

ТЕХНОЛОГИЯ РАСПРЕДЕЛЕНИЯ ТИПОВЫХ ЗАДАНИЙ СО СЛУЧАЙНЫМ ВРЕМЕНЕМ ВЫПОЛНЕНИЯ И УБЫВАЮЩЕЙ ФУНКЦИЕЙ ПОТЕРИ ЦЕННОСТИ РЕШЕНИЯ В РАСПРЕДЕЛЕННОЙ РАЗНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

П.Е. Голосов

Для решения заданий типов A_1, A_2, \dots применяется совокупность распределенных разнородных вычислительных ресурсов R_1, R_2, \dots , объединенных в единую систему. Относительно независимые пользователи U_1, U_2, \dots , начиная с некоторого момента времени $t_0=0$, обращаются к администратору системы с запросами на выполнение имеющихся у них наборов заданий:

$$\{W_v^{(i)}\}, v=1, 2, \dots, i=1, 2, \dots, \quad (1)$$

где через $W_v^{(i)}$ обозначим v -й запрос пользователя U_i . Тип этого задания обозначим через $j_v^{(i)}$.

Особенности задач таковы, что объем вычислений в них является случайным и не зависит от интересов пользователя и типа используемого вычислительного ресурса. Это означает следующее. Конкретное задание состоит из конечного набора элементарных заданий, очередность выполнения которых указывает пользователь, но количество элементарных заданий, необходимых для выполнения всего задания, является случайной величиной ξ_j , вероятностное распределение которой зависит от типа соответствующего задания A_j . Такая постановка вопроса не исключает ситуации, когда случайная величина ξ_j принимает одно фиксированное значение. Время выполнения таких заданий изучалось в [1,2]. Характерным свойством рассматриваемых заданий является возможность распараллеливания их выполнения. При этом ветви выполняемого параллельного задания зависимы по данным. Максимальную степень распараллеливания задания типа A_j (максимальное количество одновременно выполняемых ветвей задания) обозначим как m_j .

Ресурсоемкость задания типа A_j жестко связана с алгоритмом его решения, потенциально ориентированном на конкретный ресурс R_g , поэтому в качестве ее количественной характеристики рассмотрим вектор:

$$\bar{r}_j = (r_{j1}, r_{j2}, \dots), \quad (2)$$

в котором r_{jg} - это количество элементарных заданий задания типа A_j , выполняемых на ресурсе R_g за время T . Время T выбирается единым для всех типов задач.

Такой подход к выбору общей единицы сравнения ресурсоемкости предлагался ранее, например, в [3].

В свою очередь векторы (2) определяют интегральную характеристику ресурсов всей системы за время T в виде матрицы, составленной из векторов \bar{r}_j для всех типов заданий:

$$\tilde{R} = \|\bar{r}_{jg}\|, \quad (3)$$

размеры которой определяются числом типов заданий и числом различных видов вычислительных ресурсов распределенной системы. Далее всюду полагаем, что мощности множеств $A = \{A_1, A_2, \dots\}$ и $R = \{R_1, R_2, \dots\}$ конечны и равны, соответственно, $|A| = a$ и $|R| = r$.

Отметим аддитивность характеристики (3), которая заключается в том, что для того, чтобы оценить возможности применения системы за время \mathcal{Y} достаточно умножить матрицу \tilde{R} на скаляр \mathcal{Y} .

При определенных условиях, которые довольно часто возникают в реальных приложениях, общий объем необходимых вычислений для выполнения всех заданий (1) и пожелания пользователей в части сроков их выполнения приходят в противоречие с возможностями системы.

Тем самым взаимные интересы пользователей системы и интересы ее администратора приходят к неразрешимым противоречиям.

При коммерческом использовании ресурсов вычислительных систем, в частности при выполнении пользовательских заданий на ресурсах распределенных сред, образованных с применением грид-технологий [4], предлагаются способы разрешения этих противоречий за счет использования условно-денежного исчисления при определении очередности заданий для выполнения [7].

Говорить о реально денежном исчислении в административно зависимых системах довольно сложно, вследствие чего требуется определить отличный от «коммерческого» распределения ресурсов (за плату) способ. Практика использования многоядерных вычислителей позволяет предположить следующее.

Ожидаемая ценность (получаемая выгода) деятельности пользователей системы позволяет лицу, принимающему решения на основе анализа матрицы (3), выделить квоты каждому пользователю на каждое необходимое для обеспечения его деятельности задание типа A_j . Эта квота представима в виде следующего вектора:

$$\bar{U}_i = (U_{i1}, U_{i2}, \dots), \quad (4)$$

в котором U_{ij} - это количество элементарных заданий, выделенных пользователю U_i для решения задания типа A_j на время T . Понятно, что элементы матрицы квот составлены из строк \overline{U}_i :

$$\tilde{U} = \|U_{ij}\|, \quad (5)$$

размеры такой матрицы определяются числом пользователей и числом типов заданий, и должны удовлетворять условиям:

$$\sum_i U_{ij} \leq \sum_g r_{jg}, \quad (6)$$

Соблюдение соглашения (6) обеспечивает лицо, принимающее решение.

Основываясь на результатах непрерывного анализа ситуаций, определяющих ценность получаемых решений заданий и прогноза характера их развития, пользователь может управлять порядком расходования собственной квоты (4).

Для этого пользователь должен обладать возможностью приостановки выполнения собственных заданий, заменяя их по собственному усмотрению на более приоритетные (на основе субъективной оценки). Обратной стороной такой замены является потенциальная потеря выгоды от неполученной информации, ожидаемой от выполнения предыдущих заданий. Для управления процессами выполнения заданий в системе вводится следующее понятие – функция актуальности задания (функция старения) – любая невозрастающая функция от времени, которая обращается в ноль на конечном отрезке времени. С некоторыми такими функциями можно познакомиться в [2]. В общем случае вид и значения, принимаемые функцией старения, не зависят от типа задания.

Поскольку соблюдение квот – необходимое условие деятельности администратора, пользователь для задания обязан указать предельное время пребывания задания в системе и выбрать одну из предлагаемых системой функций потери актуальности (старения) выполнения задания.

Функцию актуальности v -го задания i -го пользователя на временном интервале $[0, T]$, численно выражающую получаемую выгоду, потенциально ожидаемую пользователем при использовании результатов выполнения данного задания обозначим через $F_v^{(i)}(t)$ (7).

Как правило, функция актуальности выбирается из заданного множества.

Таким образом, администратор системы обладает следующей совокупностью знаний:

1. Перечень заданий и условия их выполнения (включая количество необходимых и возможных ветвей распараллеливания задания);
2. Наборы заданий пользователей (1);
3. Матрица ресурсов системы (3);
4. Матрица квот пользователей (4);
5. Функции старения пользовательских заданий (7).

В конечном счете, основная задача (выгода) администратора заключается в обработке таких заданий, которые в совокупности максимизируют величину их суммарной актуальности, за интервал планирования T .

В количественном выражении в рамках введенных выше понятий этот функционал можно записать в виде:

$$\mathfrak{Z}(t) = \sum_i \sum_v F_v^{(i)}(\tau(W_v^{(i)})), \quad (8)$$

где суммирование ведется по всем невыполненным заданиям всех пользователей. В данном выражении $\tau(W_v^{(i)})$ – есть оставшееся время выполнения задания $W_v^{(i)}$ и является случайной величиной.

Для практических расчетов значения случайных величин в (8) заменяют на их математические ожидания. Таким образом основная функция администратора состоит в составлении такого порядка выполнения заданий и указания такого допустимого распараллеливания их выполнения, которые максимизируют функционал (8) при ограничениях (6).

Результаты экспериментов показали, что планирование выполнения заданий на основе введения функции актуальности повышает общую результативность функционирования распределенных вычислительных систем без потерь использования ресурсов, возможно, с задержками выполнения заданий с невысокой актуальностью.

ЛИТЕРАТУРА:

1. В.Н. Суриков. Задача о ключах. – Обзорение прикладной и промышленной математики. т. 16, вып. 2., 2009, стр. 274-275.
2. А.Ф. Ронжин А.Ф., В.Н. Суриков. Линейные разделимые статистики и процессы восстановления, связанные с обобщением «задачи о ключах». - Обзорение прикладной и промышленной математики. т. 16, Вып. 2., 2008, стр.272-273.
3. О.И. Ларичев. Методы принятия управленческих решений.
<http://old.ultsu.ru/people/sosn/n/umk/decisions-support/metod/model.html>

4. I. Foster, I. and C. Kesselman, (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.
5. П.Е. Голосов . Планирование заданий и ресурсов в сетевой среде распределенных вычислений на основе экономических моделей. // Труды международной конференции «Программные системы: теория и приложения» / Под редакцией С. М. Абрамова – М.: Физматлит, 2006 т. 1 – с. 297-305 – ISBN 5-94052-128-1
6. С.А. Мотылев С.А. Проблемы подготовки и проведения деловых игр, моделирующих информационно-аналитическую деятельность. Журнал «Дистанционное образование» № 6, 1999 г, с. 23
7. И.А. Бахарев, В.А. Крюков. Управление прохождением задач на ЭВМ, препринт Института прикладной математики им. М. В. Келдыша АН СССР, 1981, № 149, 24 с.