

© 2011 г. Т.С. Дьячук

Запорожский национальный технический университет, Запорожье

ОПРЕДЕЛЕНИЕ ПРИОРИТЕТОВ ЗАДАНИЙ В РАСПРЕДЕЛЕННОЙ СИСТЕМЕ

Предложен метод расчета приоритета задания, поступающего в распределенную систему. Сортировка заданий согласно приоритетам необходима для решения задачи планирования ресурсов и определения порядка решения заданий. Рассмотрен алгоритм обслуживания пользовательского задания и предложена методика, позволяющая оценить общие затраты времени на выполнение задания.

The method for calculating the task priorities in a distributed system is submitted. Sorting of tasks according to the priorities is needed to address the problem of resource scheduling and determine the order of decision tasks. The algorithm of user's task service and estimation method of the total time spent for the task execution are developed.

Введение

В настоящее время все большее распространение получает использование Grid-систем в различных областях науки и промышленности. Grid предполагает объединение вычислительных ресурсов различных организаций и лабораторий с целью совместного использования, то есть предоставления своих ресурсов, когда они простаивают, и получения вычислительных мощностей при необходимости. Для эффективного использования ресурсов в Grid-среде необходимо учитывать характеристики заданий, поступающих в систему. Эти характеристики отражаются в параметрах ресурсного запроса [1].

При планировании ресурсов важной задачей является определение приоритетов пользовательских запросов на вычислительные ресурсы. Приоритет определяет порядок выполнения задания.

1. Подходы для определения приоритета

Существует несколько подходов для решения данной проблемы. В системе OurGrid [2] применяется схема предпочтений, при которой каждый узел ведет учет выполненных работ для другого узла. И в зависимости от этого может использовать чужие ресурсы для своих задач. Таким образом, приоритет запроса зависит от приоритета узла, с которого этот запрос поступил. Хотя данный подход и решает вопрос райдинга [2] (райдер – пользователь, который

потребляет ресурсы, но не предлагает свои), но имеет и свои недостатки.

Для определения приоритета запроса может быть использован экономический подход к Grid – вычислениям [3], который является системой учета потребления ресурсов. Экономический подход позволит определить приоритет ресурсного запроса, а также поможет решить вопрос райдинга. Пользователи должны платить за использование ресурсов владельцам этих ресурсов, валюта может быть реальной или условной. Продавцами выступают владельцы ресурсов, а покупателями – пользователи. Таким образом, в среде Grid могут быть применены рыночные модели.

Запросы на ресурсы бывают локальные и глобальные. Необходимо распределять имеющиеся ресурсы между локальными и глобальными заданиями, учитывая их приоритеты. Локальные пользователи имеют преимущество на использование локальных ресурсов. Приоритет локальным запросам назначает администратор. Приоритет глобального запроса определяется, исходя из его параметров. Возможна ситуация, при которой глобальный запрос на ресурсы будет иметь больший приоритет, чем локальный.

С одной стороны ресурсы не должны простаивать, но приносить наибольшую прибыль, а с другой задания пользователей должны быть решены как можно быстрее с наименьшими затратами. Цена ресурсов должна меняться во времени в зависимости от спроса и внутренних

потребностей. Путем повышения цены возможно прекращение поступления глобальных запросов. А пользователь должен иметь возможность повысить приоритет своего запроса с помощью поднятия предложенной платы за ресурсы, и таким образом конкурировать с локальными запросами. Все эти факторы формируют цену и соответственно влияют на приоритет ресурсного запроса.

Факторы, формирующие цену использования ресурсов:

- объем и характеристики ресурса;
- отношение реальной производительности ресурса к эталонным показателям;
- часы пик;
- спрос со стороны локальных запросов;
- приоритет локального задания, претендующего на данный ресурс.

2. Параметры ресурсного запроса

Рассмотрим параметры ресурсного запроса [1], которые характеризуют задание, поступающее в систему:

s – число параллельных ветвей задания;

t_o – ориентировочное время выполнения ветви задания на эталонном ненагруженном процессоре, при запуске задания происходит пересчет времени в соответствии с параметрами исполнительных ресурсов;

T_{max} – максимальное время ожидания выполнения задания;

$(h1, h2, \dots)$ – вектор, содержащий необходимые характеристики ресурса для выполнения ветви задания;

Pay – предложенная плата за использование ресурсов.

Таким образом, ресурсный запрос Z имеет вид:

$$Z(s, t_o, T_{max}, (h1, h2, \dots), Pay) \quad (1)$$

Введем следующие обозначения:

g_i – стоимость i -го ресурса в единицу времени;

wn_i – нормирующий коэффициент для определения реального времени выполнения ветви задания, который учитывает реальную производительность i -го ресурса и текущую нагрузку.

После подбора подходящих по характеристикам ресурсов проверяется выполнение условий по времени выполнения задания и предложенной плате.

Прогнозируемое время счета задания:

$$T_{count} = \max_i (t_o \cdot wn_i) \quad (2)$$

T_{wait} – время ожидания выполнения задания.

Время выполнения задания:

$$T_{max} \geq T_{count} + T_{wait} \quad (3)$$

Стоимость выполнения задания:

$$Cost = \sum_{i=1}^s g_i \cdot (t_o \cdot wn_i) \quad (4)$$

3. Алгоритм обслуживания пользовательского задания

Алгоритм обслуживания пользовательского задания отражает следующие состояния процесса выполнения задания (рис. 1):

- C1 – проверка наличия подходящих ресурсов;
- C2 – проверка условий оплаты;
- C3 – проверка наличия свободных ресурсов;
- C4 – оценка времени выполнения заданий;
- C5 – процесс выполнения заданий;
- C6 – наличие сбоя в работе системы;
- C7 – вычисление приоритета заданий;
- C8 – резервирование ресурсов;
- C9 – оценка времени выполнения задания;
- C10 – формирование очереди на обслуживание;
- C11 – отказ в обслуживании;
- C12 – получение результата.

Состояние C1 является исходным, а состояния C11 и C12 – конечные состояния. Между перечисленными состояниями существуют следующие переходы:

- C1-C2 – переход к проверке условий оплаты;
- C1-C11 – отказ в обслуживании в связи с отсутствием необходимых ресурсов;
- C2-C3 – переход к проверке наличия свободных ресурсов;
- C2-C11 – отказ в обслуживании в связи с недостаточностью денежных средств;
- C3-C4 – переход к оценке времени выполнения задания;
- C3-C7 – переход к оценке приоритета;
- C4-C11 – завершение процесса в связи с превышением допустимого времени выполнения задания;
- C4-C5 – переход к выполнению задания;
- C5-C6 – переход к проверке выполнения задания;
- C6-C4 – переход к повторному запуску задания на выполнение в связи с наличием сбоев;
- C6-C12 – получение положительного результата выполнения задания;
- C7-C8 – переход к резервированию необходимых ресурсов;
- C8-C9 – переход к оценке времени выполнения задания;

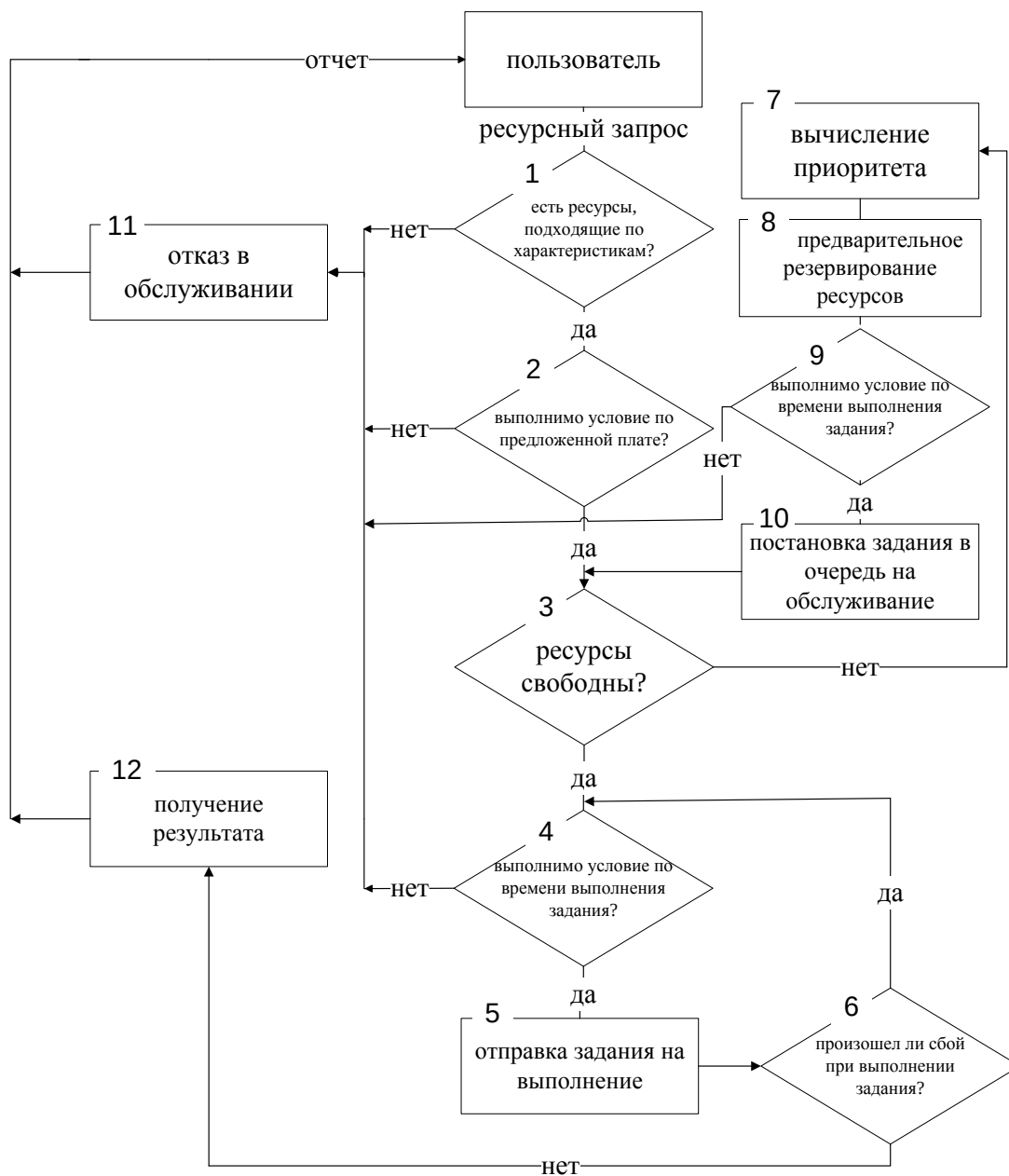


Рис. 1. Алгоритм обслуживания пользовательского задания

C9-C10 – переход на постановку в очередь на обслуживание;

C9-C11 – отказ в обслуживании в связи с необходимостью значительного времени на выполнение задания;

C10-C3 – переход к проверке наличия свободных ресурсов.

Описанный процесс подразумевает решение потока разнообразных задач и, следовательно, переходы между состояниями имеют случайный характер. Будем считать, что существуют постоянные вероятности переходов между со-

стояниями, тогда данный процесс может быть представлен конечной цепью Маркова [4]. Наличие состояний завершения процесса позволяет сделать вывод о том, что данная цепь Маркова является поглощающей.

Структура матрицы переходов для поглощающей цепи Маркова имеет вид [4]:

$$P = \begin{bmatrix} Q & R \\ O & E \end{bmatrix}, \quad (5)$$

где O, E – нулевая и единичная подматрицы; Q – подматрица, описывающая поведение про-

цесса до попадания в поглощающее состояние;
 R - подматрица переходов в поглощающие состояния.

Использование фундаментальной матрицы $F = (E - Q)^{-1}$ позволяет получить ряд важнейших характеристик исследуемого процесса. Элемент $f_{i,j}$ матрицы F дает ожидаемое количество моментов времени, которое проводит процесс в состоянии j до попадания в погло-

щающее состояние при условии, что он начался в состоянии i .

Вероятность завершения процесса в том или ином поглощающем состоянии в зависимости от того, какое состояние является исходным, определяется элементами матрицы $B = FR$.

Составим матрицу переходов P , из которой выделим матрицы Q и R :

$$P = \begin{bmatrix} 0 & p_{1-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{1-11} & 0 \\ 0 & 0 & p_{2-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{2-11} & 0 \\ 0 & 0 & 0 & p_{3-4} & 0 & 0 & p_{3-7} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{4-5} & 0 & 0 & 0 & 0 & 0 & p_{4-11} & 0 \\ 0 & 0 & 0 & 0 & 0 & p_{5-6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{6-4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{6-12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{7-8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{8-9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{9-10} & p_{9-11} & 0 \\ 0 & 0 & p_{10-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

$$Q = \begin{bmatrix} 0 & p_{1-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{2-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{3-4} & 0 & 0 & p_{3-7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{4-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & p_{5-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{6-4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{7-8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{8-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{9-10} \\ 0 & 0 & p_{10-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$R = \begin{bmatrix} p_{1-11} & 0 \\ p_{2-11} & 0 \\ 0 & 0 \\ p_{4-11} & 0 \\ 0 & 0 \\ 0 & p_{6-12} \\ 0 & 0 \\ 0 & 0 \\ p_{9-11} & 0 \\ 0 & 0 \end{bmatrix}.$$

Фундаментальная матрица определяется выражением

$$F = \begin{bmatrix} 1 & -p_{1-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -p_{2-3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -p_{3-4} & 0 & 0 & -p_{3-7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -p_{4-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -p_{5-6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -p_{6-4} & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -p_{7-8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -p_{8-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -p_{9-10} \\ 0 & 0 & -p_{10-3} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1}.$$

Так как процесс начинается в состоянии 1, то нас будут интересовать только элементы первой строки матрицы F , которые будут представлять собой вектор f .

Для нахождения этих элементов воспользуемся решением системы уравнений, коэффициенты которой представляют транспони-

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -p_{1-2} & 1 & 0 & 0 & 0 & 0 \\ 0 & -p_{2-3} & 1 & 0 & 0 & 0 \\ 0 & 0 & -p_{3-4} & 1 & 0 & -p_{6-4} \\ 0 & 0 & 0 & -p_{4-5} & 1 & 0 \\ 0 & 0 & 0 & 0 & -p_{5-6} & 1 \\ 0 & 0 & -p_{3-7} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -p_{10-3} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -p_{7-8} & 1 & 0 & 0 \\ 0 & -p_{8-9} & 1 & 0 \\ 0 & 0 & -p_{9-10} & 1 \end{bmatrix} \begin{bmatrix} f_{1-1} \\ f_{1-2} \\ f_{1-3} \\ f_{1-4} \\ f_{1-5} \\ f_{1-6} \\ f_{1-7} \\ f_{1-8} \\ f_{1-9} \\ f_{1-10} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

или в виде системы уравнений

$$\begin{cases} f_{1-1} = 1; \\ f_{1-2} = p_{1-2} \cdot f_{1-1} = p_{1-2}; \\ f_{1-3} = p_{2-3} \cdot f_{1-2} + p_{10-3} \cdot f_{1-10}; \\ f_{1-4} = f_{1-3} \cdot p_{3-4} + f_{1-6} \cdot p_{6-4}; \\ f_{1-5} = f_{1-4} \cdot p_{4-5}; \\ f_{1-6} = f_{1-5} \cdot p_{5-6}; \\ f_{1-7} = f_{1-3} \cdot p_{3-7}; \\ f_{1-8} = f_{1-7} \cdot p_{7-8}; \\ f_{1-9} = f_{1-8} \cdot p_{8-9}; \\ f_{1-10} = f_{1-9} \cdot p_{9-10}. \end{cases}$$

Выразим f_{1-10} и подставим найденное выражение в f_{1-3} с учетом значения f_{1-2} :

$$\begin{aligned} f_{1-10} &= f_{1-9} \cdot p_{9-10} = f_{1-8} \cdot p_{8-9} \cdot p_{9-10} = \\ &= f_{1-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10} = f_{1-3} \cdot p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10}; \\ f_{1-3} &= p_{1-2} \cdot p_{2-3} + p_{10-3} \cdot f_{1-10} = p_{1-2} \cdot p_{2-3} + \\ &+ p_{10-3} \cdot f_{1-3} \cdot p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10}. \end{aligned}$$

Найдём f_{1-3} :

$$f_{1-3} = \frac{p_{1-2} \cdot p_{2-3}}{1 - p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10} \cdot p_{10-3}}.$$

Выразим f_{1-6} и подставим найденное выражение в f_{1-4} :

$$\begin{aligned} f_{1-6} &= f_{1-5} \cdot p_{5-6} = f_{1-4} \cdot p_{4-5} \cdot p_{5-6}; \\ f_{1-4} &= f_{1-3} \cdot p_{3-4} + f_{1-6} \cdot p_{6-4} = \end{aligned}$$

рованную матрицу $(E - Q)^t$, а правая часть представлена столбцом, у которого первый элемент равен 1, а остальные нули:

$$(E - Q)^t \cdot f = C,$$

то есть

$$\begin{aligned} &= \frac{p_{1-2} p_{2-3} p_{3-4}}{1 - p_{3-7} p_{7-8} p_{8-9} p_{9-10} p_{10-3}} + f_{1-4} p_{4-5} p_{5-6} p_{6-4}. \\ \text{Найдём } f_{1-4} &= \\ &= \frac{p_{1-2} p_{2-3} p_{3-4}}{(1 - p_{3-7} p_{7-8} p_{8-9} p_{9-10} p_{10-3}) \cdot (1 - p_{4-5} p_{5-6} p_{6-4})} \end{aligned}$$

Таким образом, запишем выражения для значений элементов первой строки матрицы F :

$$\begin{cases} f_{1-1} = 1; \\ f_{1-2} = p_{1-2}; \\ f_{1-3} = \frac{p_{1-2} \cdot p_{2-3}}{1 - p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10} \cdot p_{10-3}}; \\ f_{1-4} = \frac{p_{1-2} p_{2-3} p_{3-4}}{(1 - p_{3-7} p_{7-8} p_{8-9} p_{9-10} p_{10-3})(1 - p_{4-5} p_{5-6} p_{6-4})}; \\ f_{1-5} = \frac{p_{1-2} p_{2-3} p_{3-4} p_{4-5}}{(1 - p_{3-7} p_{7-8} p_{8-9} p_{9-10} p_{10-3})(1 - p_{4-5} p_{5-6} p_{6-4})}; \\ f_{1-6} = \frac{p_{1-2} p_{2-3} p_{3-4} p_{4-5} p_{5-6}}{(1 - p_{3-7} p_{7-8} p_{8-9} p_{9-10} p_{10-3})(1 - p_{4-5} p_{5-6} p_{6-4})}; \\ f_{1-7} = \frac{p_{1-2} \cdot p_{2-3} \cdot p_{3-7}}{1 - p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10} \cdot p_{10-3}}; \\ f_{1-8} = \frac{p_{1-2} \cdot p_{2-3} \cdot p_{3-7} \cdot p_{7-8}}{1 - p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10} \cdot p_{10-3}}; \\ f_{1-9} = \frac{p_{1-2} \cdot p_{2-3} \cdot p_{3-7} \cdot p_{7-8} \cdot p_{8-9}}{1 - p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10} \cdot p_{10-3}}; \\ f_{1-10} = \frac{p_{1-2} \cdot p_{2-3} \cdot p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10}}{1 - p_{3-7} \cdot p_{7-8} \cdot p_{8-9} \cdot p_{9-10} \cdot p_{10-3}}. \end{cases}$$

Вероятность $P_{вып}$ выполнения запущенной программы определяется произведением полученной строки на второй столбец матрицы R . Так как все элементы столбца, кроме элемента под номером 6, равны нулю, то получим

$$P_{вып} = f_{1-6} \cdot p_{6-12} = \frac{p_{1-2} p_{2-3} p_{3-4} p_{4-5} p_{5-6} p_{6-12}}{(1 - p_{3-7} p_{7-8} p_{8-9} p_{9-10} p_{10-3})(1 - p_{4-5} p_{5-6} p_{6-4})}.$$

Вероятность $\overline{P_{вып}}$ того, что программа не будет выполнена, как противоположного события, равна

$$\overline{P_{вып}} = 1 - f_{1-6} \cdot p_{6-12}$$

Общие затраты T_z на выполнение задания определяется суммой

$$T_z = \sum_{i=1}^{10} f_{1-i} t_i,$$

где t_i – время на выполнение соответствующего этапа.

4. Принцип назначения приоритетов

Входящим заданиям приписывается ряд параметров, определяющих относительную позицию сообщения в очереди. Данная позиция является переменной функцией времени, благодаря появлению в очереди заданий с более высоким приоритетом. Для любого момента времени t приоритет данного задания определяется как функция приписываемых параметров. Чем больше значение функции, тем больше приоритет задания.

Существует несколько видов систем приоритетов [5]:

- 1) система с фиксированным приоритетом без прерывания;
- 2) система с фиксированным приоритетом с прерыванием;
- 3) система динамических приоритетов без прерывания;
- 4) система динамических приоритетов с прерыванием.

Фиксированный приоритет означает, что задание получает неизменный приоритет сразу после поступления в очередь на основании своих характеристик. Однако при данном методе возможен случай бесконечного ожидания некоторым заданием в очереди момента запуска на выполнение, из-за прихода более высокоприоритетных заданий.

Динамический приоритет (или приоритет с зависимой задержкой [5]) предполагает изменение приоритета в зависимости от времени нахождения задания в очереди. Чем дольше находится задание в очереди, тем выше становится приоритет.

Система без прерывания предусматривает, что как только задание поступит на обработку, оно уже не может быть прервано более высокоприоритетным заданием.

В системе с прерыванием допускается остановка обработки задания при поступлении задания с высшим приоритетом. Когда прерванное задание вновь поступает в устройство обработки, его обслуживание необходимо продолжить с момента остановки (возобновление после прерывания). К недостаткам данного метода можно отнести накладные расходы, связанные с остановкой и возобновлением обслуживания. Невозможность спланировать и оценить время обслуживания заданий.

Таким образом, был выбран механизм динамических приоритетов без прерывания.

Приоритет задания определяется как:

$$Pr = a + (t - T) \cdot b, \quad (6)$$

где $a = Pay - Cost$, если значение $a < 0$, то задание не проходит, по параметру предложенной платы и такой ресурсный запрос отклоняется (рис.1);

T – время поступления задания в очередь;

t – текущее время, которое изменяется от значения T до времени начала обслуживания задания;

b – величина, характеризующая владельца задания; с помощью данного параметра может быть назначен больший приоритет локальным заданиям, так же могут быть отсечены запросы недобросовестных пользователей.

Задания сортируются по убыванию значения приоритета. Механизм динамических приоритетов является гибким механизмом для упорядочения заданий в общей очереди. Так как исключается возможность бесконечного ожидания момента запуска на выполнение задания, чем дольше задание находится в очереди, тем выше становится его приоритет.

Выводы

В работе проведен анализ и предложен метод расчета приоритета задания, поступа-

ющего в распределенную систему. Сортировка заданий согласно приоритетам необходима для решения задачи планирования ресурсов и определения порядка решения заданий. Предложенный механизм назначения приоритетов учитывает время нахождения задания в очереди, предложенную плату за выполнение задания, а также характеристику самого владельца задания. Что позволяет глобальным запросам конкурировать с локальными, а владельцам ресурсов получать наибольшую прибыль. В работе рассмотрен алгоритм обслуживания пользовательского задания. Предложенная методика позволяет оценить время ожидания обслуживания задания, с определенными параметрами.

завдання, що надходить в розподілену систему. Сортування завдань відповідно до пріоритетів необхідне для вирішення задачі планування ресурсів та визначення порядку вирішення завдань. Розглянуто алгоритм обслуговування завдання користувачів та запропоновано методику, що дозволяє оцінити загальні затрати часу на виконання завдання.

СПИСОК ЛИТЕРАТУРЫ

1. Дьячук Т.С. Параметры ресурсного запроса в распределенной системе. – высокопроизводительные параллельные вычисления на кластерных системах. материалы Девятой международной конференции-семинара. – Владимир, 2–3 ноября 2009г.: изд-во Владимирского государственного университета, 2009. – с. 163-165
2. Cirne W., Brasileiro F., Andrade N., Costa L.B., Andrade A., Novaes R., Mowbray M. Labs of the World, Unite!!! – [Электронный ресурс] – Режим доступа: <http://copin.ufcg.edu.br/twiki-public/pub/OG/OurPublications/LabsoftheWorldUnitev19.pdf>
3. Buyya R., Giddy J., Abramson D. A case for economy Grid architecture for service-oriented Grid computing// Proceedings of the Heterogeneous Computing Workshop, April 2001 – [Электронный ресурс] – Режим доступа: <http://gridbus.org/papers/ecogrid.pdf>
4. Кемени Дж., Снелл Дж. Конечные цепи Маркова. – Главная редакция физико-математической литературы изд-ва «Наука», М., 1970 – 272с.
5. Клейнрок Л. Коммуникационные сети (стохастические потоки и задержки сообщений). – Главная редакция физико-математической литературы изд-ва «Наука», М., 1970 – 256с.

T.S. Diachuk. Defining the task priorities in distributed system.

Т.С. Дьячук. Визначення пріоритетів завдань у розподіленій системі.

Запропоновано метод розрахунку пріоритету
