УДК 61.681

МОДЕЛЬ УЗЛА GRID-СИСТЕМЫ НА ОСНОВЕ СЕТЕЙ ПЕТРИ

А.Ю. Шелестов
Институт космических исследований НАНУ-НКАУ

В доповіді запропоновано модель вузла Grid-системи на основі апарату мереж Петрі. Доведено основні властивості моделі: живучість, обмеженість, взаємне виключення та рівноправність.

Grid-система представляет собой распределенную систему доступа и обработки данных, состоящую из следующих основных компонентов: пользователи, формирующие запросы к системе, вычислительные узлы, выполняющие обработку данных, хранилища данных и управляющие узлы.

Задача состоит в построении Grid-системы, обеспечивающей прозрачную для пользователя обработку его запросов на поиск и обработку данных, в том числе, синхронизацию доступа к необходимым данным, обработку этих данных и предоставление пользователю результатов обработки. Такая Grid-система относится к Grid-системам смешанного типа (является одновременно и вычислительной и информационной [1]), поскольку она включает как высокопроизводительные вычислительные узлы так и обеспечивает доступ к данным распределенных хранилищ [**2**].

Синхронизация доступа к данным и их корректная обработка предполагает выполнение некоторых базовых свойств, а именно:

взаимное исключение (mutex) — синхронизация событий при обработке данных: два или более вычислительных узла не могут одновременно иметь доступ к общей области данных;

равноправие (fairness) — отсутствие дискриминации заданий. Если пользователь сформировал запрос (задание) и отправил его в систему, то обязательно наступит момент, когда это задание начнет выполняться и будет выполнено.

Работу Grid-системы смешанного типа без детализации можно проиллюстрировать рис. 1. Из рис. 1 видно, что управляющий узел обеспечивает синхронизацию при распараллеливании задания между ресурсами или при выполнении операций доступа к общей памяти.

Учитывая высокую стоимость компонентов Grid-систем, прежде чем приступать к построению системы, обладающей указанными свойствами, необходимо построить ее модель и исследовать свойства этой модели. В докладе рассматривается модель функционирования вычислительного узла, работающего над общей памятью, и исследуются свойства этой модели.

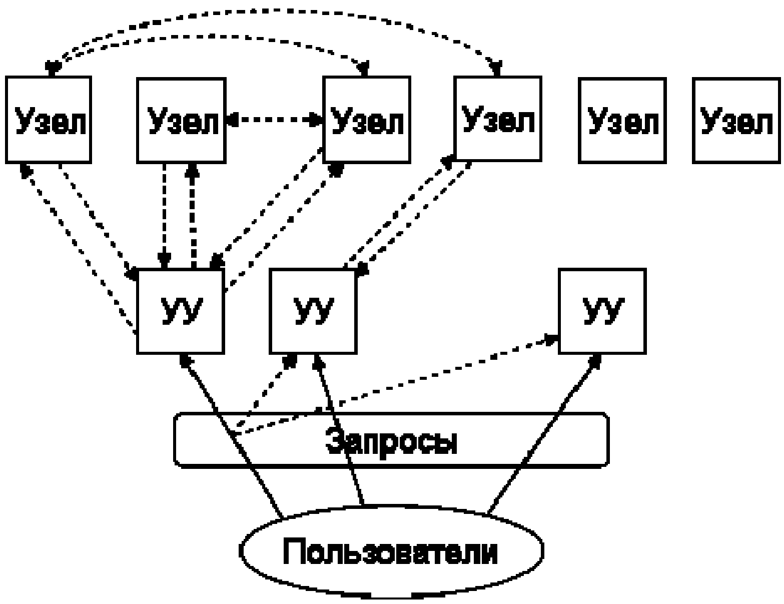

Рис. 1. Модель информационных потоков в Grid-системе

Учитывая высокую стоимость компонентов Grid-систем, прежде чем приступать к построению системы, обладающей указанными свойствами, необходимо построить ее модель и исследовать свойства этой модели. В докладе рассматривается модель функционирования вычислительного узла, работающего над общей памятью, и исследуются свойства этой модели.

Поскольку Grid-система представляет собой набор взаимодействующих между собой компонентов (вычислительных узлов и хранилищ данных), которые могут функционировать параллельно и в работе которых должна быть обеспечена синхронизация, наиболее подходящим математическим аппаратом для моделирования динамики таких систем являются сети Петри [3].

Модель работы узла в виде сети Петри

Модель работы вычислительного узла с 4 процессорами, обрабатывающего задания пользователя под управлением маршрутизатора в соответствии с общей схемой рис. 1 показана на рис. 2. Эта модель представляет собой одноцветную сеть Петри вида

$$C = (P, T, A), \tag{1}$$

где P и T множества позиций и переходов сети Петри соответственно, A - матрица инцидентности графа сети, связывающая позиции с переходами сети. Начальное состояние сети описывается разметкой

M_0=(proc, want, query, block, free, busy, wait, work)=(4, 0, 0, 1, 1, 0, 0, 0).

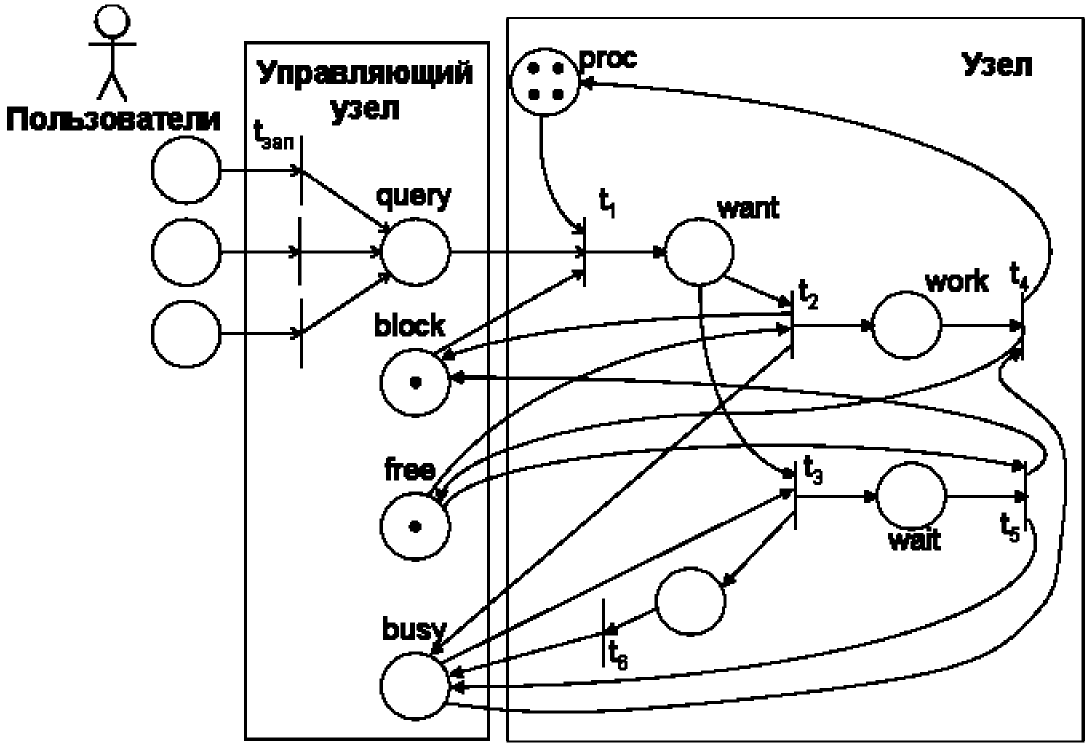

Рис. 2. Модель вычислительного узла, работающего под управлением планировщика над общей памятью с глубиной очереди 1

Позиции и переходы в данной сети Петри (СП) имеют следующую семантику.

proc — свободные в данный момент времени процессоры;

query — запрос на выполнение задания от пользователя;

want — процессор активен и желает начать работу;

work — процессор получил доступ к общей памяти и выполняет вычисления;

wait — активный процессор ожидает доступа к общей памяти;

block — блокируется доступ к общей памяти и очереди ожидания, если они заняты другими процессорами;

free — управляющий узел сигнализирует о снятии блокировки общей памяти;

busy — управляющий узел сигнализирует о том, что общая память заблокирована для доступа.

При начальной разметке M_0= (4, 0, 0, 1, 1, 0, 0, 0) система находится в состоянии покоя (который иногда называют состоянием «сна»). Из семантики позиций следует и смысл переходов:

$t_{зап}$ — наличие запроса на обработку данных;

t_1 — один из свободных процессоров хочет работать;

t_2 — процессор получает доступ к общей памяти;

t_3 — активный процессор переходит в состояние ожидания;

t_4 — процессор завершил работу с общей памятью и она разблокирована;

t_5 — активный процессор из состояния ожидания переходит в состояние работы с общей памятью, при этом одновременно разблокируется переход активного процессора в состояние ожидания.

При появлении запроса пользователя на выполнение задания, срабатывает сначала один из переходов $t_{зап}$, а затем переход t_1. В результате этого один из свободных процессоров переходит в состояние активности (готовности к работе), т.е. переходит в позицию `want`. Если одновременно поступает 2 запроса, то выполнение одного из них будет задержано отсутствием маркера в позиции `block`. Дальнейшая работа модели очевидна.

Исследование свойств модели

Сначала исследуем структурные свойства построенной на рис. 2 сети Петри, а именно проверим выполнение свойства ограниченности сети Петри, удостоверимся в отсутствии мертвых (лишних) переходов, и недостижимых позиций.

Для этого воспользуемся уравнением состояний

$$Ax=0, \tag{2}$$

где A — целочисленная n×m матрица инцидентности сети Петри, *n* и m — мощности множеств *P* и *T* соответственно, а x — m-мерный вектор Париха [**4**]. С помощью уравнения состояния определим S- и T-инварианты сети Петри. Это и даст возможность выявить мертвые переходы, недостижимые позиции и проверить ограниченность сети.

S-инвариант — это линейное отношение на маркировке подмножества позиций, выражающееся в том, что сумма различных меток в позициях положительна. Если в S-инвариант входят все позиции сети Петри, то в построенной модели все позиции являются достижимыми. T-инвариант соответствует последовательности срабатывания переходов, переводящей сеть из маркировки M в ту же самую маркировку M [**5**]. Если T-инвариант включает все переходы сети, то она *жива*.

Размерность матрицы инцидентности A из (10) для данной сети Петри составляет 10×7 (10 уравнений, 7 неизвестных), а вектора Париха x — 7. В соответствии с табл. 1 матрица инцидентности для данной сети Петри будет иметь вид

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}.$$

Система диофантовых уравнений (2) имеет 2 решения:

$x_1 = \{1, 1, 1, 0, 1, 0, 0\}^T$

$x_2 = \{1, 1, 0, 1, 1, 1, 1\}^T$

Эти решения задают Т-инварианты системы. Поскольку в перечне решений нет ни одного столбца, содержащего только нулевые элементы, то все переходы в сети Петри являются живыми при данной начальной разметке, т.е. каждый переход в сети срабатывает хотя бы один раз.

Сгенерируем S-инварианты данной сети Петри. Для этого найдем решения системы уравнений

$$A^{T}y=0, \tag{3}$$

где A — целочисленная n×m матрица инцидентности сети Петри, т — символ транспонирования матрицы, y — n-мерный вектор.

Система диофантовых уравнений (3), содержащая 7 уравнений и 10 неизвестных имеет следующие решения.

$y_1 = \{0, 0, 0, 0, 0, 1, 1, 0, 0, 0\}^T$

$y_2 = \{0, 0, 1, 1, 1, 0, 0, 0, 0, 0\}^T$

$y_3 = \{0, 1, 0, 1, 1, 1, 0, 0, 0, 0\}^T$

$y_4 = \{1, 0, 0, 1, 1, 1, 0, 0, 1, 0\}^T$

$y_5 = \{0, 0, 0, 0, 0, 0, 1, 1, 0, 1\}^T$

$y_6 = \{0, 1, 0, 1, 1, 0, 0, 1, 0, 1\}^T$

$y_7 = \{1, 0, 0, 1, 1, 1, 0, 0, 1, 0\}^T$

Решения системы уравнений (3) определяют S-инварианты сети Петри. Из перечня решений следует, что сеть Петри является *ограниченной*, поскольку все позиции в ней соответствуют положительным инвариантам. Это означает, что ни в одной позиции сети не может скапливаться бесконечное число фишек.

Для получения множества S- и T-инвариантов сети Петри используется TSS-алгоритм [**6**, **7**], который позволяет построить минимальную порождающую систему решений однородной системы линейных диофантовых уравнений над множеством натуральных чисел N.

Исследование выполнимости свойств взаимного исключения и равноправия

Для исследования выполнимости свойств взаимного исключения (`mutex`) и равноправия (`fairness`) необходимо построить *транзиционную систему* [**8**] или *граф достижимых разметок* сети Петри .

Для начальной разметки, определяемой соотношением

M_0=(proc, want, query, block, free, busy, wait, work)=(4, 0, 3, 1, 1, 0, 0, 0),

Транзиционная система сети Петри, показанной на рис 2, имеет вид, представленный на рис. 3.

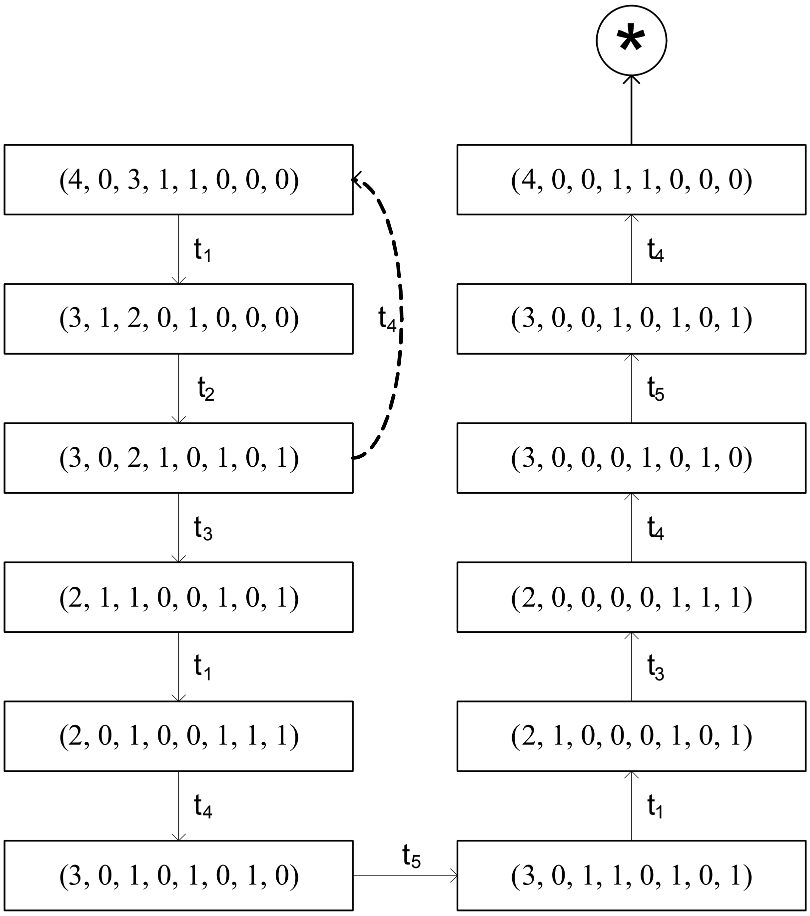

Рис. 3. Транзиционная система сети Петри, моделирующей работу узла Grid-системы

Простой анализ полученной транзиционной системы показывает, что свойство взаимного исключения (`mutex`) выполняется, поскольку в позиции `work` во время функционирования сети Петри появляется не более одной фишки. Это же справедливо и для позиции `wait`. Число фишек в этих позициях в транзиционной системе на рис. 3 задается двумя последними элементами векторов разметки.

Свойство равноправия (`fairness`) тоже, очевидно, выполняется, поскольку при появлении активного процессора (1 в позиции `want`) в разметке μ из этой разметки достижима разметка μ\`, при которой в позиции `want` находится 0 фишек. А это означает, что процессор рано или поздно получит доступ к соответствующему ресурсу (памяти, данным и т.п.).

Литература

1. Krauter K., Buyya R., Maheswaran M. A Taxonomy and Survey of GRID Resource Management Systems and Distributed Computing // Software-Practice and Experience, John Wiley & Sons, Ltd. — 2001. — P. 1–10.
2. Куссуль Н.Н., Шелестов А.Ю., Лобунец А.Г. Применение методов операционного анализа для оценки производительности GRID-систем // Кибернетика и вычислительная техника. – 2004. – Выпуск 144. – С. 3-20.
3. Дж. Питерсон Теория сетей Петри и моделирование систем. — М.: Мир, 1984. — 264 с.
4. Parikh R. On Context-Free Languages// J. of the ACM, 13, # 4, 1966. — PP. 570-581.
5. Дубинин В.Н., Зинкин С.А. Языки логического программирования в проектировании вычислительных систем и сетей: Учеб. пособие. — Пенза: Изд-во Пенз. гос. техн. ун-та, 1997. — 100 с.
6. Кривой С.Л. Критерий совместности систем линейных диофантовых уравнений над множеством натуральных чисел// Доповіді НАНУ. — 1999. — № 5. — С. 107-112.
7. Krivoi S. A criteria of Compatibility Systems of Linear Diophantine Constraints // Lecture Notes in Comp. Science. — 2002. — № 2328. — С. 264-271.
8. Летичевський О.А. Сучасні проблеми кібернетики. Нормативний курс. Навчальна електронна бібліотека факультету кібернетики Київського національного університету ім. Тараса Шевченка// http://www.unicyb.kiev.ua/Library/

Получено 21.05.06