

УДК 681.324

С.В. Минухин

Харьковский национальный экономический университет имени Семена Кузнеця, Харьков

МЕТОДЫ И ОЦЕНКИ ИНТЕРВАЛОВ ОБРАБОТКИ ПАКЕТОВ ЗАДАНИЙ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Рассмотрены методы пакетной обработки заданий в распределенных вычислительных системах. Проанализированы особенности методов обработки пакетов заданий в детерминированной постановке. Исследованы метод планирования на основе задачи о наименьшем покрытии. Предложены оценки интервалов обработки заданий. Приведены результаты вычислительных экспериментов и использования оценок интервалов планирования, позволяющие повысить эффективность функционирования распределенных систем в условиях различных интенсивностей и законов распределения входных потоков заданий.

Ключевые слова: пакет заданий, распределенная вычислительная система, покрытие, интенсивность, интервал

Введение

В последнее время в распределенных системах получили развитие методы обработки пакетов заданий, характеризующиеся интервалом (периодичностью) обработки и количеством обрабатываемых заданий.

К ним относятся:

немедленный (on-line) режим, характеризующийся тем, что поступившее в систему задание сразу отправляется на выполнение;

пакетный (off-line) режим, в котором задания предварительно собираются в пакеты (bag of tasks), далее выбираются подходящие ресурсы и назначаются для выполнения заданий [1–3, 5, 8].

Планирование реализуется так называемым отображением (mapping) заданий на ресурсы – задачей, являющейся NP-сложной [1], в связи с чем методы ее решения постоянно находятся во внимании исследователей и актуальность которых несомненна.

Отметим сложность определения ключевых характеристик пакетного планирования – размера пакета заданий и периода (интервала) времени между двумя последовательными тактами планирования (отображениями). Выделим некоторые из факторов, влияющих на их выбор:

- в случае, если количество заданий в пакете значительно, используемый метод планирования требует применения быстрых алгоритмов;

- для выполнения большого количества заданий требуется большое количество ресурсов. В противном случае увеличивается время ожидания заданий в очередях, формируемых на ресурсы;

- ресурсы в моменты времени планирования должны быть свободны или для некоторых из них должен быть небольшой промежуток времени до завершения выполнения тех заданий, которые на них еще решаются;

- в случае частого планирования резко увеличивается количество служебной информации по пересылке необходимых для управления

процессом планирования данных, что приводит к увеличению времени выполнения заданий;

- в случае редкого планирования увеличивается количество простаивающих ресурсов, в случае, если количество коротких заданий достаточно велико. что приводит к снижению коэффициента использования ресурсов.

Целью данного исследования является повышение эффективности работы пакетных методов обработки заданий на основе анализа параметров характеристик распределенной системы и определения интервала обработки для различных законов распределения интенсивностей входных потоков заданий.

Формулировка проблемы

Рассматривается функционирование распределенной системы, использующей обработку пакетов заданий (метазаданий), причем назначение этих заданий на машины (отдельные ПК, кластеры, многопроцессорные системы, мультикластерные и Грид-системы) должно обеспечивать высокую производительность системы в целом.

Формально проблема формулируется следующим образом: имеется набор заданий, определяемый множеством $T = \{t_1, t_2, \dots, t_m\}$, набор машин, определяемый множеством $M = \{m_1, m_2, \dots, m_n\}$, ожидаемое время выполнения каждого задания t_i на каждой машине $m_j - e_{ij}$ ($1 \leq i \leq m, 1 \leq j \leq n$), определяемое матрицей $ETC(i, j)$ и показатель (метрика) эффективности выполнения заданий $F(X)$. Необходимо найти такое назначение заданий $\{t_1, t_2, \dots, t_m\}$ на машины $\{m_1, m_2, \dots, m_n\}$, при котором величина $F(X)$ достигает оптимума.

В качестве критериев оптимизации (метрик производительности) работы распределенной системы используем следующие [2, 3, 5, 7–11]:

минимизацию времени завершения выполнения последнего задания (*makespan*),

определяемое как: $\min \max_{t_i \in T} F$;

минимизацию общего времени завершившихся заданий (*flowtime*), определяемое в соответствии с выражением: $\min \sum_i F_{ii}$;

максимизацию коэффициента использования ресурса m_j (resource utilization) R_{u_mj} , который характеризуют уровень использования (загрузки) ресурса с учетом спланированных на этот ресурс заданий за время планирования:

$$R_{u_mj} = \sum_1^n \frac{(mat(m_j) / makespan)}{n},$$

где $mat(m_j)$ – время выполнения последнего назначенного на ресурс m_j задания.

Детерминированные методы пакетной обработки заданий

Метод *Min-Min* включает расчет матрицы ожидаемых значений времени выполнения $ETC(i, j)$ для любых заданий i и ресурсов j , использующий $ETC(i, j)$ и значение $available(j)$ (или готовности ресурса - $(ready)(j)$) по формуле:

$$ETC(i, j) = ETC(i, j) + available(j).$$

Для любого задания t_i и ресурса m_i определяется наименьшее время завершения задания для всех элементов j -й строки матрицы. После этого выбирается задание t_k с наиболее ранним временем завершения и отображается на соответствующий ресурс m_k . Далее это задание удаляется из списка заданий и значений времени завершения матрицы и для каждого из оставшихся заданий и ресурсов процедура повторяется. Она повторяется до тех пор, пока все задания очереди не будут назначены на ресурсы.

Метод *Max-Min* использует то, что для любого задания i , как только определился ресурс m_i , на котором выполняется задание с наименьшим временем выполнения, определяется задание t_k с наибольшим временем и отображается на этот ресурс. Метод *Max-Min* планирует и короткие, и длинные задания, в то время как *Min-Min* метод планирует сначала короткие задания, а потом – длинные, и при этом значение параметра времени выполнения последнего задания из заданий входной очереди (*makespan*) увеличивается.

Метод *Sufferage* позволяет лучше планировать в случае, когда для выполнения задания выбирается тот ресурс, на котором оно будет «терять» (*sufferage*) больше, если бы оно было назначено на любой другой ресурс. Для реализации этого метода параметры задания определяются как разница между вторым скорейшим временем завершения задания на ресурсе m_1 и первым скорейшим временем завершения задания на ресурсе m_k . Работа метода начинается с маркировки всех доступных ресурсов, и далее, на каждой итерации выбирается в соответствии с расписанием задание j . С этой целью для задания j , ресурсов m_i и m_j вычисляются значения «потерь» (*sufferage*). Если ресурс m_i доступен, то задание j назначается на ресурс m_i , если m_i уже занят заданием j' , то задания j и j' и будут конкурировать за

ресурс m_i , а «победителем» становится задание с большим значением «потерь» (*sufferage*).

Метод *LJFR-SJFR* существенно отличается от рассмотренных методов тем, что одновременно минимизирует значения двух метрик производительности распределенных систем – *makespan* и *flowtime*. Его можно рассматривать как использующий параллельно два метода: самое длинное задание размещается на самый производительный ресурс (метод *LJFR*), и при этом минимизируется значение *makespan*, и самое короткое задание размещается на самый производительный ресурс (метод *SJFR*), и при этом минимизируется значение *flowtime*. В начале первого этапа реализации данного метода самые длинные задания назначаются на простаивающие ресурсы (самое длинное задание – на самый быстрый ресурс). Далее, для оставшихся для выполнения заданий на каждом шаге выбирается самый быстрый (производительный) из ресурсов, на которых задание завершилось (является свободным), при этом выбирается и назначается для выполнения на этом ресурсе самое короткое задание (*SJFR*) или самое длинное задание (*LJFR*).

Метод *Relative Cost method* (относительной стоимости) учитывает, как загрузку ресурсов, так и время выполнения заданий на ресурсах. Поэтому для конкретного задания метод определяет ресурс, который в наибольшей степени соответствует времени выполнения задания. Используется критерий – соответствие по близости (*matching proximity*), который отличается от ранее рассмотренных критериев как критерий оценки работы именно пакетных методов планирования.

Отметим, что во всех рассмотренных методах важным фактором построения эффективного расписания является выполнение того требования, что время, затраченное на реализацию приведенных эвристик, должно быть значительно меньше, чем время выполнения заданий на ресурсах.

В качестве *нижней оценки* интервала планирования предлагается использовать выражение:

$$O(\max(m^2n, m^3)) + (\min(ETC(i, j)) = \min((ETC(i, j) + available(j))).$$

В приведенной формуле предполагается, что нижняя граница периода планирования определяется временной сложностью самого алгоритма планирования (первый член в сумме – в данном случае использованы оценки жадных алгоритмов и минимальным временем освобождения ресурса в соответствии с выражением, использующем самое раннее освобождение ресурса на момент времени t (второй член в сумме). Таким образом, основная идея предлагаемого метода заключается в определении момента времени, когда освободится, по крайней мере, один ресурс. Отметим, что если одновременно освобождается несколько ресурсов, это дает возможность одновременно планировать задания на несколько ресурсов.

Для определения верхней оценки периодичности планирования необходимо:

1. Выбирать по аналогии с нижней оценкой планирования в качестве верхней оценки значение $\min \max (t_i, t_i \in T)$, определяющее время завершения выполнения последнего задания, нельзя вследствие того, что на этот момент времени практически все ресурсы освободятся, и значительное количество ресурсов при этом будет простаивать.

2. Для оптимизации производительности системы необходимо планировать задания до момента времени завершения последнего задания, чтобы, таким образом, в очереди на ресурс в любой момент времени находилось хотя бы одно задание.

Для решения этой задачи предлагается использовать следующие процедуры:

определить минимальный допустимый радиус устойчивости на момент времени планирования. В качестве радиуса устойчивости можно использовать среднеквадратическое отклонение: в случае, если для заданной интенсивности потоков заданий для фиксированного количества используемых ресурсов значение радиуса устойчивости находится в пределах допустимого, то интервал планирования определяется количеством заданий n , поступивших в очередь, и равняется количеству доступных на момент времени планирования ресурсов. Это позволит обеспечить минимальное время нахождения заданий в очереди на ресурсы;

в случае увеличения интенсивности потока заданий необходимо уменьшать период планирования, т. е. сокращать промежуток времени между двумя последовательными событиями планирования и, таким образом, уменьшать размерность матрицы $ETC(i, j)$. Это позволит компенсировать увеличение интенсивности за счет уменьшения времени работы алгоритма планирования, которое увеличивается с увеличением количества заданий t . При этом используется приведенная процедура определения времени планирования для фиксированного потока заданий.

Метод пакетной обработки заданий на основе задачи о наименьшем покрытии

Основывается на задаче линейного булевого программирования [12], математическая постановка которой имеет вид:

$$L_t = \sum_{j=1}^n x_j(t_k) \rightarrow \min \quad (1)$$

при ограничениях

$$\sum_{j=1}^n \beta_{ij} x_j(t_k) \geq 1, i = \overline{1, m}; \quad (2)$$

$$\beta_{ij} \in \{0, 1\}; x_j(t_k) \in \{0, 1\}$$

где m – количество заданий, подлежащих планированию; n – количество ресурсов системы, доступных и свободных на момент планирования; $t_k \in [T_0, T_N]$.

Планирование осуществляется на интервале времени $[T_0, T_N]$, где T_0 – время начала

планирования; T_N – время окончания планирования заданий очереди.

Задачу (1), (2) можно рассматривать как задачу определения минимального числа столбцов в булевой матрице B , покрывающего единицами все строки матрицы, элементы которой в контексте решения задачи планирования интерпретируются следующим образом: столбцам соответствуют свободные на момент планирования ресурсы системы, а строкам – задания, подлежащие планированию и которые должны быть решены на этих ресурсах.

Данный подход базируется на следующих положениях.

1. Система планирования организована в виде двухуровневой структуры, на первом уровне которой из глобальной очереди, например, по приоритету, выбирается множество заданий (пакет заданий, называемый в предлагаемой модели *пулом*), подлежащие планированию, к ним применяется метод решения задачи (1), (2). Далее выбранные, как результат ее решения, задания назначаются на доступные и свободные на момент планирования ресурсы и решаются на них под управлением локального планировщика.

2. Метод планирования на каждом шаге планирования максимально загружает минимальное количество свободных и доступных на момент планирования ресурсов. Таким образом, для последующего распределения заданий очереди количество ресурсов для возможного назначения на них заданий будет максимальным.

3. Алгоритм решения задачи (1), (2) должен иметь малую временную сложность его реализации для минимизации времени, отводимого на процесс планирования заданий для их выполнения на ресурсах.

4. Система планирования использует пакетную технологию: задания, организованные в форме пакета (пула) заданий, выбираются из глобальной очереди, по мере их планирования на ресурсы помещаются в пакет заданий на назначенный ресурс (ресурсы) и передаются на решение на этот ресурс (ресурсы).

В качестве единицы времени планирования и проводимых расчетов используется внутреннее время имитационной модели системы – 1 такт, которое соответствует времени решения одного задания, имеющего сложность 100 000 MI (million instructions), на ресурсе, производительность которого 100 000 MIPS (million instructions per second).

Для вычислительных экспериментов, в которых учитывается сложность задания, задания малой сложности (трудоемкости) определялись в интервале от 10 до 300 тактов, средней – от 300 до 10 000 тактов, высокой – от 10 000 и выше тактов.

В качестве метрик производительности работы распределенной системы используются следующие: среднее время выполнения задания (время ответа), время выполнения всех заданий

очереди и коэффициент использования (загрузки) ресурсов.

Время выполнения одного задания определяется по формуле:

$$t_{i \text{ выполнения}} = t_{i \text{ нул}} + t_{i \text{ планир}} + t_{i \text{ пакет}} + t_{i \text{ решения}},$$

где $t_{i \text{ нул}}$ – время нахождения i -го задания в пуле;

$t_{i \text{ планир}}$ – время планирования i -го задания;

$t_{i \text{ пакет}}$ – время нахождения i -го задания в пакете заданий на ресурсе;

$t_{i \text{ решения}}$ – время решения i -го задания на ресурсе.

Среднее время выполнения одного задания определяется по формуле:

$$t_{\text{среднее_время_выполнения_задания}} =$$

$$\frac{1}{N} \sum_{i=1}^N t_{i \text{ выполнения}},$$

где N – количество заданий, находящихся во входной очереди.

Время выполнения всех заданий очереди T_N – время работы модели распределенной системы, начиная от момента времени формирования глобальной (входной) очереди и заканчивая моментом времени завершения выполнения последнего задания $T_{\text{посл}}$ рассчитывается по формуле:

$$T_N = T_{\text{посл}} - T_{\text{перв}}$$

Коэффициент использования ресурса R_j определяется по формуле:

$$K_{\text{использ}} = \frac{T_{R_i}}{T_N},$$

где T_{R_i} – время выполнения всех заданий глобальной очереди, которые выполнялись на ресурсе R_j .

При проведении вычислительного эксперимента для исследования эффективности метода на основе решения задачи (1), (2) были использованы следующие значения характеристик среды распределенных вычислений:

1. Интенсивность входного потока заданий генерировалась на основе экспоненциального закона распределения с математическим ожиданием равным 5, 10, 15, 20, 40 и нормального закона распределения с математическим ожиданием равным 5, 10, 20 и среднеквадратическим отклонением равным 2.

2. Количество ресурсов – 10.

Производительность ресурсов – генерировалась по равномерному закону в интервале от 3 до 5.

3. Сложность (трудоемкость) задания генерировалась по нормально распределенному закону с математическим ожиданием 30 и среднеквадратическим отклонением 5.

4. Периодичность (интервал) планирования – время между последовательными процедурами планирования, в тактах.

Верхняя граница оценки величины интервала планирования определяется по формуле:

$$t_{\text{периодичность}} = \min t_{\text{планир}} + \quad (3)$$

$$\min t_{\text{пакет_заданий}} + \min(t_{\text{сложность}} / P_{\text{ресурса}}),$$

где $t_{\text{планирован и я}}$ – время работы алгоритма планирования;

$t_{\text{пакет_заданий}}$ – время нахождения в пакете заданий;

$t_{\text{сложность}}$ – сложность задания;

$P_{\text{ресурса}}$ – производительность ресурса.

Для применения формулы (3) использованы методы теории экстремальных значений (величин) определения максимального значения одиноко распределенных случайных величин с функциями плотности распределения и средним значением, определяемых по формулам [13, 14]:

$$f_1(x) = n f(x) [1 - F(x)]^{n-1},$$

$$E [X_1] = \int_{-\infty}^{\infty} x f_1(x) dx = n \int_{-\infty}^{\infty} x f(x) [1 - F(x)]^{n-1} dx.$$

Приведенные на рис. 1 – 6 результаты вычислительных экспериментов определяют закономерности изменения значений времени выполнения заданий, времени ответа и коэффициента использования.

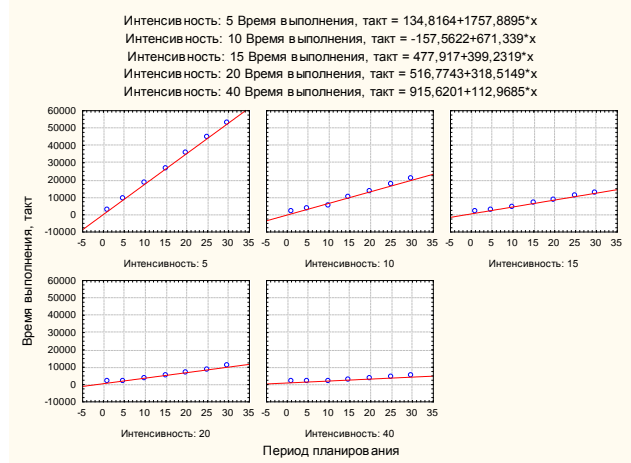


Рис. 1. Зависимость времени выполнения от периодичности планирования и различных интенсивностей для экспоненциального закона распределения входного потока заданий

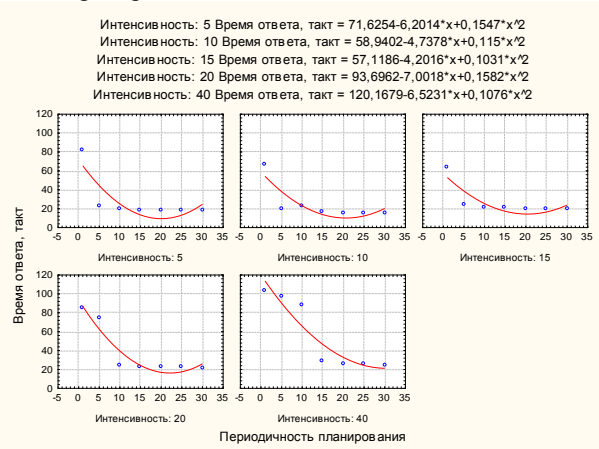


Рис. 2. Зависимость времени ответа от периодичности планирования и различных интенсивностей

интенсивностей для экспоненциального закона распределения входного потока заданий

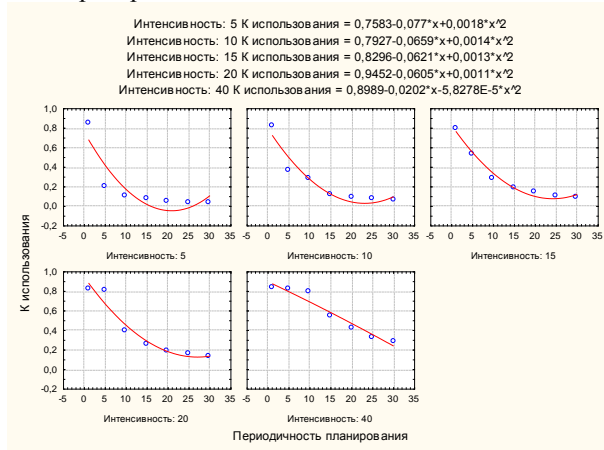


Рис. 3. Зависимость коэффициента использования от периодичности планирования и различных интенсивностей для экспоненциального закона распределения входного потока заданий

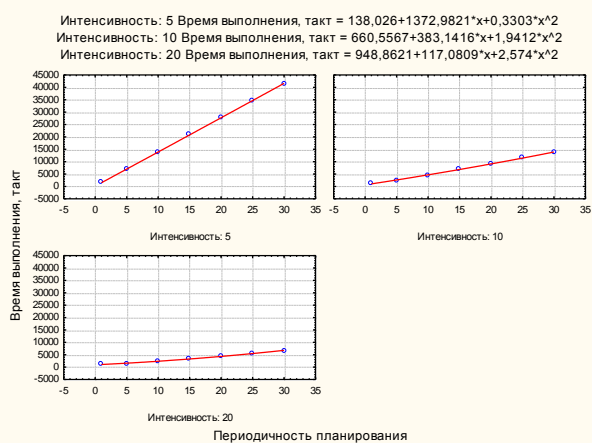


Рис. 4. Зависимость времени выполнения заданий от периодичности планирования и различных интенсивностей для нормального закона распределения входного потока заданий

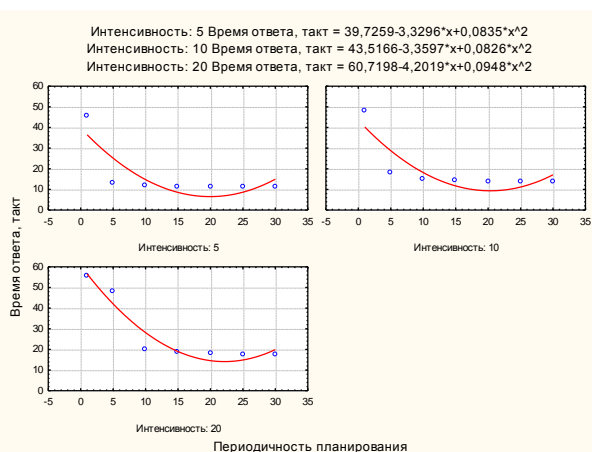


Рис. 5. Зависимость времени ответа от периодичности планирования и различных интенсивностей для нормального закона распределения входного потока заданий

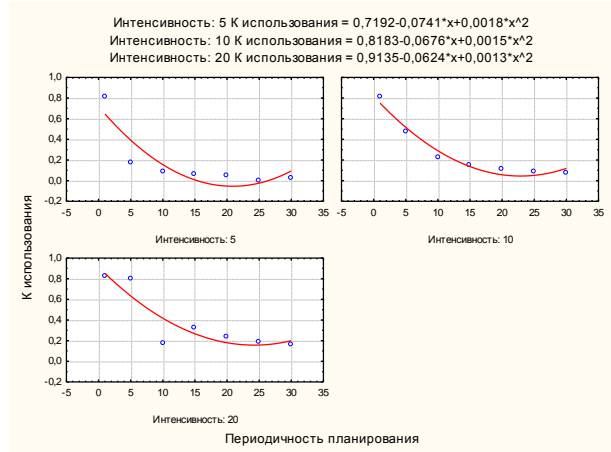


Рис. 6. Зависимость коэффициента использования от периодичности планирования и различных интенсивностей для нормального закона распределения входного потока заданий

Выводы. 1. При больших интенсивностях поступления заданий, т.е. значительном превышении количества поступивших заданий по отношению к количеству ресурсов (в эксперименте – 10 ресурсов), закономерности и тенденции для коэффициента использования, времени ответа и времени выполнения всех заданий для потоков заданий, имеющих разные законы распределения, являются аналогичными. При этом экспоненциальный закон интенсивности потока заданий является более «трудоемким» по сравнению с нормальным законом, что следует из полученных значений времени выполнения для интенсивностей 10 и 20 (рис. 1 и 4).

2. В случае, если среднее значение интенсивности потока заданий меньше количества ресурсов (10) и периодичность планирования меньше среднего времени освобождения ресурса (в эксперименте она равна 6 тактам, определяемая как сложность 30 тактов /производительность 5 тактов), как это показано на первой диаграмме с интенсивностью 5 на каждом из рис. 1–6, то имеет место минимальное время выполнения заданий и максимальное время ответа, а также максимальное значение коэффициента использования (для случая, если такт планирования равен единице 1).

3. Если среднее значение интенсивности потока заданий меньше количества ресурсов (в данном случае – 10) и периодичность планирования больше среднего времени освобождения ресурса, наблюдается линейный рост времени выполнения заданий и экспоненциальное уменьшение времени ответа и коэффициента использования (см. диаграмму с интенсивностью 5 и периодичностью в диапазоне от 10 до 30 на каждом из рис. 1–6). Время ответа при этом стремится к своему минимальному значению, которое возможно в случае постоянного наличия свободного и выделяемого для задания ресурса. При этом увеличение периодичности планирования приводит к простоям ресурсов и уменьшению коэффициента использования.

4. В случае, если среднее значение интенсивности потока заданий больше количества ресурсов и периодичность планирования меньше среднего времени

освобождения ресурса (в данном случае – 6 тактов), наблюдается незначительный линейный рост времени выполнения и близкое к линейному уменьшение времени ответа и коэффициента использования (см. диаграммы с интенсивностью 20 и периодичностью в диапазоне от 1 до 5 на каждом рис. 1–6). Оптимальные значения периодичности планирования пропорциональны уровню интенсивности потока заданий, т. е. с увеличением интенсивности по отношению к количеству ресурсов в 2 раза оптимальный период планирования равен 5 тактам, при увеличении интенсивности в 4 раза – 10 тактам. В данном случае достигается минимальное значение времени выполнения заданий и максимальное значение коэффициента использования.

5. Если среднее значение интенсивности заданий больше количества ресурсов и периодичность планирования больше среднего времени освобождения ресурса (например, в интервале от 10 до 30 тактов), имеет место незначительный линейный рост времени выполнения и линейное уменьшение времени ответа и коэффициента использования (см. диаграммы с интенсивностью 20 на каждом из рис. 1 – 6).

Список литературы

1. Maheswaran M. *Dynamic mapping of a class of independent tasks onto heterogeneous computing systems* / M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, R. F. Freund // *Journal of Parallel and Distributed Computing*. – 1999. – V. 59. – № 2. – P. 107 – 121.
2. Xhafa F. A. *Batch mode scheduling in grid systems* / F. Xhafa, L. Barolli, A. Durresi // *Int. J. Web and Grid Services*. – 2007. – V. 3. – № 1. – P. 19 – 37.
3. Xhafa F. *Meta-heuristics for Grid Scheduling Problems* / F. Xhafa, A. Abraham // *Meta. for Sched. in Distri. Comp. Envi., SCI 146*, 2008. – P. 1–37.
4. Braun T. D. *A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems* / T. D. Braun, H. J. Siegel, N. Beck, L. B'ol'oni, R. F. Freund, D. Hensgen, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao // *Journal of Parallel and Distributed Computing*. – 2001. – V. 61. – № 6. – P. 810 – 837.
5. Smith J. *Batch Mode Stochastic-Based Robust Dynamic Resource Allocation in a Heterogeneous Computing System* / J. Smith, J. Apodaca, Anthony A. Maciejewski, H. J. Siegel // *In Proceedings of PDPTA'2010*. – P. 263 – 269.
6. Al-Qawasmeh Abdulla M. *Statistical measures for quantifying task and machine Heterogeneities* / Abdulla M. Al-Qawasmeh, Anthony A. Maciejewski, Haonan Wang, J. Smith, Howard J. Siegel, J. Potter // *J. Supercomput.* – 2011. – V. 57. – №1. – P. 34–50.
7. Shestak V. *Stochastic robustness metric and its use for static resource allocations* / V. Shestak, J. Smith, Anthony A. Maciejewska, Howard J. Siegel // *Journal of Parallel and Distributed Computing*. – 2008. – V. 68. – №. 8. – P. 1157 – 1173.
8. Ashish M. Mehta. *Dynamic Resource Management Heuristics for Minimizing Makespan while Maintaining an Acceptable Level of Robustness in an Uncertain Environment* / M. Mehta Ashish, J. Smith, H. J. Siegel, Anthony A. Maciejewski, A. Jayaseelan, B. Ye // *In 12th International Conference on Parallel and Distributed Systems (ICPADS 2006)*. – V. 1. – P. 107 – 114.
9. Al-Qawasmeh Abdulla M. *Characterizing Task-Machine Affinity in Heterogeneous Computing Environments* / Abdulla M. Al-Qawasmeh, Anthony A. Maciejewski, Rodney G. Robert, Howard J. Siegel // *25th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2011, Anchorage, Alaska, USA, 16-20 May 2011 - Workshop Proceedings, IEEE 2011*. – P. 34 – 44.
10. Yarmolenko V., Duato J., Panda D. K., Sadayappan P. *Characterization and enhancement of dynamic mapping heuristics for heterogeneous systems* // *In International Conference on Parallel Processing Workshops 2000 (ICPPW '2000), Aug. 2000*. – P. 437 – 444.
11. Perathoner S. *Reliable Mode Changes in Real-Time Systems with Fixed Priority or EDF Scheduling* / S. Perathoner, N. Stoimenov, L. Thiele // *Proceedings of the Conference on Design, Automation and Test in Europe Conference & Exhibition*. – 2009. – P. 99 – 104.
12. Листрової С.В. *Метод рішення задач о мінімальному вершинному покритті в произвольному графі і задачі о найменшому покритті* / С.В. Листрової, С.В. Мінухін // *Електронне моделювання*. – 2012. – Т. 34. – №1. – С. 29 – 43.
13. Gumbel E. J. *Statistics of extremes*. – New York: Columbia University Press, 1958. – 375 p.
14. Cremonesi P. *Modeling the Effects of Node Heterogeneity on the Performance of Grid Applications* / P. Cremonesi, R. Turrin, V. N. Alexandrov // *Journal of Networks*. – 2009. – V. 4. – № 9. – P. 837 – 854.

Рецензент: д-р техн. наук, проф. О.Г. Руденко, зав. кафедрою ЕВМ Харківського національного університету радіоелектроніки, Харків, Україна.

Автор:

МИНУХИН Сергей Владимирович

Харківський національний економічний університет імені Семена Кузнеця, Харків, професор кафедри інформаційних систем, к.т.н., доцент, e-mail: ms_vl@mail.ru.

МЕТОДИ ТА ОЦІНКИ ІНТЕРВАЛІВ ОБРОБЛЕННЯ ПАКЕТІВ ЗАВДАНЬ У РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

С.В. Мінухін

Розглянуто методи пакетної обробки завдань в розподілених обчислювальних системах. Проаналізовано особливості методів обробки пакетів завдань в детермінованій постановці. Досліджено метод планування на основі задачі про найменшому покритті. Запропоновано оцінки інтервалів обробки завдань. Наведені результати обчислювальних експериментів і використання оцінок інтервалів планування, що дозволяють підвищити ефективність функціонування розподілених систем в умовах різних інтенсивностей та законів розподілу вхідних потоків завдань.

Ключові слова: пакет завдань, розподілена обчислювальна система, покриття, інтенсивність, інтервал

METHODS AND EVALUATION OF THE PROCESSING INTERVAL OF A PACKAGE TASKS IN DISTRIBUTED COMPUTING SYSTEMS

S.V. Minukhin

The methods of batch mode of tasks in distributed computing systems are developed. The features of packet processing methods jobs in deterministic statement are investigated. Investigated method based on a minimal cover. The assessment intervals processing tasks are proposed. The results of numerical experiments and the use of estimates intervals scheduling that improve the efficiency of distributed systems under different distribution laws of input streams of tasks are obtained.

Keywords: task package, distributed computing system, cover, intensity, interval