УДК 004.415.2.052.03

И. В. Ковалев, П. В. Ковалев, В. С. Скориков, С. Н. Гриценко

ОЦЕНКА ВРЕМЕНИ ВЫПОЛНЕНИЯ МУЛЬТИВЕРСИОННЫХ ПРОГРАММ НА КЛАСТЕРЕ С ПОСЛЕДОВАТЕЛЬНОЙ И ПАРАЛЛЕЛЬНОЙ АРХИТЕКТУРОЙ ОБМЕНА ДАННЫМИ

Эффективность использования неспециализированного гетерогенного кластера для реализации мультиверсионного программного обеспечения напрямую зависит от умения учитывать и планировать его нагрузку. Повысить точность прогнозирования времени выполнения мультиверсионных программ позволяют стохастические методы оценки времени их выполнения. В качестве математической основы для такой оценки предлагается применение GERT-сетей.

Ключевые слова: кластер, надежность, мультиверсионное программирование.

Методология мультиверсионного программирования подразумевает параллельное выполнение набора программ (мультиверсий) с последующим голосованием и принятием решения о правильности выполнения задачи (или алгоритма). Параллельные вычисления способны улучшить реализацию мультиверсионного программного обеспечения (ПО) и решить часть проблем, связанных с временными и ресурсными ограничениями вычислительных средств отказоустойчивых информационно-управляющих систем [1–5]. В данной статье представлен один из способов оценки времени выполнения вычислений для самой простой и дешевой архитектуры суперкомпьютера – кластера.

Под *кластером* будем понимать связанный набор полноценных компьютеров, используемый в качестве единого ресурса. *Полноценный компьютер* – это завершенная компьютерная система, обладающая всем необходимым для ее функционирования, включая процессоры, память, подсистему ввода-вывода, а также операционную систему, подсистемы, приложения и т. д. А *единый ресурс* означает наличие программного обеспечения, дающего возможность пользователям, администраторам и даже приложениям считать, что имеется только одна сущность – кластер [4]. Наиболее популярными (в частности, в России) сейчас являются библиотеки MPI, PVM и DVM [5; 6]. Тесты производительности кластеров, приведенных в [6], показывают, что библиотека MPI обладает наибольшей производительностью при использовании детерминированных схем вычислений. Однако узкими местами данной библиотеки являются трудоемкость разработки и отсутствие устойчивости к программно-аппаратным отказам. В случае сбоя вычисления возобновляются с последней контрольной точки (если это было учтено при разработке). Таким образом, кластер, работающий с библиотекой MPI, должен обладать высокой стабильностью работы. Таким требованиям отвечают специализированные кластеры, список которых представлен в [6].

Для расчета в гетерогенных кластерных системах, а также в кластерах, не обладающих высокой надежностью узлов и отказоустойчивостью при различных сбоях, используются библиотеки типа Condor. В частности, эти библиотеки применяют в составе единого кластера узлы, не только различающиеся в аппаратной части, но и работающие в разных операционных системах. Таким образом, библиотека Condor позволяет использовать для вычислений существующую компьютерную технику и уже имеющиеся коммуникации, тем самым существенно удешевляя стоимость создания кластера [5; 7–9].

В данной статье будут рассмотрены два метода аналитической оценки времени выполнения мультиверсионной программы в подобных кластерах [10].

Постановка задачи. Ниже мы будем рассматривать вариант организации параллельных вычислений для алгоритма перемножения двух матриц размерности $N \times N$ на K процессорах, описываемый для узла процессора следующим образом:

1) получение первой матрицы;

2) получение полосы от второй матрицы (вторая матрица режется на полосы для каждого из узлов);

3) вычисление полосы результирующей матрицы;

4) возврат данных.

Аппаратные реализации кластера могут быть весьма разнообразны [11]. Ниже будут представлены два варианта: кластер с последовательной архитектурой обмена данными (последовательной шиной передачи данных) (рис. 1) и кластер с параллельной архитектурой обмена данными (параллельной шиной передачи данных) (рис. 2).

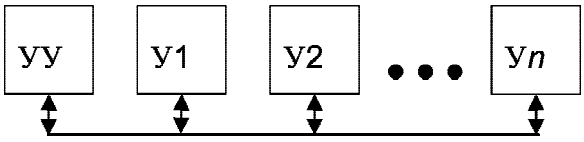

Рис. 1. Кластер с последовательной шиной передачи данных

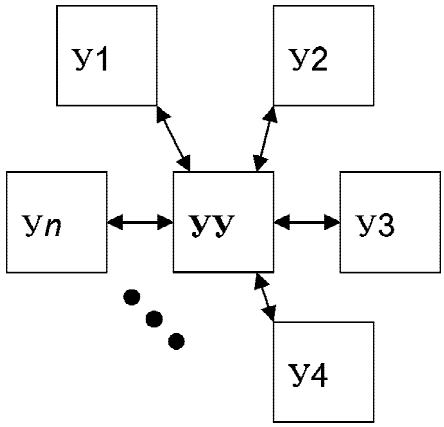

Рис. 2. Кластер с параллельной шиной передачи данных

Принципиальным отличием данных архитектур является то, что в первом случае узлы получают задание по очереди, друг за другом, тогда как во втором – одновременно. И в том, и другом случае кластер обладает одним управляющим узлом (УУ) и несколькими вычисляющими (У*n*)

Детерминированная оценка времени выполнения вычислений. Будем оценивать время вычисления произведения двух матриц большой размерности при условии отсутствия сбоев в работе кластера. Используем следующие данные:

– размерность матрицы N – 10 000;

– производительность узла H – 1,00E + 09 тактов в 1 с;

– производительность сети F – 6,00E + 06 б/с;

– количество байт на число D – 8;

– тактов на операцию *Tick* – 30.

Размерность матрицы N была выбрана таким образом, чтобы вычисления были ощутимы по времени. Оценочные измерения проводились на P-III 1000 Mhz. Поэтому производительность узла H выбрана 1e + 9. Очевидно, что производительность является не единственным фактором, влияющим на скорость вычисления с такими большими массивами данных, поэтому опытным путем был рассчитан коэффициент тактов на операцию *Tick* (оценка производилась на восьми компьютерах с различной конфигурацией).

Под *операцией* будем понимать вычисление базового выражения $c_{i,j} = c_{i,j} + a_{i,k} \cdot b_{k,j}$. Количество байт на число D – это объем памяти, необходимый для хранения одного вещественного числа. Производительность сети F соответствует скорости передачи данных в сети 100 Мб.

Будем считать, что все вычисления выполняются в памяти компьютера без использования файла подкачки.

Время вычисления на одной ЭВМ

$$T\_one = N^2 \cdot N \cdot Tick / H = 30\,000 \text{ с (примерно 8,3 ч).}$$

Формулы оценки времени вычислений на кластере с *последовательной шиной* передачи данных будут следующие:

$$No = (N^2 / K)\, N,$$

$$SN = N^2 + N^2/K,$$

$$RN = N^2/K,$$

$$To = No \cdot Tick / H,$$

$$St = SN \cdot D / F,$$

$$ST = St \cdot K,$$

$$Rt = RN \cdot D / F,$$

$$T = To + ST + Rt,$$

где No – количество операций на узле; SN и RN – количество чисел, переданных узлу и возвращенных узлом соответственно; To – время вычисления на узле; St и Rt – время передачи данных узлу и от узла соответственно; ST – совокупное время передачи данных (время, через которое последний узел в очереди приступит к вычислениям); T – общее время вычисления на кластере.

Таким образом, мы можем вычислить коэффициент ускорения вычислений $Kt = T\_one / T$ и коэффициент эффективности использования процессоров $Ke = Kt / (K + 1)$. В предельном случае, если считать временные потери на передачу данных равными 0, $Kt = K$ и $Ke = K /(K + 1)$.

Результаты вычислений на кластере с последовательной архитектурой передачи данных для разного количества процессоров (узлов) представлены в [10].

Формулы для оценки времени выполнения мультиверсионной программы на кластере с параллельной шиной передачи данных совпадают с формулами оценки времени последовательных вычислений, за исключением совокупного времени передачи данных *ST*, которое в этом случае равно *St*. Результаты вычислений на кластере с параллельной архитектурой передачи данных для разного количества процессоров (узлов) приведены в [11]. В этой же работе сделаны следующие выводы:

– для последовательной шины передачи данных существует так называемая точка насыщения – количество процессоров, дающих наименьшее время вычисления. Дальнейшее увеличение количества процессоров не увеличивает производительность кластера, а уменьшает ее. Причина очевидна: время, необходимое на передачу данных узлам кластера, оказывается значительнее времени полезного счета. В нашем случае показатель $K = 15$;

– для параллельной шины передачи данных такой точки насыщения нет, так как $T(K)$ – монотонно убывающая функция при K, стремящемся к бесконечности;

– для последовательной и параллельной шин существует такое значение K, при котором все узлы используются наиболее эффективно. При дальнейшем увеличении количества процессоров прирост производительности будет постоянно снижаться.

Обобщением этих выводов является закон Амдала [1; 6; 8–11]. Однако полученные результаты имеют два отличия, связанные с архитектурными особенностями кластера: во-первых, в качестве времени вычисления на одном компьютере учитывается время на загрузку данных для расчетов; во-вторых, при расчете коэффициента эффективности использования процессоров не учитывается применение выделенного управляющего узла кластера.

Оценка времени выполнения вычислений при помощи стохастической сети. Одной из главных проблем выполнения мультиверсионного ПО и, следовательно, параллельных вычислений является надежность всех узлов системы и стоимость этой надежности. Технология Condor позволяет использовать в составе вычислительного кластера произвольные компьютеры, объединенные в единый пул. Более того, вычисления могут производиться во время простоя (*Idle*) компьютера, т. е. когда пользователь не проявляет активности. В случае если вычисления были прерваны (при перезагрузке, прерывании вычислений и пр.), задача переносится на другой узел. Для простоты оценки будем считать, что вычислительная система состоит из одинаковых узлов, хотя данный метод легко переносится и на гетерогенный кластер.

Для оценки используем стохастические GERT-сети [7; 10–12]. Под *стохастической сетью* будем понимать ориентированный граф $G = (N, A)$ с узлами определенного типа. Узлы стохастической сети могут быть интерпретированы как состояния системы, а дуги – как переходы из одного состояния в другое. Каждый внутренний узел

состоит из двух функций: входной и выходной. Входная функция отвечает за условие активации узла, а выходная – за результат его активации. Начальные узлы (источники) выполняют только функцию выхода, а конечные узлы (стоки) – только функцию входа. Типы входных и выходных функций рассмотрены в [7; 10; 12].

Произведем оценку среднего времени выполнения вычислений на кластере, допускающем отказ его узлов. Рассмотрим стохастическую схему выполнения вычислений (рис. 3).

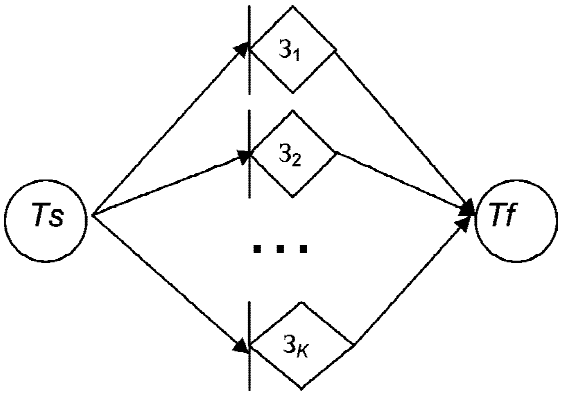

Рис. 3. Стохастическая схема вычислений

Каждая задача $З_i$ вычисляется на отдельном узле, который функционирует следующим образом (рис. 4).

Номера узлов графа на этой схеме соответствуют состояниям узла кластера, а дуги – действиям (табл. 1).

Данная сеть удовлетворяет свойствам A1...A6 [12], следовательно она является допустимой GERT-сетью, но не является EOR-сетью из-за AND-функции входа у стока, которая существенно усложняет вычисления и ограничивает применение методов расчета [10; 12]. Однако мы можем разбить GERT-сеть на подсети и вычислить ожидаемое время выполнения стока *Tf* как

$$E(Tf) = \max E(T_{3i})$$

для всех *i* от 1 до *K*.

Полученные подсети являются STEOR GERT-сетями, для которых существуют достаточно простые алгоритмы вычисления времени выполнения сети.

Для расчета характеристик сети воспользуемся алгоритмом, предложенным в [7].

Произведем расчет математического ожидания и дисперсии времени выполнения стохастической сети *N*. Для расчета добавим обратную дугу $A = <5, 1>$ (рис. 5).

Запишем для этой сети топологическое уравнение Мейсона (или правило Мейсона):

$$W_{\text{E}} = \frac{1}{W_A},$$

$$H = 1 - W_0W_1W_2W_5 / W_E - W_1W_3W_4 = 0,$$

$$W_E(s) = \frac{W_0W_1W_2W_5}{1 - W_1W_3W_4}.$$

Пусть

$$Q_1(s) = W_0W_1W_2W_5,$$
$$Q_2(s) = W_1W_3W_4.$$

Тогда

$$W_E = \frac{Q_1}{1 - Q_2},$$

$$\mu_{1E} = \frac{\partial^1}{\partial s^1} M_E(s) = \frac{1}{W_E(0)} \frac{Q_1'(1 - Q_2) + Q_1Q_2'}{(1 - Q_2)^2},$$

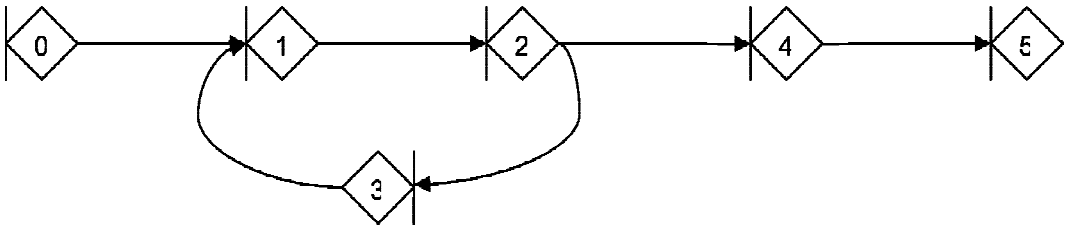

Рис. 4. Схема функционирования узла (обозначения см. в тексте)

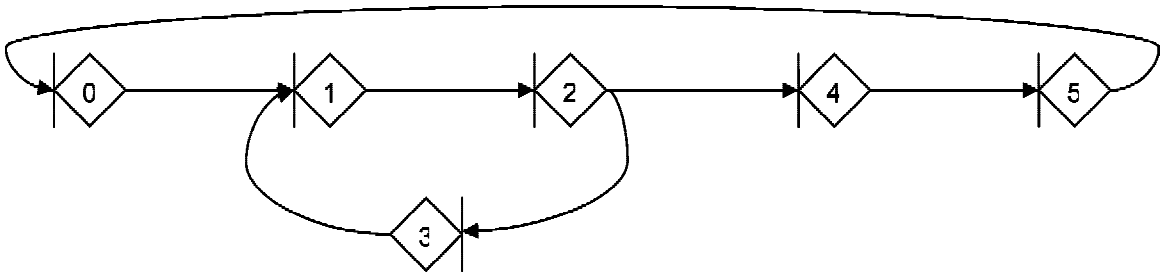

Рис. 5. Измененная GERT-сеть

Таблица 1

№ действия	Действие	Описание
0	<0, 1>	Ожидание в очереди момента получения данных
1	<1, 2>	Получения данных с головного узла кластера (ГУК)
2	<2, 4>	Выполнение вычислений, завершившихся успехом
5	<4, 5>	Возврат данных
3	<2, 3>	Ошибка в ходе выполнения вычислений
4	<3, 1>	Устранение сбоя или замена компьютера

$$\mu_{2E} = \frac{\partial^2}{\partial s^2} M_E(s) = \frac{1}{W_E(0)} \times$$

$$\times \frac{(Q_1''(1-Q_2) + Q_1 Q_2'')(1-Q_2) + 2Q_2'(Q_1'(1-Q_2) + Q_1 Q_2')}{(1-Q_2)^3},$$

$$E_E = \mu_{1E},$$

$$\sigma_E = \sqrt{\mu_{2E} - \mu_{1E}^2}.$$

ведем *W*-функции для рассматриваемой нами GERT-сети. С этой целью введем переменную, характеризующую среднее время безотказной работы узла кластера $T_р$. Для разных кластеров эта величина различна, но именно она вносит недетерминированность в метод оценки времени выполнения расчетов.

Приведем таблицу вероятностей перехода и распределений в GERT-сети [2] (табл. 2).

Тогда имеем табл. 3.

Оценим полученные нами параметры применительно к задаче перемножения двух больших матриц. Предположим, что исследуемый кластер основан не на выделенных машинах, а на классе общего доступа университета, и время непрерывной работы узла составляет около 12 ч, т. е. компьютеры используются преподавателями и студентами 12 ч. Тогда $T_р = 12 \cdot 3\,600 = 43\,200$ с. Пусть допустимое время ожидания отклика узла составляет 60 с, после этого управляющий узел переносит задачу на другой узел. Таблица констант (время дано в секундах) имеет следующий вид (табл. 4).

Если мы используем параллельную шину передачи данных, то задачи не стоят в очереди, а запускаются одновременно, и, следовательно, $T_0 = 0$ для любых *k*. Результаты вычислений на кластере с параллельной и последовательной архитектурой передачи данных для разного количества процессоров (узлов) представлены в [10].

Сопоставляя результаты детерминированной и стохастической оценок времени параллельного выполнения мультиверсионной программы, следует обратить внимание на их существенное различие при малом количестве узлов (процессоров). Увеличение оценки ожидаемого времени выполнения с использованием стохастической сети связано с влиянием вероятности возникновения сбоя в процессе вычислений. Очевидно, что чем больше время вычислений на конкретном узле, тем выше вероятность того, что возникнет состояние ошибки и потребуется перезапуск вычислений. В качестве решения этой проблемы можно предложить разбиение задач с целью уменьшения времени счета либо обмен промежуточными (или частичными) результатами с возможностью возобновления вычислений в случае ошибки.

В данной статье на простой задаче были продемонстрированы возможности стохастического метода оценки времени выполнения параллельных вычислений при реализации мультиверсионных программ на кластерах под управлением систем, подобных библиотеке Condor. Полученные результаты показывают, что при условии достаточности доступных ресурсов, возможности умеренного распараллеливания задачи и небольшого (сопоставимого со временем доступности узлов) времени вычис-

Таблица 2

№ действия	Действие	Описание	p_i	$f_i(t)$
0	<0, 1>	Ожидание в очереди момента получения данных	1	Постоянная величина $T_0 = k \cdot T_1$, где k – номер в очереди; T_1 – время получения данных с УУ
1	<1, 2>	Получения данных с головного узла кластера	1	Постоянная величина T_1
2	<2, 4>	Выполнение вычислений, завершившихся успехом	$P = 1 - m_2/T_р$	Нормальное распределение $N(m_2, d_2)$, где m_2 – ожидаемое время вычисления подзадачи; d_2 – предполагаемые допустимые отклонения, например 10 % от m_2
3	<2, 3>	Ошибка в ходе выполнения вычислений	$1 - p$	Нормальное распределение $N(0 + m_2)/2$, $(m_2 - 0)/5)$, так как равномерное распределение на [0; m_2] недопустимо в данном методе
4	<3, 1>	Устранение сбоя или замена вычисляющего узла	1	Постоянная величина T_4
5	<4, 5>	Возврат данных	1	Постоянная величина T_5 – время передачи данных с узла кластера на ГУК

Таблица 3

№ действия	Действие	Описание	p_i	$M_i(s)$
0	<0, 1>	Ожидание в очереди момента получения данных	1	$\exp(sT_0)$
1	<1, 2>	Получения данных с головного узла кластера	1	$\exp(sT_1)$
2	<2, 4>	Выполнение вычислений, завершившихся успехом	$p = 1 - m_2/T_р$	$\exp(sm_2 + 0{,}5s^2d_2^2)$
3	<2, 3>	Ошибка в ходе выполнения вычислений	$q = 1 - p$	$\exp(sm_3 + 0{,}5s^2d_3^2)$
4	<3, 1>	Устранение сбоя или замена вычисляющего узла	1	$\exp(sT_4)$
5	<4, 5>	Возврат данных	1	$\exp(sT_5)$

Таблица 4

$T_р$	T_1	m_2	d_2	T_4	T_5
43 200	*St*	*T*o	0,1·*T*o	60	*Rt*

ления каждой подзадачи на узле использование таких систем вполне оправданно. Также следует отметить, что разницы во времени вычислений на кластерах с параллельной и последовательной архитектурой обмена данных при использовании более 10 узлов практически нет.

В дальнейшем авторы планируют рассмотреть более сложные архитектуры мультиверсионного ПО и стратегии организации вычислений, такие как оценка времени выполнения вычислений очереди задач (в частности, для реализации RB-блока [13]), расчет времени вычислений по графу распараллеливания мультиверсионной программы и т. п.

Библиографический список

1. Букатов, А. А. Программирование многопроцессорных вычислительных систем / А. А. Букатов, В. Н. Дацюк, А. И. Жегуло. Ростов н/Д : Изд-во ООО ЦВВР, 2003.

2. Вентцель, Е. С. Теория вероятностей и ее инженерные приложения : учеб. пособие для втузов / Е. С. Вентцель, Л. А. Овчаров. 2-е изд., стер. М. : Высш. шк., 2000.

3. Лебедев, В. А. Параллельные процессы обработки информации в управляющих системах / В. А. Лебедев, Н. Н. Трохов, Р. Ю. Царев ; НИИ систем упр., волновых процессов и технологий. Красноярск, 2001.

4. Шнитман, В. Современные высокопроизводительные компьютеры [Электронный ресурс] / В. Шнитман // Информ.-аналит. материалы Центра информ. технологий. Электрон. дан. 1996. Режим доступа: http://www.citforum.ru/hardware/svk/contents.shtml. Загл. с экрана.

5. Pfister, G. Sizing Up Parallel Architectures [Electronic resource] / G. Pfister. Electronic data. 1998. Access mode: http://www.dbpd.com/vault/9805feat.htm ; http:/www.citforum.ru/hardware/articles/art_5.shtml. Title from display.

6. Шпаковкий, Г. И. Программирование для многопроцессорных систем в стандарте MPI / Г. И. Шпаковкий, Н. В. Серикова. Минск : Изд-во Белорус. гос. ун-та, 2002.

7. Филлипс Д. Методы анализа сетей / Д. Филлипс, А. Гарсиа-Диас. М. : Мир, 1984.

8. Shi, Yu. Reevaluating Amdahl's Law and Gustafson's Law [Electronic resource] / Yu. Shi. Electronic data. Access mode: http://www.cis.temple.edu/~shi/docs/amdahl/amdahl.htm. Title from display.

9. Thain, D. Distributed Computing in Practice: The Condor Experience / D. Thain, T. Tannenbaum, Miron Livny ; University of Wisconsin. Madison, 2004.

10. Ковалев, И. В. Модели оценки времени выполнения задачи на кластере с последовательной и параллельной архитектурой обмена данными / И. В. Ковалев, Д. М. Письман, М. Ю. Слободин // Системы упр. и информ. технологии. 2005. № 3 (20). С. 58–62.

11. Письман, Д. М. Анализ временных параметров сетевых моделей на базе модифицированной ГЕРТ-сети / Д. М. Письман // Проблемы машиностроения и автоматизации. 2006. № 1. С. 18–26.

12. Neumann, K. Stochastic Project Networks. Temporal Analysis, Scheduling and Cost Minimization / K. Neumann. New York : Springer Verlag,1990.

13. Kovalev I. System of Multi-Version Development of Spacecrafts Control Software / I. Kovalev. Sinzheim : Universitate Verlag, 2001.

I. V. Kovalev, P. V. Kovalev, V. S. Skorikov, S. N. Gritsenko

TIME EVALUATION OF MULTI-VERSION PROGRAMS EXECUTING BY CLUSTER SYSTEM USING SERIAL AND PARALLEL ARCHITECTURE OF DATE EXCHANGE

The effectiveness of using non-specialized heterogeneous cluster for the multi-version software realization directly depends on the ability to control and plan its loading. Stochastic methods of the multi-version programs time evaluation can raise precision of execution time prediction. As a mathematical base of such evaluation ones offer to use GERT-networks.

Keywords: cluster system, reliability, multi-version programming.