

УДК 004.457+004.031.2+004.382.2

DOI: 10.15827/2311-6749.17.3.8

ЭКСПЕРИМЕНТАЛЬНЫЕ ОЦЕНКИ ВЛИЯНИЯ ПАКЕТИРОВАНИЯ НА НЕКОТОРЫЕ ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ ПЛАНИРОВАНИЯ ПАРАЛЛЕЛЬНЫХ ЗАДАНИЙ

А.В. Баранов, к.т.н., доцент, ведущий научный сотрудник, Anton.Baranov@jssc.ru;

Д.С. Ляховец, стажер-исследователь, anetto@inbox.ru

(Межеведомственный суперкомпьютерный центр РАН – филиал ФГУ

«ФНЦ Научно-исследовательский институт системных исследований Российской академии наук», просп. Ленинский, 32а, г. Москва, 119334, Россия)

Статья посвящена вопросам объединения в пакеты однотипных заданий с длительным временем инициализации. Длительная инициализация влечет за собой снижение эффективности использования вычислительных ресурсов и планирования параллельных заданий. Авторами экспериментально исследована зависимость некоторых показателей эффективности планирования параллельных заданий от доли времени инициализации и параметров пакетирования. В статье приводятся результаты экспериментов, демонстрирующие влияние пакетирования на такие показатели эффективности, как полная и полезная загрузка вычислительных ресурсов, среднее время ожидания задания в очереди и среднее время обслуживания задания.

Ключевые слова: *пакетирование параллельных заданий, система управления заданиями, показатели эффективности планирования заданий.*

В настоящей работе под системой высокопроизводительных вычислений будем понимать вычислительную установку типовой кластерной архитектуры, состоящую из нескольких *вычислительных модулей* (ВМ), объединенных одной или несколькими высокоскоростными сетями. ВМ такой системы представляет собой самостоятельный многопроцессорный компьютер [1], часто оснащаемый сопроцессорами – ускорителями на базе графических процессоров, *программируемых логических интегральных схем* (ПЛИС) или многоядерных мультитредовых решений (Intel Xeon Phi). Как правило, подобные вычислительные установки функционируют в режиме коллективного пользования, в котором прикладные программы оформляются в виде специальных пользовательских заданий, поступающих на вход общей очереди. Жизненный цикл задания включает следующие этапы:

- нахождение задания в очереди;
- инициализация задания;
- обработка задания на вычислительной установке;
- завершение обработки и сохранение результатов.

Выделим несколько категорий заданий, снижающих эффективность использования вычислительных ресурсов из-за длительного времени инициализации:

- для задания требуется инициализация высокопроизводительного ускорителя (GPU или ПЛИС), время инициализации ускорителя сравнимо со временем обработки задания [2];
- для задания необходима специфическая виртуальная программная платформа, развертывание ее перед стартом задания может занять существенное время [3, 4];
- для задания необходимо до начала обработки загрузить на ВМ значительный объем входных данных.

Задания перечисленных категорий объединяет длительное относительно остальных этапов жизненного цикла время инициализации, которое может включать в себя время перепрограммирования ПЛИС, компиляции программы на GPU, запуска виртуальных машин, подготовки контейнеров, копирования данных и т.п. При инициализации могут быть задействованы CPU, оперативная память и сеть, так что, даже интеллектуальная биллинговая система не способна достоверно отличить инициализацию задания от процесса обработки. В то же время потери времени на инициализацию равнозначны потерям эффективности использования ресурсов.

Идея пакетирования состоит в том, что во входном потоке можно выделить задания, для инициализации которых необходимо совершить одинаковые действия. Будем называть задания с одной и той же процедурой инициализации заданиями одинакового типа. Доля затраченного на инициализацию времени может быть уменьшена за счет однократной инициализации перед выполнением нескольких заданий одного типа, объединяемых в пакет (метазадание).

Интуитивное представление подсказывает, что пакетирование должно положительно повлиять на эффективность. Однако при планировании в современных системах управления заданиями, таких как СУППЗ [5], SLURM [6], Moab [7], используются достаточно сложные алгоритмы (например обратное за-

полнение), что не позволяет однозначно спрогнозировать влияние пакетирования на эффективность планирования заданий. Например, в работе [8] рассмотрен квазипланировщик, поставляющий в очередь СУППЗ специальным образом подобранные задания. Подбор заданий осуществлялся с учетом особенностей работы алгоритма обратного заполнения, и использование квазипланировщика, по интуитивным предположениям авторов, должно было положительно повлиять на загрузку вычислителя. Однако результаты проведенных экспериментов [8] не позволили выявить сколь-либо существенное влияние квазипланировщика на показатели эффективности СУППЗ.

Настоящая статья является продолжением работы [9], в которой авторы исследовали влияние пакетирования на загрузку (утилизацию) вычислителя для случая планирования идеально масштабируемых заданий. Время выполнения таких заданий уменьшается прямо пропорционально объему выделенных вычислительных ресурсов. Например, если время обработки задания на 1 ВМ составляет 30 мин., тогда на 2 ВМ составит 15 мин., на 30 ВМ – 1 мин.

Время инициализации для идеально масштабируемых заданий не масштабируется и является константой для задания. Так, если время инициализации составляет 10 мин., время обработки задания на 1 ВМ – 20 мин. (суммарное время выполнения 30 мин.), то на 2 ВМ время выполнения составит $10 + (20/2) = 20$ мин., на 30 ВМ – $10 + (20/30) = 10,67$ мин. и т.д.

Исследуемые показатели эффективности планирования заданий

Пусть изменения в системе (добавление заданий, изменение количества доступных ВМ и др.) происходят в моменты времени t_i . Разобьем все время работы системы на промежутки времени: $\Delta t_i = t_{i+1} - t_i$.

Под модуль-часом будем понимать вычислительный ресурс, эквивалентный 1 ВМ, доступному для обработки задания в течение 1 часа. 1 модуль-час может означать, что 1 ВМ работал 1 час или 2 ВМ работали по 0,5 часа. Пусть n_i – число работоспособных ВМ в i -й момент времени, m_i – число занятых выполнением заданий ВМ в i -й момент времени. К выполнению задания относятся инициализация и обработка задания. Пусть также u_i – число занятых обработкой заданий ВМ в i -й момент времени. Занятые инициализацией задания модули не учитываются в u_i .

Определим как S_{all} число модуль-часов, доступных для выполнения заданий за определенный временной период:

$$S_{\text{all}}(t) = \sum_{i=0}^t n_i \Delta t_i.$$

Определим как S_{work} количество модуль-часов, использованных для выполнения заданий за тот же период:

$$S_{\text{work}}(t) = \sum_{i=0}^t m_i \Delta t_i, \text{ где } m_i \in (0, n_i),$$

а как S_{useful} – количество модуль-часов, использованных для обработки заданий за тот же период (без учета инициализации):

$$S_{\text{useful}}(t) = \sum_{i=0}^t u_i \Delta t_i, \text{ где } u_i \in (0, n_i).$$

Пусть q_i – время ожидания i -го задания в очереди с момента постановки в очередь до начала его выполнения, v_i – реальное время выполнения i -го задания (включает в себя инициализацию и обработку).

Приведем ряд известных показателей эффективности, влияние пакетирования на которые будет исследовано в настоящей статье.

1. Полезная загрузка вычислительных ресурсов U за определенный период времени, которая может быть определена как

$$U(t) = \frac{S_{\text{useful}}(t)}{S_{\text{all}}(t)}.$$

2. Среднее время ожидания задания в очереди Q . Для k заданий этот показатель может быть рассчитан как

$$Q = \frac{\sum_{i=0}^k q_i}{k}.$$

3. Среднее время полного обслуживания задания F . Для k заданий формула имеет вид:

$$F = \frac{\sum_{i=0}^k (q_i + v_i)}{k}.$$

Параметры входного потока заданий и подсистемы формирования пакетов

Для проведения экспериментов авторами был использован стенд, рассмотренный в работе [9]. Стенд включает в себя симулятор СУППЗ и подсистему пакетирования заданий. Входной поток тестовых заданий формировался аналогично работе [9] таким образом, чтобы обеспечить загрузку вычислительных ресурсов на уровне 95–100 % и постоянное наличие заданий в очереди. Меньшая загрузка нецелесообразна, поскольку ресурсы заведомо будут простаивать, большая загрузка приведет к бесконечной очереди заданий.

Этапу обработки задания предшествует инициализация, время которой обозначим как $t_{\text{иниц}}$. Если e_i – реальное время обработки i -го задания, то отношение

$$O_i = \frac{t_{\text{иниц}}}{e_i}$$

можно определить, как накладные расходы на инициализацию i -го задания. В проводимых авторами экспериментах накладные расходы на инициализацию были определены в качестве параметра входного потока заданий и в зависимости от эксперимента составляли 10, 25, 50 и 100 % от среднего времени обработки задания. При этом использовался следующий алгоритм формирования пакетов.

Пусть в формировании пакета участвуют n идеально масштабируемых заданий с временем обработки на 1 VM $e_1, e_2, \dots, e_i, \dots, e_n$ и временем инициализации $t_{\text{иниц}}$. Тогда общее время обработки на 1 VM пакета из n заданий составит некоторую величину $e_{\text{пакета}}^1$:

$$e_{\text{пакета}}^1 = \sum_{i=1}^n e_i.$$

Соответственно, время $e_{\text{пакета}}^m$ выполнения пакета на m VM составит:

$$e_{\text{пакета}}^m = \sum_{i=1}^n \frac{e_i}{m}.$$

Время $v_{\text{пакета}}^m$ выполнения пакета на m VM составит:

$$v_{\text{пакета}}^m = t_{\text{иниц}} + \sum_{i=1}^n \frac{e_i}{m}.$$

При формировании пакета система пакетирования должна принять два решения: какое число n заданий войдет в очередной пакет и какое число m VM следует выделить для обработки пакета. Главным параметром при принятии этих решений является т.н. порог целесообразности k , определяющий отношение времени обработки пакета заданий ко времени инициализации пакета. Другими словами, порог целесообразности k говорит о том, что время $e_{\text{пакета}}^m$ обработки пакета на m VM должно не менее чем в k раз превышать время инициализации пакета $t_{\text{иниц}}$:

$$e_{\text{пакета}}^m > k \cdot t_{\text{иниц}}.$$

Пусть в некоторый момент времени в системе присутствуют $n - 1$ заданий определенного типа, наличествует m свободных VM и на вход поступает следующее задание n того же типа. Подсистема пакетирования примет положительное решение о формировании пакета и его направлении на обработку на m VM, если выполняется условие

$$m < \frac{\sum_{i=1}^n e_i}{t_{\text{иниц}} \cdot k}.$$

и в вычислительной системе наличествует $m_{\text{своб}} \geq m$ свободных VM. Если $m_{\text{своб}} < m$, пакет заданий будет направлен на обработку на $m_{\text{своб}}$ VM.

В отличие от работы [9], где авторы задавали значение порога целесообразности больше или равным 1 и при этом не выявили его заметного влияния на полезную загрузку вычислителя, в настоящей работе были исследованы значения порога целесообразности, меньшие 1.

Кроме малых величин порога целесообразности, авторы в настоящей работе исследовали зависимость показателей эффективности от числа типов заданий: эксперименты проводились с 1, 4 и 8 типами заданий.

Результаты проведенных экспериментов

Эксперименты проводились авторами с многократными повторениями, в ходе которых результаты измерений рассматривались как случайная величина. При осреднении шкала измерений разбивалась на диапазоны, среднее значение по диапазонам также рассматривалось как случайная величина. Таким образом, многочисленные повторы экспериментов давали в качестве итога ряд значений случайной вели-

чины, для которого рассчитывался коэффициент вариации. В ходе проведения экспериментов коэффициент вариации не превысил 10 %, что говорит о достаточной стабильности полученных результатов.

Особенность алгоритма формирования пакетов состоит в скачкообразном выделении вычислительных ресурсов пакетированным заданиям. На рисунке 1 показана зависимость среднего числа выделяемых пакетам заданий ВМ от величины порога целесообразности, накладных расходов на инициализацию для 8 типов заданий. Накладные расходы изменялись в процессе экспериментов от 10 до 100 %, каждое значение показано на рисунке отдельной кривой.

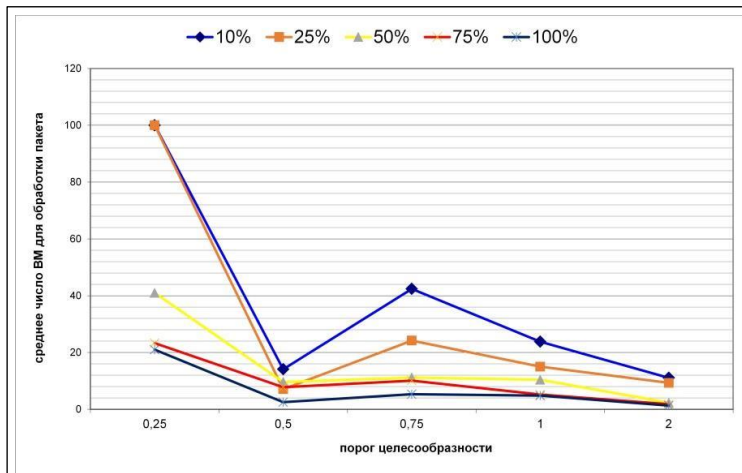


Рис. 1. Зависимость среднего числа выделяемых пакетам заданий ВМ от величины порога целесообразности (от 0,25 до 2) и накладных расходов на инициализацию (от 10 до 100 %) для 8 типов заданий

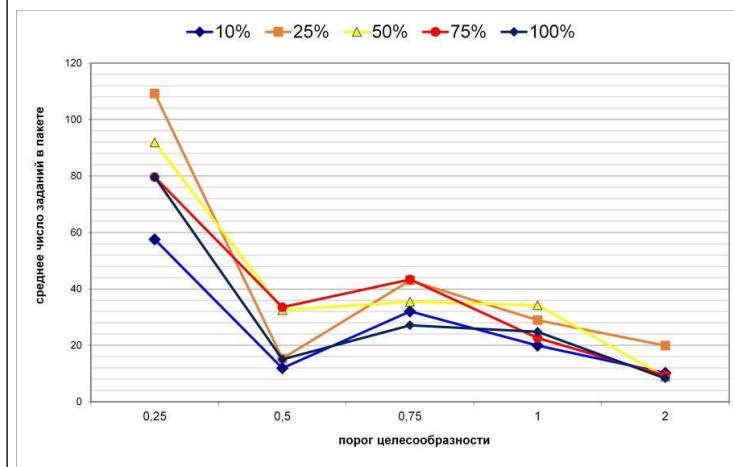


Рис. 2. Зависимость среднего числа заданий в пакете от величины порога целесообразности (от 0,25 до 2) и накладных расходов на инициализацию (от 10 до 100 %) для 8 типов заданий

ную загрузку вполне заметно, на рисунке 3 видно, что положительный эффект от применения пакетирования может быть достигнут при значениях порога целесообразности 0,5 и выше, но, начиная со значения 1, влияние порога целесообразности на полезную загрузку вычислителя сходит на нет. При накладных расходах в 10 % заметна зависимость полезной загрузки от числа типов заданий – с увеличением числа типов на загрузку большее влияние оказывает порог целесообразности.

Рисунок 4 демонстрирует любопытный результат, свидетельствующий о положительном влиянии пакетирования на такой показатель эффективности, как среднее время ожидания задания в очереди. Подобный результат был неочевиден при начале исследований, поскольку задания неизбежно задерживаются в очереди на время формирования пакета. Тем не менее, стабильный положительный эффект по

Как видно из рисунка 1, число выделяемых пакету заданий ресурсов снижается при увеличении порога целесообразности до 0,5, затем повышается при 0,75 и стабилизируется при значении 2. Чем больше накладные расходы, тем меньше в итоге ресурсов выделяется для сформированного пакета заданий. Это логично, поскольку с точки зрения системы пакетирования большие объемы выделенных ресурсов должны проходить инициализацию как можно быстрее.

На рисунке 2 показано среднее число заданий, помещаемых в один сформированный пакет. С ростом порога целесообразности число заданий в пакете уменьшается (система пакетирования готова выделить для пакета все меньшее число ВМ (рис. 1)), равно как и расхождение среднего числа заданий при разных накладных расходах (при пороге целесообразности в 0,25 наблюдается расхождение почти на порядок, при пороге в 2 – лишь двукратное расхождение).

Рисунок 3 демонстрирует зависимость полезной загрузки вычислителя от величины порога целесообразности, накладных расходов на инициализацию и числа типов заданий. На рисунке также отмечено значение полезной загрузки при отсутствии системы пакетирования. Видно, как и в работе [9], что положительный эффект от пакетирования ощутим уже при накладных расходах в 10 %, увеличиваясь с их ростом. С 50 % расходов прекращается влияние на полезную загрузку величины порога целесообразности. При малых накладных расходах (что на практике характерно для большинства заданий) влияние порога целесообразности на полез-

сравнению с ситуацией отсутствия пакетирования достигается, начиная с накладных расходов в 10 % на инициализацию и значения порога целесообразности в 0,5 и выше. Аналогичный эффект можно наблюдать на рисунке 5 для полного времени обслуживания задания, включающего как время ожидания задания в очереди, так и время его обработки на вычислителе. Как для среднего времени ожидания задания в очереди, так и для среднего времени полного обслуживания задания характерно уменьшение влияния числа типов заданий с ростом накладных расходов на инициализацию.

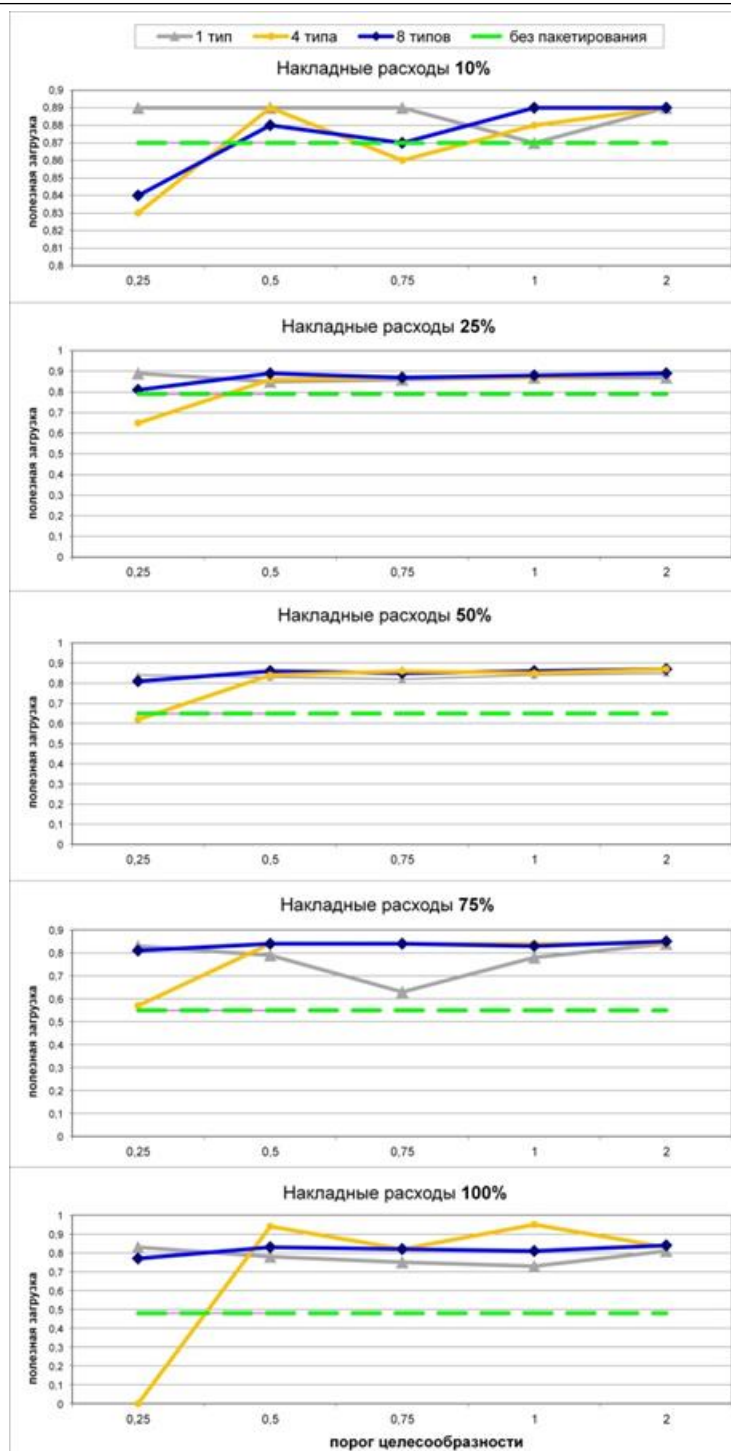


Рис. 3. Зависимость полезной загрузки вычислителя от величины порога целесообразности (от 0,25 до 2), накладных расходов на инициализацию (от 10 до 100 %) и числа типов заданий

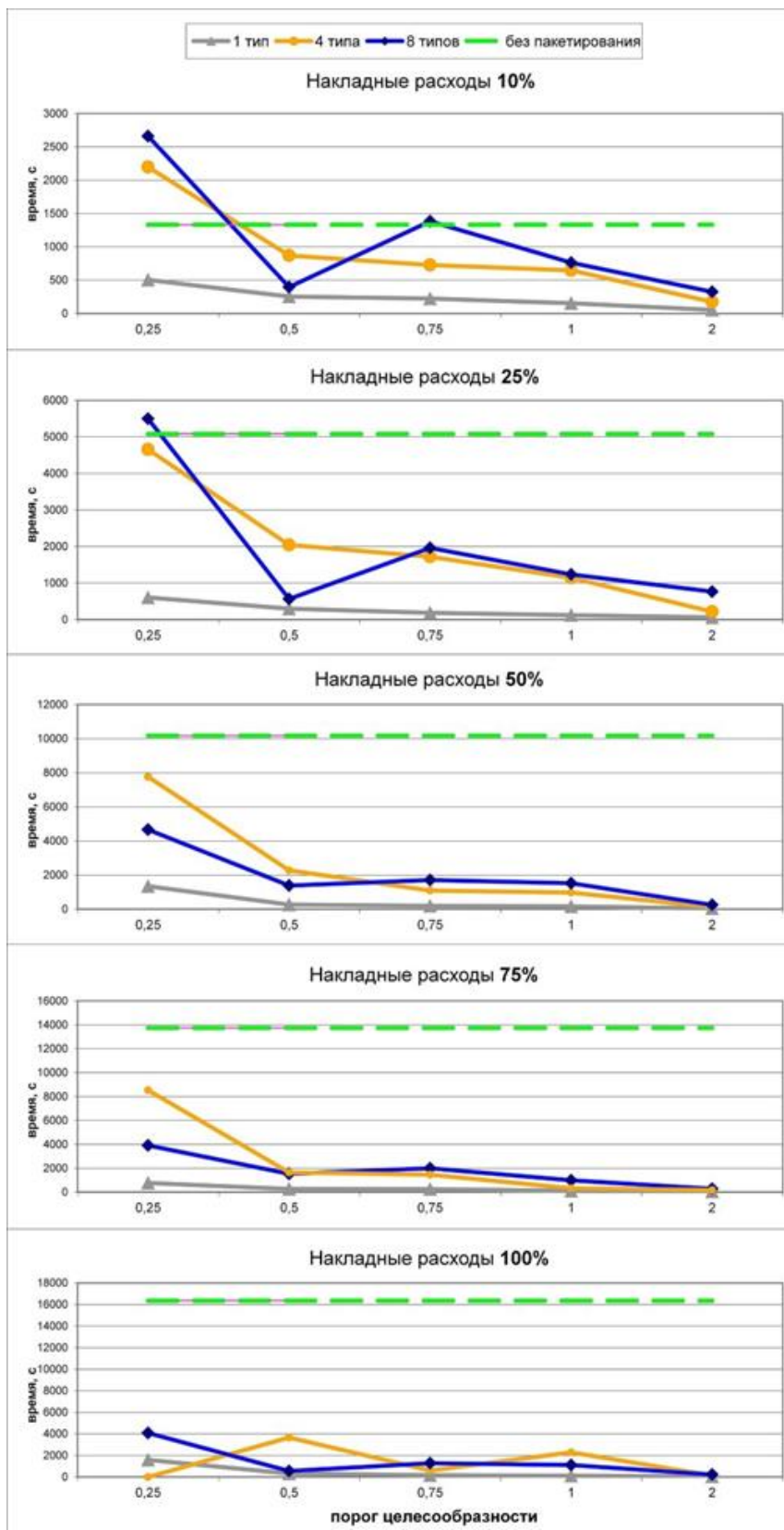


Рис. 4. Зависимость среднего времени ожидания в очереди заданий от величины порога целесообразности (от 0,25 до 2), накладных расходов на инициализацию (от 10 до 100 %) и числа типов заданий

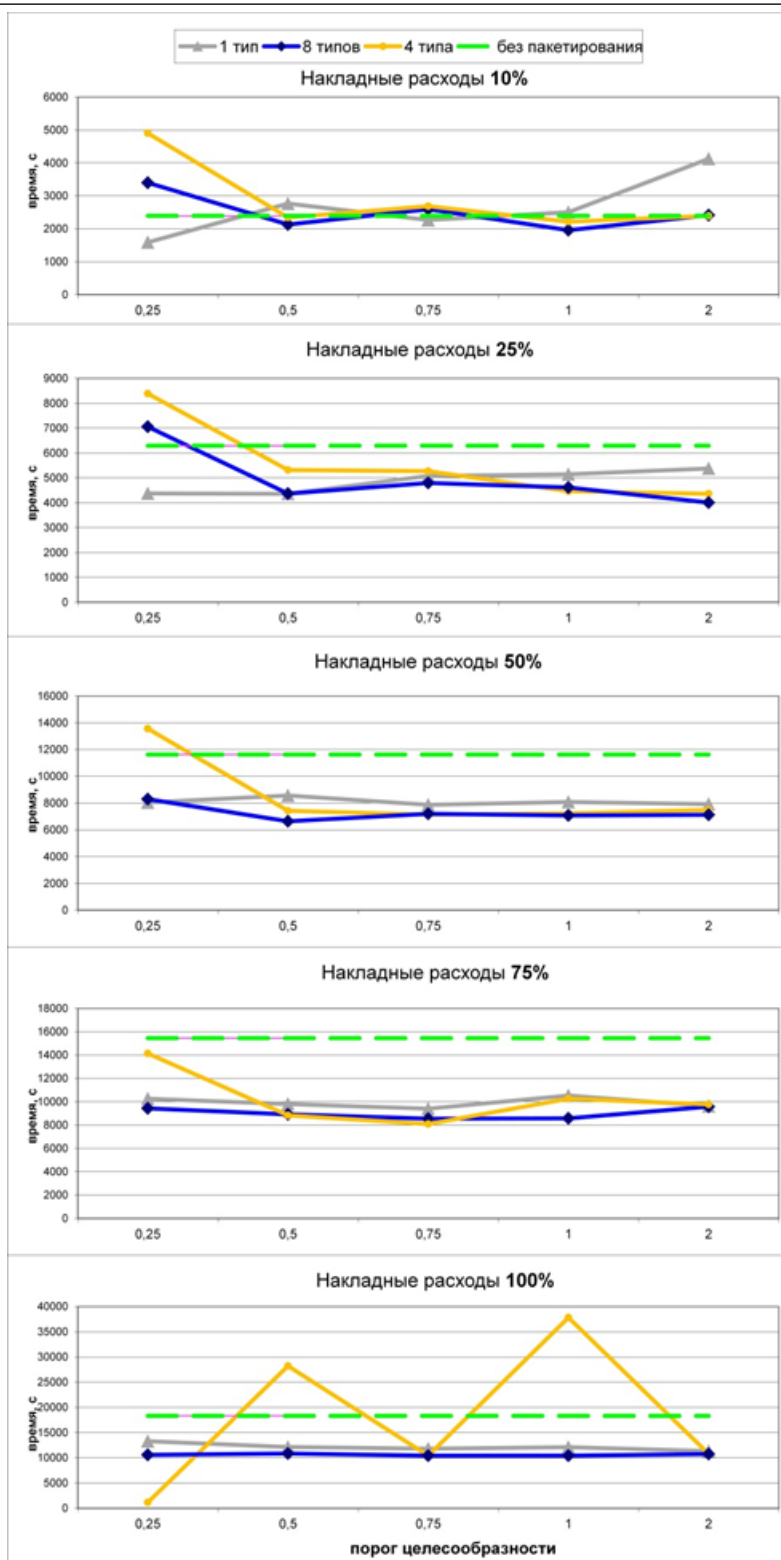


Рис. 5. Зависимость среднего времени обслуживания задания от величины порога целесообразности (от 0,25 до 2), накладных расходов на инициализацию (от 10 до 100 %) и числа типов заданий

Заключение

Авторами работы была проведена серия экспериментов с целью определения влияния пакетирования однотипных заданий на показатели эффективности подсистемы планирования параллельных заданий: полезную загрузку вычислителя, среднее время ожидания задания в очереди, среднее время полного об-

служивания задания. В ходе экспериментов варьировались число типов заданий (от 1 до 8), накладные расходы на инициализацию (от 10 до 100 %) и такой параметр подсистемы пакетирования, как порог целесообразности, отражающий отношение времени обработки пакета заданий ко времени инициализации пакета.

Результаты экспериментов показывают, что применение пакетирования приносит положительный эффект, начиная с накладных расходов в 10 %. На исследуемые показатели эффективности в этом случае заметное влияние оказывают число типов заданий и порог целесообразности. Для случая достаточно больших накладных расходов (50 % и выше) эксперименты продемонстрировали достаточно ожидаемые результаты: применение пакетирования существенно улучшает исследуемые показатели эффективности, существенного влияния числа типов заданий и порога целесообразности при этом не наблюдается.

Разработанная авторами подсистема пакетирования однотипных заданий может быть применена в высокопроизводительных системах коллективного пользования при обработке заданий, представленных в виде набора виртуальных машин или контейнеров. В этом случае старт задания будет предваряться достаточно длительной процедурой развертывания виртуальных машин или контейнеров из заранее подготовленных образов, хранимых в специально выделенной области файловой системы. Предполагается, что пользователь, желающий развернуть для своего задания определенную виртуальную программную платформу, будет указывать при постановке задания в очередь конкретный размещенный в хранилище подготовленный образ, фактически определяя этим тип задания для подсистемы пакетирования.

Авторы планируют продолжить экспериментальные исследования с целью определения влияния пакетирования на показатели эффективности планирования заданий для случая более сложной модели входного потока заданий, включающей не только идеально масштабируемые задания.

Данная работа выполнена при поддержке гранта РФФИ №16-07-01098

Литература

1. Шабанов Б.М., Телегин П.Н., Аладышев О.С. Особенности использования многоядерных процессоров // Программные продукты и системы. 2008. № 2. С. 7–9.
2. Левин И.И., Дордопуло А.И., Каляев И.А., Доронченко Ю.И., Раскладкин М.К. Современные и перспективные высокопроизводительные вычислительные системы с реконфигурируемой архитектурой // Вестн. ЮУрГУ: сер.: Вычислительная математика и информатика. 2015. № 3. С. 24–39.
3. Баранов А.В., Николаев Д.С. Использование контейнерной виртуализации в организации высокопроизводительных вычислений // Программные системы: теория и приложения. 2016. Т. 7. № 1. С. 117–134.
4. Аладышев О.С., Баранов А.В., Ионин Р.П., Киселев Е.А., Орлов В.А. Сравнительный анализ вариантов развертывания программных платформ для высокопроизводительных вычислений // Вестн. УГАТУ. 2014. № 3. С. 295–300.
5. СУППЗ – Система Управления Прохождением Параллельных Заданий. URL: <http://suppz.jssc.ru/> (дата обращения: 13.11.2016).
6. Slurm Workload Manager. URL: <https://slurm.schedmd.com/> (дата обращения: 23.08.2017).
7. Moab HPC Suite. URL: <http://www.adaptivecomputing.com/products/hpc-products/moab-hpc-basic-edition/> (дата обращения: 23.08.2017).
8. Баранов А.В., Киселев Е.А., Ляховец Д.С. Квазипланировщик для использования простаивающих вычислительных модулей многопроцессорной вычислительной системы под управлением СУППЗ // Вестн. ЮУрГУ: сер.: Вычислительная математика и информатика. 2014. № 4. С. 75–84.
9. Баранов А.В., Ляховец Д.С. Влияние пакетирования на эффективность планирования параллельных заданий // Программные системы: теория и приложения. 2017. 8:1 (32). С. 193–208.