

Генетический алгоритм организации мультипрограммного функционирования распределенных вычислительных систем*

А.В. Ефимов

Сибирский государственный университет телекоммуникаций и информатики, Новосибирск

Рассмотрена проблема организации мультипрограммного функционирования распределенных вычислительных систем (ВС). Предложен генетический алгоритм оптимизации функционирования распределенных ВС при обработке наборов параллельных программ с нефиксированными параметрами. Представлены результаты моделирования.

В архитектурном плане распределенная вычислительная система (ВС) представляется множеством взаимодействующих элементарных машин, оснащенных средствами коммуникаций и внешними устройствами. Элементарная машина (ЭМ) – это основной функциональный и структурный элемент ВС; конфигурация ЭМ допускает варьирование в широких пределах – от процессорного ядра до ЭВМ. Все основные ресурсы распределенных ВС (арифметико-логические устройства, память, средства управления и коммуникаций) являются логически и технически рассредоточенными. Число ЭМ в распределённых ВС допускает варьирование от десятков до сотен тысяч.

Перспективными считаются мультипрограммные режимы функционирования ВС [1] (обработка наборов задач и обслуживание потока задач), которые позволяют организовать разделение ресурсов между несколькими одновременно выполняющимися задачами.

Исследования пользовательских запросов показывают, что более 80% параллельных программ, выполняемых с использованием распределенных ВС, обладают свойством адаптивности – способностью подстраиваться под размер выделенной подсистемы из ЭМ. Существующие системы управления ресурсами распределенных ВС позволяют запускать адаптирующиеся программы, однако в них отсутствует возможность учитывать предпочтения пользователя на ранги выделяемых подсистем ЭМ.

* Работа выполнена при поддержке РФФИ (гранты № 09-07-90403, 09-07-13534, 09-07-00095, 08-07-00022), в рамках ФЦНТП (ГК 02.740.11.0006).

Множество вариантов подсистем ЭМ, допустимых для выполнения параллельной программы, может быть представлено в виде вектора рангов подсистем и сопоставленных им приоритетов. Это позволяет учитывать специфику параллельной задачи при оптимизации функционирования распределенных ВС.

Рассмотрим формальную постановку задачи.

Имеется распределенная ВС, состоящая из N ЭМ. На вход системы поступает набор из L задач. Каждая задача $j \in J = \{1, 2, \dots, L\}$ является параллельной адаптирующейся программой и характеризуется вектором параметров

$$P_j = (p_{j1}, p_{j2}, \dots, p_{jq_j}), \quad i=1, 2, \dots, q_j,$$

где параметр $p_{ji} = (r_{ji}, t_{ji}, \mu_{ji})$ описывает i -й вариант конфигурации ресурсов, допустимой для реализации задачи. Параметр r_{ji} – ранг задачи (количество ЭМ, необходимое для её решения, $r_{ji} \in C = \{1, 2, \dots, N\}$); t_{ji} – время, требуемое для решения задачи с использованием r_{ji} ЭМ; μ_{ji} задаёт приоритет i -й конфигурации ресурсов, $\mu_{ji} \in (0, 1]$. Чем выше значение μ_{ji} , тем более предпочтительным является выделение для решения задачи r_{ji} элементарных машин на время t_{ji} . Параметр q_j – количество параметров j -й задачи.

Считаем, что количество ЭМ в распределенной ВС не достаточно, чтобы одновременно решить все задачи набора.

Требуется построить расписание S решения параллельных задач на распределенной ВС. Для каждой задачи необходимо определить момент времени τ_j начала решения её ветвей и их распределение по элементарным машинам. Пусть x_j – номер выбранного параметра (конфигурации ресурсов) задачи $j \in J$ для её решения ($x_j \in \{1, 2, \dots, q_j\}$).

Обозначим через $x_{ji} \in C = \{1, 2, \dots, N\}$ – номер ЭМ, на которую распределена ветвь $i \in \{1, 2, \dots, r_{jx_j}\}$ задачи $j \in J$.

Введем функцию $J(t) = \{j \in J \mid \tau_j \leq t \leq \tau_j + t_j\}$ – множество задач, решаемых на распределенной ВС в момент времени t .

Будем называть расписание

$$S = (\tau_1, \tau_2, \dots, \tau_L; x_1, x_2, \dots, x_L; x_{11}, x_{12}, \dots, x_{1r_{1x_1}}, \dots, x_{L1}, x_{L2}, \dots, x_{Lr_{Lx_L}})$$

допустимым, если оно удовлетворяет ограничениям.

1) В любой момент времени на ресурсах распределенной ВС решается не более N ветвей параллельных задач: $\sum_{j \in J(t)} r_{jx_j} \leq N, \quad \forall t \in \mathbf{R}$.

2) Ветви параллельных задач решаются на разных элементарных машинах:

$$\prod_{j \in J(t)} \prod_{j' \in J(t) \setminus \{j\}} (x_{ji} - x_{j'i'}) \neq 0, \quad \forall t \in \mathbf{R}, \quad i = 1, 2, \dots, r_{jx_j},$$

$i' = 1, 2, \dots, r_{j'x_{j'}}$ Обозначим через Ω множество допустимых расписаний.

В качестве показателей оптимальности расписаний будем использовать время $T(S)$ окончания решения последней задачи и суммарную «удовлетворенность» пользователей $M(S)$:

$$T(S) = \max_{j \in J} \{\tau_j + t_j\}, \quad M(S) = \sum_{j=1}^L \mu_{jx_j}.$$

Введем составной показатель $F(S)$ эффективности расписания S :

$$F(S) = \alpha * M(S) / M_{\max} + \beta * T_{\min} / T(S),$$

где T_{\min} – нижняя оценка времени решения задач набора; M_{\max} – абсолютный максимум суммарного значения «предпочтений» пользователей для данного набора задач; α, β – весовые коэффициенты целевых функций.

Итак, требуется найти допустимое расписание $S \in \Omega$, доставляющее максимум целевой функции $F(S)$. Формально

$$F(S) = \alpha * M(S) / M_{\max} + \beta * T_{\min} / T(S) \rightarrow \max_{S \in \Omega} \quad (1)$$

при ограничениях:

$$\sum_{j \in J(t)} r_{jx_j} \leq N, \quad \forall t \in \mathbf{R}, \quad (2)$$

$$\prod_{j \in J(t)} \prod_{j' \in J(t) \setminus \{j\}} (x_{ji} - x_{j'i'}) \neq 0, \quad \forall t \in \mathbf{R}, \quad i = 1, 2, \dots, r_{jx_j}, \quad (3)$$

$$i' = 1, 2, \dots, r_{j'x_{j'}},$$

$$x_{ji} \in C, \quad j = 1, 2, \dots, L, \quad i = 1, 2, \dots, r_{jx_j}, \quad (4)$$

$$\tau_j \in \mathbf{R}, \quad j = 1, 2, \dots, L. \quad (5)$$

Задача (1)–(5) относится к дискретной оптимизации и является трудноразрешимой.

Разработан приближенный алгоритм решения задачи (1)–(5). В основу алгоритма положены эволюционные (генетические) методы оптимизации. Изначально строятся несколько допустимых расписаний («особей»), путем распределения задач набора по пакетам («генам»), с применением алгоритмов упаковки объектов в контейнеры (Best Fit, First Fit, Next Fit). При этом для каждой задачи при каждом распределении случайным образом назначается один из вариантов параметров (описанных вектором значений с приоритетами). Допустимое распи-

сание предполагает последовательное выполнение на распределенной ВС пакетов, объединяющих несколько задач, решаемых параллельно. Совокупность полученных расписаний формирует начальную «популяцию». Последующие шаги формирования итогового расписания эмитируют жизнедеятельность популяции и состоят из следующих этапов: развитие текущей популяции или образование «потомства» с использованием кроссинговера (алгоритма перетасовки генов) и «мутации» – формирования новой популяции. Процесс повторяется до тех пор, пока на протяжении определенного количества популяций не будет получено «потомство» (новая популяция) с лучшими значениями целевых функций. Таким образом, генетический алгоритм позволяет строить расписания, обеспечивающие субминимум времени выполнения задач набора и субмаксимум относительной «удовлетворенности» пользователей.

Алгоритм реализован на языке программирования C++. Моделирование и численные эксперименты разработанного алгоритма осуществлялись с использованием ресурсов пространственно-распределенной мультикластерной вычислительной системы Института физики полупроводников СО РАН и Центра параллельных вычислительных технологий ГОУ ВПО «Сибирский государственный университет телекоммуникаций и информатики». Поступающие задачи генерировались с помощью модели рабочей загрузки, предложенной W. Cirne и F. Berman [2] на основе статистики загрузки некоторых промышленных ВС. Приоритеты параметрам программ (требуемым ресурсам) задавались случайным образом с использованием равномерного распределения. Рассматривались наборы с количеством задач от 100 до 10000.

Результаты моделирования показали, что время поиска расписаний генетическим алгоритмом компенсируется уменьшением суммарного времени выполнения всех параллельных задач набора.

Литература

1. **Хорошевский В. Г.** Архитектура вычислительных систем. М.: МГТУ им. Н.Э. Баумана, 2008. 520 с.
2. **Cirne, W.** A Model for Moldable Supercomputer Jobs / Walfredo Cirne, Francine Berman // 15th Intl. Parallel & Distributed Processing Symposium. April 2001.
3. (<http://www.cs.huji.ac.il/labs/parallel/workload/models.html>)