УДК 004.75, 004.272

В.В. Лукашенко, В.А. Романчук

РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ РЕСТРУКТУРИЗУЕМОГО ПОД КЛАССЫ ЗАДАЧ, ВИРТУАЛИЗУЕМОГО КЛАСТЕРА ВЫЧИСЛИТЕЛЬНОЙ GRID-СИСТЕМЫ НА БАЗЕ НЕЙРОПРОЦЕССОРОВ

В статье предлагается математическая модель кластера, реструктуризуемого по заданным структурам под определенные классы задач, рабочие узлы (машины) которого работают на базе нейропроцессоров. Детально представлена модель диспетчера кластера и управляющего модуля планирования задач на машины кластера.

распределенные вычисления, неропроцессоры, кластеризация.

В настоящее время наибольшее распространение нейроподобные сети (НПС) и соответственно нейрокомпьютерные системы (НКС) получили для решения трудно формализуемых и в основном неформализуемых задач.

В перечисленных случаях применение традиционных универсальных компьютеров неэффективно. Отличительными чертами таких задач являются:

1. Нестандартные характеристики обрабатываемой информации.
2. Информация запоминается и содержится не в отдельных ячейках памяти, а в связях между нейронами и в их величине, при этом состояние каждого отдельного нейрона определяется состоянием многих других, связанных с ним нейронов.
3. Потеря одной или нескольких связей между нейронами не оказывает существенного влияния на результат работы системы в целом. Кроме того, высокая естественная помехоустойчивость и функциональная надежность нейроподобных сетей не зависят от зашумляющих потоков и отказов отдельных процессорных элементов. Тем самым в нейронной системе обеспечиваются высокая

© Лукашенко В.В., Романчук В.А., 2014

ии.

4. Возможность обучения, дообучения и переобучения нейроподобной сети при изменении внешних факторов. Вышеперечисленные операции позволяют сделать нейроподобные сети универсальными, позволяющими легко перестроить их на новые виды решаемых задач.

Однако производительность современных нейропроцессорных модулей не позволяет добиться оптимальной скорости решения подобных задач. В связи с этим совершенно логично использовать опыт распределенных вычислений для наращивания суммарных мощностей объединенной виртуализуемой системы нейромикроконтроллеров.

Распределенный виртуализуемый реструктуризуемый (под задачу) вычислительный кластер на базе нейропроцессоров может быть описан в виде полного взвешенного графа следующим образом:

$$G = (CN, \Delta, Pi, Ki, Ei, VO, F(CN), q, l, b, c, m, d, \delta), \quad (1)$$

где: $CN = \{CN1, \ldots CNi\}$, $I = [1,\infty]$ – множество вычислительных машин (узлов, вершин) на базе нейропроцессоров, входящих в состав кластера; $F(CN) : CN \rightarrow 2VO \setminus \{\varnothing\}$ – функция принадлежности каждого узла к одной или нескольким виртуальным организациям, в рамках которых осуществляется разделение вычислений.

Отображение $q:CN \cup \{\Delta\} * CN \cup \{\Delta\} \rightarrow Z + \cup \{0\}$ определяет среднюю пропускную способность через сеть Интернет между каждой парой вычислительных узлов, а также между узлами и диспетчером Δ и измеряется в байтах в секунду. Функция $l:CN \cup \{\Delta\} * CN \cup \{\Delta\} \rightarrow Z + \cup : \{0\}$ задает среднюю задержку в секундах. Так как граф (1) неориентированный, то выполняется следующее свойство:

$$\forall u, v \in CN \cup \{\Delta\} \; q(u, v) = q(v, u) \& l(u, v) = l(v, u) \quad (1.1)$$

Предлагаемая модель работает по принципу коммутации пакетов при передаче данных между вычислительными узлами.

Δ – выделенная вершина кластера (диспетчер кластера), управляющая работой кластера. Формализуем вершину Δ:

$$\Delta = \{S, H, Z, SC\}, \quad (2),$$

где:

– функции $S: CN \rightarrow AP$ и $H: CN \rightarrow 2\theta$ определяют соответственно аппаратную архитектуру всех узлов вычислительного кластера и набор предустановленного программного обеспечения; в предлагаемой модели все рабочие единицы (вершины) кластера работают на базе нейропроцессоров серии NM640X, подключенных через стандартный интерфейс (PCI, PCI-E), и имеют предустановленную операционную систему семейства *NIX;

– Z – управляющий модуль (группа алгоритмов) планирования задач на определенные машины кластера, при этом Z можно выразить в виде множества:

$$Z = \{Ad, Az, Asc[i], Af(sc[i]), Ar)\}, \quad (3)$$

где:

Ad (3.1) – алгоритм определения сложности задачи, выполнение которой осуществляется в вычислительном кластере. Алогоритм разбивает задачу на конкретные операции и по их количеству относит ее к одному из возможных классов нейропроцессорных структур обработки информации.

Az (3.2) – алгоритм разбиения задач на подзадачи. Планируется использование сочетания алгоритмов выбора задачи из очереди Most Processors First Served Scan и Backfill со следующими методами назначения, учитывающими топологию вычислительной системы: Paging, Multiple Buddy System, Adaptive

NoneeContiguous Allocation, Greedy AvailableBusy List, Minimizing Contention, Minimizing Contention Incremental.

Asc[i] (3.3) – алгоритм выбора необходимой нейропроцессорной структуры из множества возможных структур $SC = \{SC_1, SC_2, ..., SC_i ...\}$ из (3), где SC_i –конвейер, вектор или их сочетание в виде комбинаций конвейерно-векторной или векторно-конвейерной структур и др.

Af(sc[i]) (3.4) – алгоритм формирования выбранной нейропроцессорной структуры из пула виртуализованных ресурсов кластера (1).

Ar (3.5) – алгоритм распределения сформированных в (3.2) подзадач по узлам структуры, полученной в результате работы (3.3) и (3.4).

Pi = {pi1, pi2,…, pig[i] } – множество вычислительных узлов, Ki = {ki1, ki2, …, kiz[i]} – множество коммутаторов, Ei – множество направленных сетевых связей между ними. В данном случае в качестве вершин рассматривается Pi ∪ Ki, представляющее сосбой совокупность узлов и коммутаторов кластера (1).

Пусть P = $\bigcup_{i=1}^{r} P_i$, K = $\bigcup_{i=1}^{r} K_i$, E = $\bigcup_{i=1}^{r} E_i$ – соответственно объединения вычислительных узлов, коммутаторов и направленных сетевых связей кластера, тогда в формуле (1) b: E → Z+ ∪ {0} – отображение, характеризующее пропускную способность каждой сетевой связи в байтах в секунду.

Функции c, m, d: P → Z_+ определяют для каждого вычислительного узла pij количество вычислительных ядер c(pij) = Cij, объемы оперативной m(pij) = = Mij и дисковой памяти d(pij) = Dij в мегабайтах.

Отображение δ: P → R_+ для узла pij задает относительную производительность δ(pij) = δij каждого его вычислительного ядра, которая определяет, во сколько раз ядра данного узла работают быстрее вычислительных ядер самого непроизводительного узла кластера. В кластере G все вычислительные узлы множества Pi связаны с выделенным управляющим узлом Δ с помощью отдельной управляющей сети, которая используется для передачи сигналов управления, доставки к ним файлов программ и входных данных, а также для возврата выходных файлов с результатами. Узел Δ в вычислениях не участвует.

Статические параметры Cij, Mij, Dij и δ описывают конфигурацию вычислительного узла pij, а динамические параметры характеризуют его состояние в произвольный момент времени t∈[0; +∞); uij(t) – загруженность вычислительных ядер; mij(t) и dij(t) – объемы доступной оперативной и дисковой памяти. Имеют место следующие неравенства:

$$0 \le uij(t) \le 1,\ 0\ \ mij(t) \le Mij,\ 0 \le dij(t) \le Dij. \qquad (4)$$

Структура типичного вычислительного узла pij кластера G изображена на рисунке 1. Вычислительные ядра могут относиться как к отдельным процессорам, так и к нескольким многоядерным процессорам. С целью упрощения модели они рассматриваются как единое множество вычислительных ядер узла pij:

$$Xij = \{\chi ij,1, \chi ij,2,\ldots, \chi ij,C[ij]\} \qquad (5)$$

Таким образом, X = $\bigcup_{\substack{i=1,r \\ j=1,g_i}} X_{ij}$ – множество вычислительных ядер всех узлов всех кластеров.

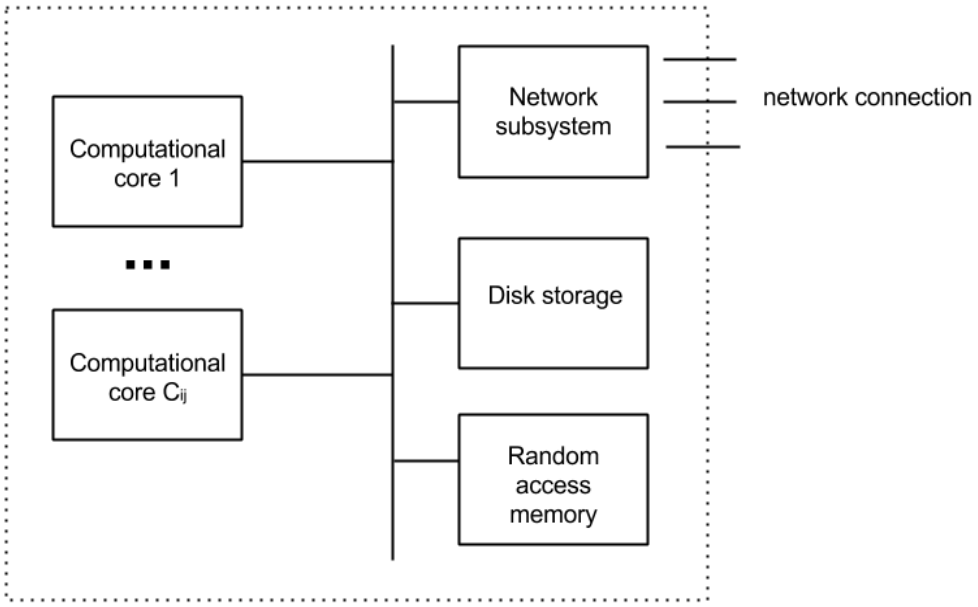

Рис. 1. Структура узла pij, кластера G

Процессы, выполняющиеся на вычислительных ядрах, могут непосредственно считывать и записывать информацию в оперативную и дисковую память данного узла, взаимодействуя с ними через системную шину или коммутатор. При необходимости передачи сообщения другому вычислительному узлу в рамках одного кластера задействуется сетевая подсистема, осуществляющая разбиение сообщения на пакеты и передачу их по высокопроизводительной коммуникационной сети. Сетевая подсистема каждого узла также содержит коммутатор, выполняющий функции маршрутизации собственных и транзитных пакетов.

Обозначим в качестве id(χ, t) номер задачи, которая выполняется на ядре $\chi \in X$ в момент времени t. Если же ядро χ в момент времени t было свободно, то id(χ,t) = 0. Пусть state(χ,t) = {free, communication, computation} – состояние вычислительного ядра $\chi \in X$ в момент времени t (free – свободно, communication – исполняет процесс на стадии коммуникации, computation – исполняет процесс на стадии вычислений).

Значения динамических параметров узла pij в момент времени t могут быть вычислены по следующим формулам:

$$u_{ij}(t) = \frac{\left|\{\chi \in X_{ij} : id(\chi,t) > 0 \,\&\, state(\chi,t) = computation\}\right|}{C_{ij}},$$

$$m_{ij}(t) = M_{ij} - \sum_{\chi \in X_{ij}, e = id(\chi,t) > 0} mr_e \;,$$

$$d_{ij}(t) = D_{ij} - \sum_{\chi \in X_{ij}, e = id(\chi,t) > 0} dr_e \;. \qquad (6)$$

где mre и dre – соответственно объемы оперативной и дисковой памяти, необходимой каждому процессу параллельной подзадачи.

Таким образом, предложенная в работе математическая модель кластера, в инструментальную основу вычислительных мощностей узлов которого положены нейропроцессорные модули, имеет явное преимущество в обработке задач как перед одиночными машинами на базе нейропроцессоров (за счет использования технологий распределенных вычислений), так и перед стандартными вычислительными кластерами для трудно формализуемых и в основном неформализуемых классов задач.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Foster, I. The anatomy of the GRID [Text] // I. Foster, C. Kesselman, St. Tuecke // International Journal of High Performance Computing Applications. – 2001. – No 3/15. – P. 200–222.

2. Lublin, U. The workload on parallel supercomputers: modeling the characteristics of rigid job [Text] / U. Lublin, G. Feitelson // Journal of Parallel and Distributed Computing archive. – 2003. – Vol. 63. – Iss. 11. – P. 542–546.

3. Grigorenko, D. Improving recoverability cluster neuroprocessor data processing systems [Text] / D. Grigorenko, V. Ruchkin // Digital signal processing. – 2012. – No 3. – P. 67–72.

4. Polezhaev, P. Experimental study of scheduling algorithms for computational grid problems-system [Text] // Control Systems and Information Technology. – 2011. – No3.2/45. – P. 266–270.

V.V. Lukashenko, V.A. Romanchuk

MATHEMATICAL MODELING OF VIRTUAL CLUSTERS OF GRID-ENABLED SYSTEM PROGRAMS ON THE BASIS OF NEUROPROCESSORS

The paper provides a survey of mathematical modeling of virtual clusters reconfigured for some specific tasks. Production nodes of such clusters function on the basis of neuroprocessors. The paper also provides detailed models of executive programs and control modules for certain nodes.

Distributed computing, neuroprocesses, clusterization.