}$ и ряде фиксированных $k^l = \overline{1,7}$.

Получено, что при любом фиксированном α_1, минимальное значение δ^l достигается при минимальном значении k^l, аналогично предыдущему случаю. В то же время при ряде $k^l \geq 4$ зависимость δ^l от α_1 становится немонотонной.

Например, пусть имеется три группы узлов, на которых запущено по 6 программ. Причем на первой группе для всех программ $\alpha_1 = 0{,}5 \cdot 10^{-5}$, на второй – $\alpha_1 = 0{,}7 \cdot 10^{-5}$, а на третьей – $\alpha_1 = 2 \cdot 10^{-5}$. С точки зрения загрузки каналов передачи данных наиболее загруженной группой является первая, а наименее загруженной – третья. Из рис. 6 видно, что оптимальным является назначение новой программы на третью группу. В то же время, оказывается, выгоднее назначить новую программу на первую группу, с наибольшей загруженностью каналов передачи данных, чем на вторую, со средней загруженностью, так как назначение программы на вторую группу приведет к большему увеличению δ^l и, следовательно, доли задержек f_Q.

Таким образом, при наличии программ, характеристики которых описываются двумя и более различными α, выбор оптимальной группы для назначения новой программы не представляется возможным априори, без решения систем (7) и (8).

5. АПРОБАЦИЯ РАЗРАБОТАННОГО МЕТОДА НАЗНАЧЕНИЯ ЗАДАЧ НА МНОГОЯДЕРНЫЕ УЗЛЫ КЛАСТЕРА

Проведено экспериментальное сравнение разработанного метода (В) с некоторыми известными методами назначения задач на узлы: Best Fit (А) и Least Utilized Node First (Б). Производилось назначение тестовых MPI-программ, с числом процессов $p = 8$, в которых на каждой итерации выполнялся вызов функции MPI_Alltoall с размером сообщения m, а также имитация вычислений в течение 300 мс. Рассмотрено два типа программ с m равным 128 КВ и 1 МВ. Проведено три серии экспериментов, в которых на шестнадцать вычислительных узлов кластера назначалось по 2^k программ каждого типа, где k – порядковый номер эксперимента. На рис. 7 представлены суммарные времена выполнения программ, полученные в результате экспериментов.

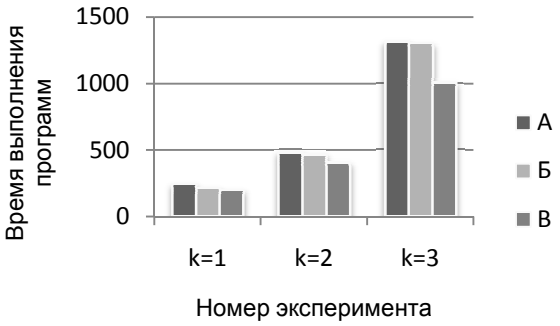

Рис. 7. Сравнение различных методов назначения задач на многоядерные узлы кластера

Во всех трех экспериментах применение разработанного метода (В) позволило сократить суммарное время выполнения программ на 6–20% по сравнению с методами А и Б. Сокращение времени было достигнуто за счет оптимального использования каналов передачи.

ЗАКЛЮЧЕНИЕ

Планируется апробация разработанного метода назначения задач на реальных приложениях для его дальнейшего внедрения в кластерный планировщик, разрабатываемый в УГАТУ, в целях более эффективного использования ресурсов вычислительного кластера.

В дальнейшем целесообразно исследовать возможность синтеза моделей конкурентного использования каналов передачи данных и конкурентного доступа к памяти для многоядерных систем. Это позволит оценить задержки, возникающие при выполнении широкого класса MPI-программ, для которых может иметь место конкуренция как за каналы передачи данных, так и при доступе к иерархии памяти. Также необходима разработка нетрудоемких алгоритмов планирования, учитывающих влияние конкуренции за общие ресурсы многоядерных узлов на эффективность выполнения параллельных программ.

СПИСОК ЛИТЕРАТУРЫ

1. **Полежаев П. Н.** Исследование алгоритмов планирования параллельных задач для кластерных вычислительных систем с помощью симулятора // ПаВТ’2010: Тр. межд. науч. конф. Челябинск: ЮУрГУ, 2010. С. 287–298.

2. **Халиуллина М. Р., Юлдашев А. В.** Тестирование коммуникационной среды суперкомпьютера УГАТУ для решения задачи балансировки нагрузки // ПаВТ’2009: Тр. межд. науч. конф. Челябинск: ЮУрГУ, 2009. С. 826.

3. **Юлдашев А. В.** Балансировка нагрузки на основе сети в рамках программного комплекса автоматизированных расчетов на кластерных системах // Актуальные проблемы в науке и технике: Сб. труд. IV всероссийск. зимн. шк.-сем. асп. и мол. ученых. Уфа: Диалог, 2009. Т. 1. С. 573–577.

4. **Hockney R. W.** The Communication Challenge for MPP: Intel Paragon and Meiko CS-2 // Parallel Computing, North-Holland. 1994. Vol. 20. P. 389–398.

5. **Клейнрок Л.** Теория массового обслуживания. М.: Машиностроение, 1979. 432 с.

ОБ АВТОРЕ

Юлдашев Артур Владимирович, мл. науч. сотр. ИКИ при НИЧ, асс. каф. ВВТиС. Дипл. инж.-мат. (УГАТУ, 2006). Готовит диссертацию в области эффективного использования ресурсов кластерных вычислительных систем.