

## ЗАДАЧА ОПТИМИЗАЦИИ ВРЕМЕНИ ВЫПОЛНЕНИЯ ПРИЛОЖЕНИЯ В DESKTOP GRID

*Н.Н. Никитина, А.С. Румянцев*

*Институт прикладных математических исследований  
Карельского научного центра РАН, Петрозаводск*

В настоящее время для ускорения вычислений широко применяются системы типа Desktop Grid. Данные системы распределенных вычислений представляют собой совокупность неспециализированных вычислительных единиц – в частности, персональных компьютеров, принадлежащих организации или частным лицам. При этом сравнительно небольшая мощность отдельных вычислительных узлов компенсируется их большим количеством и наличием высокоскоростных каналов связи между ними, таких как локальные сети организации и Интернет. Эффективность функционирования таких систем зависит от методов распределения заданий по вычислителям, который, определяет алгоритм работы менеджера очереди. Отметим работы [1–4], в которых рассматриваются различные подходы к построению эффективных расписаний для Desktop Grid систем.

Системы Desktop Grid предназначены для выполнения приложений, которые требуют большой вычислительной мощности (хранения больших объемов данных) и при этом могут быть разделены на большое количество независимых задач. Одной из ключевых особенностей систем Desktop Grid является непостоянство множества вычислительных узлов. Вычислительные узлы доступны лишь в то время, которое в одностороннем порядке устанавливают их владельцы (производя так называемые добровольные вычисления). При составлении расписания задач для Desktop Grid необходимо учитывать отказы узлов, связанные с их временной или постоянной недоступностью. Другой важной особенностью, связанной с архитектурой системы, является возможность получения в результате вычислений недостоверного либо искаженного значения, а также отсутствие результата в заданный промежуток времени (называемый deadline). Указанные особенности приводят к необходимости дублирования задач с одинаковыми исходными данными для повышения надежности вычислений, что, однако, полностью не исключает появления ошибок.

В данной работе приводится постановка задачи оптимизации расписания для системы Desktop Grid с учетом указанных особенностей. Предположим, что в состав системы Desktop Grid входит множество вычислительных узлов  $G$  мощностью  $|G|=K$ . Положим для простоты, что процесс вычислений можно представить в виде условных тактов времени (соответствующих значению deadline). Каждому узлу  $g \in G$  поставлено в соответствие значение вероятности отказа за один вычислительный такт  $P(g)$ ,  $0 \leq P(g) \leq 1$ .

Рассмотрим набор значений  $\{p_i\}_{i=1}^{\infty}$ , называемых порогами отказов узлов:  $p_0 = 0 < p_1 < p_2 < \dots < p_i < \dots$ . В соответствии с данным набором значений узлы могут быть разбиты на непересекающиеся группы  $G_i \subseteq G$ ,  $\bigcup_{i=1}^{\infty} G_i = G$ :

$$G_i = \{g \in G : P(g) \in [p_{i-1}, p_i)\}, \quad 1 \leq i \leq \infty.$$

Обозначим мощности групп узлов, полученных данным разбиением, через  $|G_i| = K_i$ , при этом  $\sum_{i=1}^{\infty} K_i = K$ . Вероятность отказа на любом из узлов группы  $G_i$ , по построению, не превосходит  $p_i$ .

Предположим, что в системе нужно выполнить приложение, состоящее из  $n$  идентичных независимых задач, каждая из которых выполняется в течение одного вычислительного такта. В целях ускорения процесса вычислений разобьем множество задач на группы  $\{n_i\}_{i=1}^{\infty}, n_i \geq 0, \sum_{i=1}^{\infty} n_i = n$ , и назначим каждую группу из  $n_i$  задач на выполнение соответствующей группе узлов  $G_i$ . При этом задача, назначенная группе узлов  $G_i$ , дублируется на  $i$  узлов этой группы для повышения надежности. Таким образом, в группе узлов  $G_i$  может одновременно выполняться не более  $\left\lfloor \frac{K_i}{i} \right\rfloor$  уникальных задач. С учетом ненадежности узлов, задача может не выполняться за один вычислительный такт. В таком случае будет сгенерирован новый экземпляр данной задачи, который будет отправлен на выполнение той же группе узлов в том же количестве экземпляров. Вероятность того, что ни один экземпляр некоторой задачи в группе  $G_i$  не выполнится успешно за один вычислительный такт, не превосходит  $p_i^i$ , за  $j$  вычислительных тактов –  $p_i^{ji}$ , что соответствует геометрическому распределению с вероятностью успеха  $p_i^i$ . Тогда математическое ожидание общего количества задач (включая исходную), порожденных одной задачей, назначенной группе узлов  $G_i$ , составляет  $\frac{1}{1-p_i^i}$ .

При полной надежности вычислительных узлов общая продолжительность вычислений в группе  $G_i$  составила бы

$$\left\lceil \frac{n_i}{\left\lfloor \frac{K_i}{i} \right\rfloor} \right\rceil$$

тактов. С учетом задач, порожденных в результате отказов, ожидаемая общая продолжительность вычислений в группе  $G_i$  составит

$$t_i = \frac{\left\lceil \frac{n_i}{\left\lfloor \frac{K_i}{i} \right\rfloor} \right\rceil}{1-p_i^i}.$$

Задача оптимизации расписания может быть сформулирована следующим образом:

$$\max_i t_i \longrightarrow \min_{\{p_i\}, \{n_i\}}.$$

Критерием оптимальности является минимизация времени выполнения приложения, то есть наибольшей продолжительности вычислений в группах узлов. Предполагается решать задачу с использованием методов теории игр.

Работа выполнена при поддержке РФФИ, проекты 10-07-00017, 12-07-31147.

## Литература

1. Benoit A. et al. Scheduling concurrent bag-of-tasks applications on heterogeneous platforms // *IEEE Transactions on Computers*, 2010. Vol. 59, Iss. 2. P. 202–217.
2. Celaya J., Marchal L. A fair decentralized scheduler for bag-of-tasks applications on desktop grids // *Cluster Computing and the Grid, IEEE International Symposium on*, 2010. P. 538–541.
3. Kondo D., Casanova H. computing the optimal makespan for jobs with identical and independent tasks scheduled on volatile hosts // *Technical Report CS2004-0796, Dept. of Computer Science and Engineering, University of California at San Deigo*, July, 2004.
4. Kondo D., Casanova H., Chien A. Scheduling task parallel applications for rapid application turnaround on enterprise desktop grids // *Journal of Grid Computing*. 2007. 5(4). P. 379–405.