

**Министерство образования и науки Российской Федерации
Российский фонд фундаментальных исследований
Южный научный центр Российской академии наук
Южный федеральный университет
НИИ многопроцессорных вычислительных систем
имени академика А.В. Каляева Южного федерального университета
АО «НИЦЭВТ»
ФГУП «НИИ «Квант»
ООО «НИЦ супер-ЭВМ и нейрокомпьютеров»
Журнал «Вестник компьютерных и информационных технологий»
Журнал «Известия ЮФУ. Технические науки»**

СУПЕРКОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

СКТ-2016

**Материалы 4-й Всероссийской
научно-технической конференции**

**19 – 24 сентября 2016 г.
Дивноморское, Геленджик**

ТОМ 2

**Ростов-на-Дону
Издательство Южного федерального университета
2016**

УДК004.272.43

ББК32.973

С 73

С 73 **Суперкомпьютерные технологии (СКТ-2016)** (19–24 сентября 2016 г.) Материалы 4-й Всероссийской научно-технической конференции : в 2 т. – Ростов-на-Дону : Издательство Южного федерального университета, 2016. – 304 с.
ISBN 978-5-9275-2038-1
ISBN 978-5-9275-2040-4 (Т. 2)

Во втором томе материалов 4-й Всероссийской научно-технической конференции «Суперкомпьютерные технологии» (СКТ-2016) представлены доклады по направлениям: распределенные вычисления и системы, облачные вычисления, применение суперкомпьютерных технологий в науке, технике и промышленности.

Издание осуществлено при финансовой поддержке Российского фонда фундаментальных исследований, проект № 16-07-20640 г.

М $\frac{2404000000}{6KO(03) - 2016}$ без объявл.

ББК 32.973

ISBN 978-5-9275-2040-4 (Т. 2)
ISBN 978-5-9275-2038-1

УДК 004.272.43
ББК 32.973

© Авторы докладов, 2016
© Научно-исследовательский институт
многопроцессорных вычислительных систем
имени академика А.В. Каляева федерального госу-
дарственного автономного образовательного учреж-
дения высшего профессионального образования
«Южный федеральный университет», составление,
оформление, 2016

ОЦЕНКА ВРЕМЕНИ ВЫПОЛНЕНИЯ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ПРИЛОЖЕНИЙ С ПОТОКОВОЙ СТРУКТУРОЙ С УЧЕТОМ ИХ ВЫЧИСЛИТЕЛЬНЫХ ПРОФИЛЕЙ*

*Южно-Уральский государственный университет, г. Челябинск,
gleb.radchenko@susu.ru, leonid.sokolinsky@susu.ru*

Характеристики производительности вычислительной машины

Определим характеристику производительности вычислительной машины как отображение:

$$\pi: m \rightarrow \mathbb{Z}_{>0},$$

где m – образ это виртуальной машины, существующий в вычислительной системе. Примерами характеристик производительности могут служить числовые характеристики машины, результаты синтетических тестов или результаты тестового выполнения конкретных классов функций с заранее определенными наборами входных данных.

Очевидно, что для качественного прогнозирования параметров выполнения задач на заданных машинах нам необходимо учитывать максимально возможное количество характеристик производительности, включая такие характеристики, как количество доступных процессоров; частота процессора; скорость обмена данными с жестким диском; характеристика машины по LINPACK и др. Таким образом, определим вектор характеристик производительности виртуальных машин, развернутых в облачной вычислительной системе $\mathcal{C}$:

$$\Pi = [\pi_1, \pi_2 \dots \pi_r].$$

Каждой машине $m \in \mathcal{M}$ облачной вычислительной системы $\mathcal{C}$ сопоставим вектор характеристик производительности, отражающий значения производительности вычислительной машины:

$$\Pi: \mathcal{M} \rightarrow \mathbb{Z}_{>0}^r.$$

В рамках предоставления вычислительных ресурсов каждой задаче выделяется одна либо несколько виртуальных машин. Прямого доступа к узлам вычислительной системы не обеспечивается.

Методы оценки времени выполнения заданий в рамках проблемно-ориентированной среды

Особенностью проблемно-ориентированной облачной вычислительной среды является то, что она использует информацию об особен-

* Исследование выполнено при финансовой поддержке РФФИ (проект № 15-29-07959 офи-м)

ностях классов решаемых задач при планировании и распределении вычислительных ресурсов. Будем требовать, чтобы в рамках проблемно-ориентированной вычислительной среды, для каждого класса задач были определены следующие функции для прогноза процесса выполнения задачи в зависимости от значений входных параметров:

- 1) функция оценки объема выходных данных при определенных входных параметрах;
- 2) функция оценки времени выполнения задачи при определенных входных параметрах на машине с указанным вектором характеристик производительности.

Таким образом, для каждой функции $f \in \mathcal{F}$ из предметной области, выполняющейся в проблемно-ориентированной среде $\mathcal{E}$, необходимо реализовать оценку следующих характеристик:

- 1) объем ожидаемого выхода $v(f, \mathcal{J}^{in})$ – это функция, возвращающая ожидаемый общий размер в байтах всех выходных информационных объектов $\mathcal{J}^{out}$:

$$v(f, \mathcal{J}^{in}) = |\mathcal{J}^{out}| = \sum_{\forall l \in \mathcal{J}^{out}} \sum_{\forall b \in l} |b|.$$

- 2) ожидаемое время выполнения функции $\tau(f, \mathcal{J}^{in}, \Pi)$, возвращающая оценочное время выполнения (в секундах) функции f при заданном множестве входных информационных объектов $\mathcal{J}^{in}$ на машине, с вектором характеристик производительности Π :

$$\tau: (f, \mathcal{J}^{in}, \Pi) \rightarrow \mathbb{N}.$$

Время выполнения функции f на конкретной машине с вектором характеристик производительности Π можно представить в виде функции, зависящей от вектора входных информационных объектов $\mathcal{J}^{in}$. К сожалению, невозможно оценить время выполнения функции с идеальной точностью, так как вычислительная работа подготовки набора выходных информационных объектов $\mathcal{J}^{out}$ может косвенно зависеть от множества факторов, которые наша модель учесть не может (возможные фоновые процессы, качество предсказания ветвления конкретной версии процессора, объем занятого кэша и др.). Для компенсации данной ошибки, оценку времени выполнения функции можно смоделировать в виде случайной величины:

$$\chi(f, \Pi, \mathcal{J}^{in}) = \tau(f, \Pi, \mathcal{J}^{in}) + \alpha,$$

где $\tau(f, \Pi, \mathcal{J}^{in})$ – детерминированная функция, представляющая зависимость времени выполнения функции f на машине с вектором характеристик производительности Π от вектора входных информационных объектов $\mathcal{J}^{in}$, α – это стохастическая величина с нулевым математическим ожиданием ($M[\alpha] = 0$), представляющая факторы, не входящие в разрабатываемую модель.

Установим ограничение на нашу облачную систему, что конкретный проблемно-ориентированный сервис (функция) реализуется на базе

виртуальной машины с фиксированными вычислительными возможностями Π_0 . В этом случае для оценки $\hat{M}[\chi(f, \Pi_0, J^{in})]$ можно применить метод k ближайших соседей (k -nearest neighbor algorithm) k -NN, основанный на анализе времени выполнения предыдущих функций того же типа на той же с близкими значениями параметров:

$$\hat{M}[\chi(f, \Pi_0, J^{in})] = \hat{t}(f, \Pi_0, J^{in}) = \frac{1}{k} \sum_{i=1}^k W_i(J^{in}) t_{J_i^{in}, \Pi_0}^f, \quad (1)$$

где $\hat{t}(f, \Pi_0, J^{in})$ – взвешенная средняя оценка времени выполнения функции f с входными параметрами J^{in} на основе k предыдущих наблюдений о времени выполнения $t_{J_i^{in}, \Pi_0}^f$ функции f при входных параметрах J_i^{in} на машине с производительностью Π_0 . Весовая функция $W_i(J^{in})$ присваивает большие веса к времени обработки сообщений, значения параметров которых ближе к значению входных параметров J^{in} .

Оценку $\sigma[\chi(f, \Pi_0, J^{in})]$ можно провести на основе аналогичного подхода:

$$\sigma^2[\chi(f, \Pi_0, J^{in})] = \frac{1}{k} \sum_{i=1}^k W_i(J^{in}) \left(t_{J_i^{in}}^f - \hat{t}(f, \Pi_0, J^{in}) \right)^2. \quad (2)$$

Для того чтобы учесть возможность выполнения функции f на виртуальной машине с вычислительными возможностями, отличными от Π_0 , расширим определение вектора параметров для оценки, добавив к вектору входных параметров вектор производительности виртуальной машины $\Pi = [\pi_1, \pi_2 \dots \pi_r]$. Таким образом, будем считать, что в качестве характеристики запуска используется вектор P размерности $n + r$, где n – число входных параметров запуска функции f , r – число характеристик производительности виртуальной машины, определенный следующим образом:

$$P = [I_1^{in}, \dots, I_n^{in}, \pi_1, \dots, \pi_r].$$

В этом случае оценку (1), (2) можно преобразовать к следующему виду:

$$\begin{aligned} \hat{M}[\chi(f, \Pi, J^{in})] &= \hat{t}(f, \Pi, J^{in}) = \frac{1}{k} \sum_{i=1}^k W_i(P) t_{P_i}^f, \\ \sigma^2[\chi(f, \Pi, J^{in})] &= \frac{1}{k} \sum_{i=1}^k W_i(P) \left(t_{P_i}^f - \hat{t}(f, \Pi, J^{in}) \right)^2, \end{aligned}$$

где $t_{P_i}^f$ – значение предыдущего наблюдения времени выполнения функции f с вектором характеристик P_i .

При отсутствии полной и достоверной информации о значениях входных параметров, оценка времени выполнения функции может производиться в режиме «средней оценки», на основе усредненных значений вы-

полнения данной функции в прошлом, с учетом значений производительности виртуальной машины. В этом случае вектор P приобретает вид:

$$P = [\pi_1, \dots, \pi_r].$$

Аналогично построим оценку объема выходных данных для функции f :

$$\hat{v}(f, j^{in}) = \frac{1}{k} \sum_{i=1}^k W_i(j^{in}) v_{j_i^{in}}^f,$$

где $\hat{v}(f, j^{in})$ – оценка объема выходных параметров функции на основе информации об выходных объемах k предыдущих запусков функции f с параметрами j_i^{in} с использованием весовой функции W .

Таким образом, для оценки времени выполнения и объема выходных данных отдельного задания необходимо хранить информацию об истории выполнения аналогичных заданий, выполняемых проблемно-ориентированной средой.

Методы оценки времени выполнения потоковых приложений в проблемно-ориентированных вычислительных средах

Метод оценки времени выполнения приложений в виде размеченного взвешенного ориентированного ациклического графа основывается на модели вычислительной среды, описанной в [1]. Разработанный метод оценки времени выполнения обеспечивает зависимость времени выполнения от выделенных ресурсов и входных параметров соответствующих действий и предусматривает возможность запуска на одном многоядерном ускорителе двух и более подзадач одновременно.

Пусть задан распланированный граф задания $G = \langle T, E, init, fin, \delta, \gamma, \omega, \xi \rangle$ и его простой путь $y = (e_1, e_2, \dots, e_n)$. Сложностью пути $u(y)$ назовем величину

$$u(y, j^{in}) = \hat{t}(f_{fin}, \Pi_{fin}, j^{in}) + \sum_{i=1}^n \left(\hat{t}(f_i, \Pi_i, j^{in}) + \omega_i * \hat{v}(f_i, j^{in}) \right),$$

где $\hat{t}(f_{fin}, \Pi_{fin}, j^{in})$ – функция оценки времени вычисления функции f на виртуальной машине с производительностью Π_{fin} , учитывающая зависимость времени выполнения от набора входных параметров; $j^{in}, \hat{v}(f, j^{in})$ – функция оценки выходного объема данных функции f при условии входных параметров j^{in} ; w_i – весовой коэффициент преобразования объема выходных параметров во время передачи данных между узлами (пропускная способность ребра).

Другими словами, сложность пути вычисляется как сумма времени выполнения всех задач, лежащих на этом пути, и времени передачи данных между задачами, входящими в различные кластеры. Время передачи любого объема данных между узлами, принадлежащими одному кластеру, равно нулю, а время передачи данных между узлами, принад-

лежащими разным кластерам, пропорционально объему передаваемых данных с коэффициентом 1.

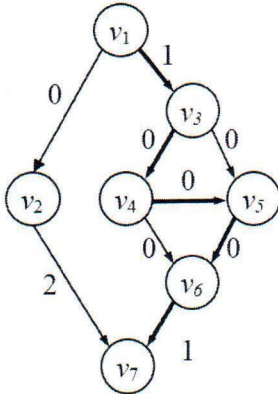


Рис. 1. Распланированный граф задания и его доминирующая последовательность

Пусть Y – множество всех простых путей в распланированном графе $G = \langle V, E, init, fin, \delta, \gamma, \omega, \xi \rangle$. Простой путь $\bar{y} \in Y$ называется *доминирующей последовательностью*, если

$$u(\bar{y}) = \max_{y \in Y} u(y),$$

т.е. доминирующая последовательность обладает максимальной сложностью. Пример доминирующей последовательности (помечен жирными стрелками) приведен на рис. 1.

Параллельным временем выполнения графа задания называется сложность доминирующей последовательности.

На рис. 2 приведен псевдокод алгоритма оценки времени выполнения потоковых приложений в виде размеченного взвешенного ориентированного ациклического графа.

```

 $u(\bar{y}) = 0$ ; // общее время выполнения графа задания
while (есть непомяченные вершины) {
    Поиск нового пути  $y$ ;
    Расчет сложности  $u(y)$  пути  $y$ ;
    Пометить вершины пути  $y$  как рассмотренные;
     $y \in Y$ ; // включить  $y$  во множество  $Y$ 
}
Найти максимальное значение по всем  $u(y)$  и поместить
его в  $u(\bar{y})$ ;
return  $u(\bar{y})$ ;

```

Рис. 2. Алгоритм оценки времени выполнения потоковых приложений в проблемно-ориентированных вычислительных средах

1. Sokolinsky L.B., Shamakina A.V. Methods of Resource Management in Problem-Oriented Computing Environment // Programming and Computer Software. – 2016. – Vol. 42. – No. 1. – P. 17 – 26.