

УДК: 004.272

ОПТИМИЗАЦИЯ СТРУКТУРЫ ПАРАЛЛЕЛЬНОЙ ПРОГРАММЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ

OPTIMIZING THE STRUCTURES OF PARALLEL PROGRAM OF COMPUTER SYSTEMS FOR SPECIAL PURPOSES

*Канд. техн. наук О.Е. Нестеренко, канд. техн. наук А.М. Гончаров,
канд. техн. наук С.А. Платонов*

PhD O.E. Nesterenko, PhD A.M. Goncharov, PhD S.A. Platonov

ВКА им. А.Ф. Можайского

В статье представлен подход к оптимизации структур параллельных программ вычислительных систем, построенных на архитектуре NUMA, в интересах развёртывания единого информационного пространства Вооруженных Сил Российской Федерации на основе модифицированного симплекс-метода, учитывающий особенности архитектуры построения вычислительных комплексов. Использование данного подхода позволяет решить ряд проблемных вопросов при проектировании программного обеспечения для вычислительных систем, осуществляющих параллельную обработку значительных объёмов информации. Преимуществом подхода является возможность получения оптимального решения учитывающего порядок выполнения операций и возможность их параллельного выполнения.

Ключевые слова: вычислительная система, вычислительный комплекс, оптимизация параллельных вычислений, планирование обработки информации, параллельные программы.

The article presents a methodological approach to optimizing the structures of parallel programs of computing systems built on the NUMA architecture, in the interests of deploying a unified information space of the Armed Forces of the Russian Federation, based on a modified simplex method that takes into account the features of the architecture of construction. The use of this approach makes it possible to solve a number of problematic issues in the design of software for computing systems that perform parallel processing of significant amounts of information. The advantage of this approach is the ability to obtain an optimal solution that takes into account the order of execution of operations and the possibility of their parallel execution.

Keywords: computer system, computer complex, optimization of parallel calculations, planning of information processing, parallel programs.

Введение

Появлению вычислительной техники способствовала необходимость выполнения сложных вычислений в рамках решения прикладных задач в приемлемые сроки. Прежде всего, первые образцы электронных вычислительных ма-

шин (ЭВМ) использовались в военных целях для решения задач криптографии и обработки радиолокационной информации.

Вторая половина XX века ознаменовалась ростом военной и политической напряжённости и, как следствие, ростом требований к вычислительным системам, осуществляющим обработку

различной информации. Исследования того времени, проводимые в рамках создания отечественной системы мониторинга космических и баллистических объектов, выявили недостаточную производительность классических вычислительных машин, построенных на Принстонской архитектуре, для решения многих прикладных задач и задач обработки радиолокационной информации в частности. В это время в нашей стране стали появляться первые многопроцессорные вычислительные системы, предназначенные для параллельной обработки значительных объёмов информации.

В 1995 году во исполнение Указа Президента Российской Федерации «О совершенствовании информационно-телекоммуникационного обеспечения органов государственной власти и порядке их взаимодействия при реализации государственной политики в сфере информатизации» была принята Концепция формирования и развития единого информационного пространства (ЕИП) России и соответствующих государственных информационных ресурсов, в рамках которой предполагалось создание единого информационного пространства Вооруженных Сил Российской Федерации (ЕИП ВС РФ). Основной целью создания ЕИП являлось повышение качества управления и принятия решений органами государственной власти Российской Федерации. Однако в результате недостаточного финансирования, отсутствия телекоммуникационной системы, обладающей необходимой пропускной способностью, а также недостаточной производительности вычислительных средств, осуществляющих обработку информации в узлах этой системы, развёртывание ЕИП значительно замедлилось.

Исторически сложилось три основных направления повышения производительности вычислительных систем, а именно: совершен-

ствование элементной базы, эволюция архитектурных решений и организация параллельной обработки информации (рис. 1).

Причём, если первые два направления лежат в области достижений микроэлектроники, то организация параллельной обработки информации является наиболее перспективным с точки зрения теоретической науки. Очевидно, что увеличение количества ЭВМ, работающих над решением одной общей вычислительной задачи, сократит общую длительность её выполнения. Однако такой подход зачастую связан с наличием сложно-разрешимых проблем. Например, при составлении параллельных программ необходимо учитывать простоту системы при вводе-выводе информации и оценивать время, необходимое на передачу информации между узлами вычислительной системы через коммуникационную среду в процессе выполнения параллельной программы. Увеличение количества узлов вычислительной системы ведёт к увеличению общей производительности всей вычислительной системы, но, кроме этого, увеличивается разрыв между реальной и пиковой производительностью. Ко всему прочему, на практике встречаются алгоритмы вычислительных задач, которые не могут быть распараллелены ввиду наличия абсолютной связности по входным-выходным данным операций этих алгоритмов. Наличие этих и других особенностей накладывает принципиальные ограничения на процесс разработки программного обеспечения (ПО) для высокопроизводительных вычислительных систем, осуществляющих обработку информации в узлах ЕИП ВС РФ.

Так, зачастую, программисты при разработке параллельных программ не используют в своей профессиональной деятельности какой-либо методический аппарат, а надеются на личный опыт, догадки и эвристические методы. К тому же использование в современных отечественных

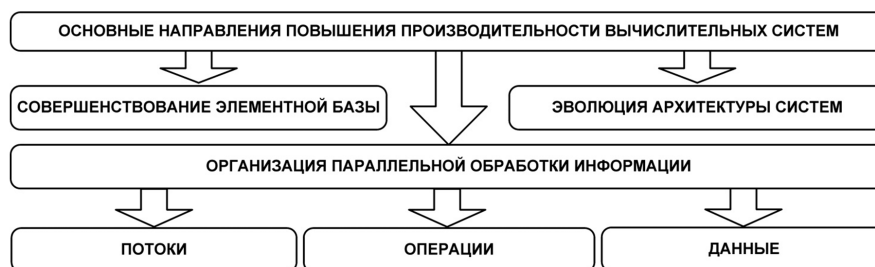


Рис. 1. Направления повышения производительности вычислительных систем

вычислительных системах архитектуры NUMA (рис. 2) и ccNUMA влечёт за собой необходимость учёта длительности передачи информации между локальными и глобальными узлами системы. Таким образом, эти два фактора усложняют разработку ПО, увеличивают трудность отладки и не гарантируют высокого качества программных продуктов.

Следовательно, можно сделать вывод, что задача оптимизации структур параллельных программ вычислительных средств автоматизированных систем специального назначения, с учётом особенностей их архитектуры, требует использования апробированного математического аппарата и является актуальной.

Постановка задачи

Параллельная программа — это последовательность операций совместимых во времени, реализующих некоторый алгоритм вычислительной задачи на многопроцессорной параллельной вычислительной системе. Как следует из этого определения параллельная программа состоит из отдельных операций, каждая из которых может быть охарактеризована, прежде всего, длительностью её выполнения, множеством входных и выходных переменных. Наиболее популярным способом формализации параллельной программы и её структуры в частности яв-

ляется использование графоаналитических моделей [1, 5]. При этом вершинами графа будут являться выполняемые операции, а наличие дуги будет определяться в случае, если выходные переменные одной операции являются входными переменными для другой.

Таким образом, структуру параллельной программы можно представить в виде графа:

$$G(V, U),$$

где $V = \{v_0, v_1, \dots, v_h\}$ — множество вершин (операций программы);

$U = \{u_0, u_1, \dots, u_n\}$ — множество дуг (связей между операциями).

А соответствующая этому графу матрица инцидентности примет вид:

$$M = \left\| m_{ij} \right\|_{h \times n}, \text{ при } i = \overline{0, h} \text{ и } j = \overline{0, n}.$$

Кроме того, параллельная программа характеризуется множеством $T = \{t_0, t_1, \dots, t_h\}$ длительностей её операций.

Несомненно, что основным показателем, характеризующим качество выполнения параллельной программы, является длительность её выполнения, т.е. разность между началом выполнения её первой операции и окончанием последней. Таким образом, при разработке параллельных программ решается задача оптимизация её

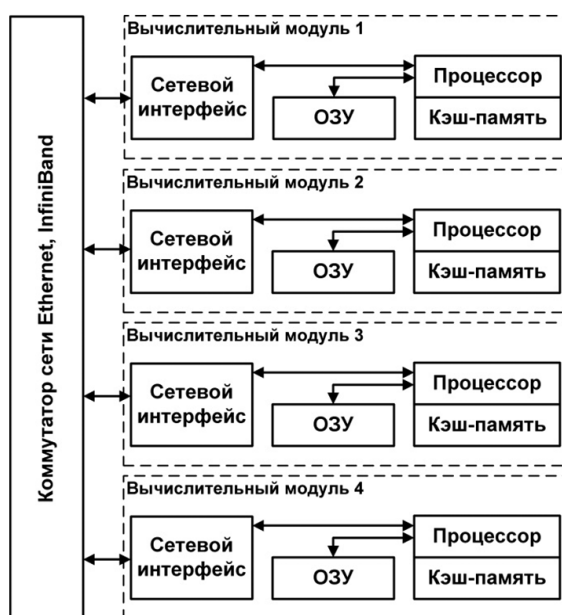


Рис. 2. Структура вычислительных систем NUMA

структуры, обеспечивающей минимальную длительность выполнения, т.е.

$$\vec{x}^* = \operatorname{argmin}_{x \in \Delta} (x_h + t_h - x_0),$$

где, компонентами вектора $\vec{x}^* = (x_0, x_1, \dots, x_h)$ являются моменты времени начала операций программы, обеспечивающих минимум длительности выполнения, а Δ — целочисленные неотрицательные x_i .

Однако, как отмечалось ранее, особенности вычислительной системы, на которой выполняется параллельная программа, накладывают ряд дополнительных ограничений, учитываемых при разработке ПО. Так, в NUMA-системах, построенных, как известно, на основе распределённой памяти, длительность обращения к переменной зависит от того, в какой области памяти она находится (локальной или глобальной, относительно вычислительного узла). Таким образом, при оптимизации ПО для таких систем определяющее значение имеет размещение переменных в распределённой памяти.

Кроме этого, наличие кэш памяти в современных микропроцессорах (при всех преимуществах её использования в «классических» системах) вызывает ряд трудностей при обеспечении когерентности данных хранящихся в различных её узлах. Однако использование, так называемого, справочника переменных позволяет минимизировать влияние этого фактора, но в процессе разработки необходимо учитывать временные затраты на обращение к нему.

Таким образом, для получения оценок длительности выполнения параллельной программы, необходимо учесть характеристики вычислительной системы, на которой будет выполняться программа.

Основные этапы процесса оптимизации структур параллельных программ вычислительных систем

В основу методического подхода для оптимизации структур параллельных программ вычислительных систем автоматизированных систем специального назначения положена соответствующая методика, которая включает основные этапы.

1. Ввод исходных данных:

а) матрица инцидентности \mathbf{M} , соответствующую исходному графу $G(V, U)$ вычислительной задачи;

б) характеристики, описывающие особенности вычислительной системы.

2. Расчёт оптимальной структуры параллельной программы модифицированным симплекс-методом без учёта особенностей вычислительной системы [2].

3. Получение оценок длительности записи и чтения переменных в распределённую память для каждой операции параллельной программы при её исполнении на конкретной вычислительной системе и оценивание общей длительности параллельной программы с оптимизированной структурой.

Расчёт оптимальной структуры параллельной программы модифицированным симплекс-методом без учёта особенностей вычислительной системы

В [2] предложено использование модифицированного симплекс-метода в целочисленной постановке общей задачи линейного программирования для сокращения вычислительной сложности получения оптимальной информационной структуры вычислительной задачи, реализуемой на средствах автоматизированных систем мониторинга космических и баллистических объектов. Тем не менее, при решении данной оптимизационной задачи была выявлена применимость этого алгоритма для других классов задач, в том числе при разработке ПО высокопроизводительных вычислительных систем, развёртываемых в интересах ЕИП ВС РФ.

Процесс оптимизации информационной структуры выполняется в 3 этапа.

1 этап. В исходный граф $G(V, U)$ и соответствующую ему матрицу инцидентности \mathbf{M} вводятся две дополнительные вершины нулевой длительности, с целью определения конкретных операций начала и окончания выполнения параллельной программы. При этом первая вершина соединяется ветвями с истоками исходного графа, а вторая — со стоками.

2 этап. Составляется система ограничений для решения общей задачи линейного программирования вида:

$$x_i + t_i \leq x_k, \text{ при } k > i,$$

где x_i и x_k — моменты времени начала выполнения двух последовательных операций, имеющих связь между собой.

При этом целевая функция соответствует задаче оптимизации по длительности выполнения параллельной программы:

$$L(\vec{x}) = x_h + t_h - x_0 \rightarrow \min .$$

3 этап. Производится вычисление моментов начала каждой операции. В основе данного алгоритма лежит использование табличного симплекс-метода с использованием улучшенных симплекс-преобразований при решении общей задачи линейного программирования, позволяющих сократить его вычислительную сложность [2].

Получение оценок длительности записи и чтения переменных в распределённую память

Для получения оценок длительности записи и чтения переменных в распределённую память для каждой операции параллельной программы при её исполнении на конкретной вычислительной системе и оценивания общей длительности параллельной программы с оптимизированной структурой предлагается использовать модель, предложенную в [3, 6], разработанную на основе модели коммутатора [4].

В основе модели лежит учёт особенностей процесса записи и чтения переменных (рис. 3) в зависимости от их относительного расположения в локальной или глобальной памяти. При этом для расчёта длительности обращений к памяти используется аналитическая форма представления процессов, отображенных на рис. 3, полученных на основе модели беспriorитетного коммутатора с очередью фиксированной длины с неоднородным входным потоком обращений к памяти.

Для описания динамики состояния коммутатора сети как системы массового обслуживания с неоднородным потоком заявок, необходимо составить систему дифференциальных уравнений Колмогорова для каждого потока поступающих заявок λ_j [3, 7].

На основе данной модели получены зависимости для оценки длительности информационного обмена при чтении значения переменной из распределённой памяти (рис. 3, б):

$$T_i^{read} = \alpha_1 \left[t_{req} + \frac{g(\lambda_g + 1)}{2c - \lambda_g g} + \frac{\sum_{j=0}^n [\lambda_j k_j (1 + \varphi_j^2)] + k_j}{2c - \sum_{j=0}^n \lambda_j k_j} \right] + p(\log_2(r + 1) - 1) + t_{fetch},$$

где g — количество вычислительных узлов вычислительной системы;

c — пропускная способность коммутатора;

t_{fetch} — средняя длительность выборки значений переменной из оперативного запоминающего устройства (ОЗУ);

t_{req} — средняя длительность формирования запроса на получение значений переменных из распределённой памяти;

p — средняя длительность операции сравнения значений переменных;

r — количество записей в справочнике переменных;

λ_j — средняя плотность поступления заявок определённого класса на запись и чтение переменных в распределённую память;

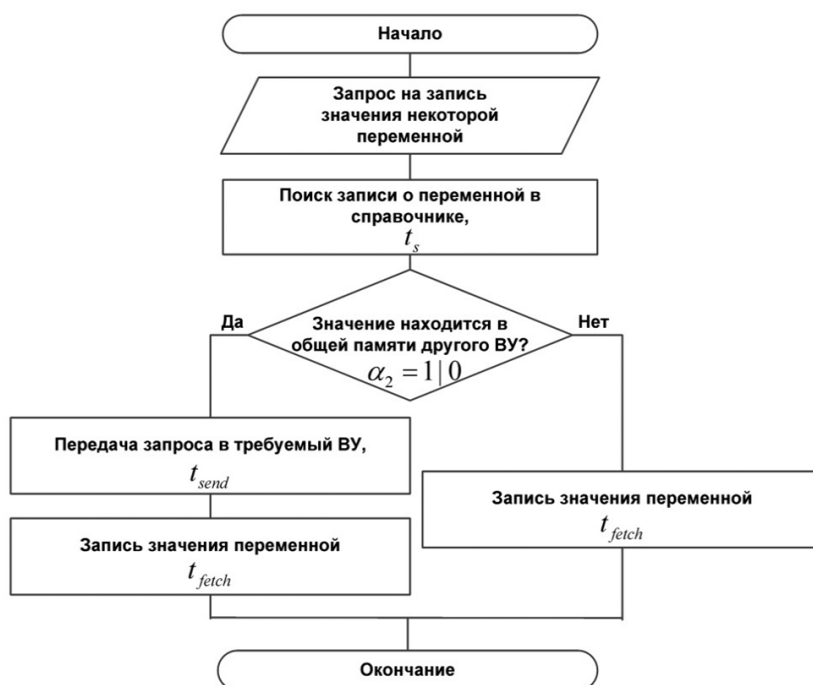
φ_j — коэффициент вариации длительности обслуживания каждого класса заявок.

Длительность записи значений (рис. 3, а) в память определяется выражением:

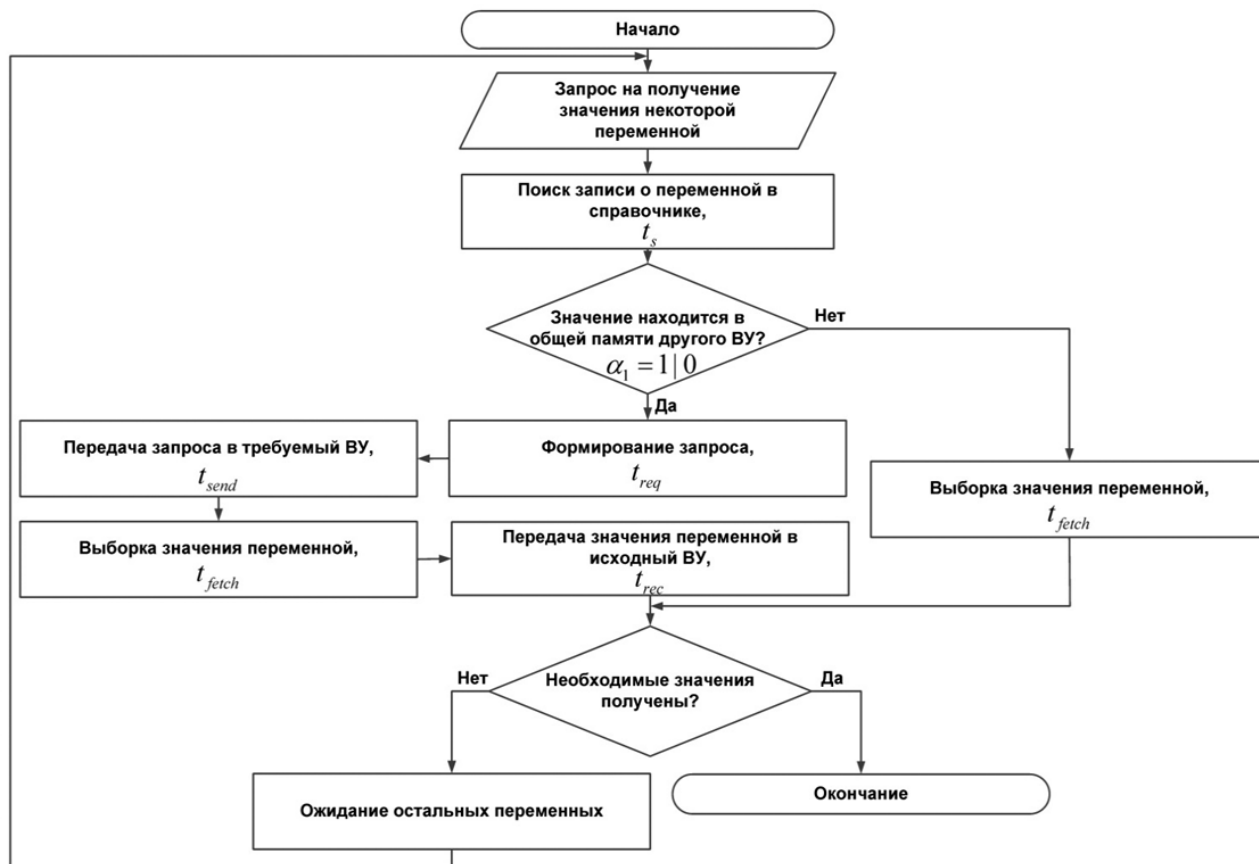
$$T_i^{write} = \alpha_2 \frac{\sum_{j=0}^n [\lambda_j k_j (1 + \varphi_j^2)] + k_j}{2c - \sum_{j=0}^n \lambda_j k_j} + p(\log_2(r + 1) - 1) + t_{fetch}.$$

При этом коэффициенты α_1, α_2 принимают единичное значение в случае, если осуществляется обращение к ОЗУ другого узла вычислительной системы.

Используя полученные оценки длительности информационного обмена, для каждой ветви графа $G(V, U)$ необходимо пересчитать общую длительность выполнения параллельной обра-



а



б

Рис. 3. Алгоритмы работы с распределённой памятью: а — запись значения переменной; б — чтение значения переменной

ботки информации с учётом времени необходимого для записи и чтения значений переменных из распределённой памяти, т.е. итоговый вектор моментов начала каждой операции примет вид:

$$\vec{x}^* = \left(\begin{array}{l} (x_0 + T_0^{write} + T_0^{read}), (x_1 + T_1^{write} + T_1^{read}), \\ \dots, (x_h + T_h^{write} + T_h^{read}) \end{array} \right).$$

Заключение

Вопросы организации параллельной обработки информации на многопроцессорных высокопроизводительных системах долгое время не привлекали интереса широкой научной общественности. Это было обусловлено незначительным количеством больших задач, требующих привлечения таких систем, и достаточной производительностью существующих вычислительных машин.

Современные вычислительные системы, используемые для построения автоматизированных систем специального назначения, в том числе развёртываемых в интересах функционирования ЕИП ВС РФ, имеют кардинальные отличия от классических фон-Неймановских систем. Особенности их архитектуры построения позволяют лишь частично использовать научно-методический аппарат, применяемый для разработки программного обеспечения, разработанного для последовательных вычислительных систем.

Разработанный методический подход позволяет разработчику специального программного обеспечения решить ряд задач для вычислительных систем, осуществляющих параллельную обработку значительных объёмов информации. При этом области применения данного методического аппарата не ограничиваются специальными вычислительными системами и возможно его использование в других областях науки и техники.

Литература

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. — СПб: БХВ-Петербург. 2002. 608 с.
2. Ледянкин И.А., Нестеренко О.Е., Платонов С.А. Модифицированный симплекс-метод для оптимизации информационных структур вычислительных задач, реализуемых на средствах автоматизированных систем мониторинга кос-

мических и баллистических объектов // Труды Военно-космической академии им. А.Ф. Можайского. 2020. № 672. С. 17–24.

3. Нестеренко О.Е. Модель процессов организации обработки оперативной информации распределёнными вычислительными комплексами специального назначения // Т-Comm: Телекоммуникации и транспорт. 2016. Т. 10. № 12. С. 48–54.

4. Буренин А.Н., Легков К.Е. Современные инфокоммуникационные системы и сети специального назначения. Основы построения управления. — М.: ООО «ИД Медиа Паблишер». 2015. 348 с.

5. Топорков В.В. Модели распределённых вычислений. — М.: ФИЗМАТЛИТ. 2004. 320 с.

6. Клейнрок Л. Теория массового обслуживания. — М.: Машиностроение. 1979. 432 с.

7. Алиев Т.И. Основы моделирования дискретных систем. — СПб: СПбГУ ИТМО. 2009. 363 с.

References

1. Voevodin V.V., Voevodin V.I. Parallel computing. — St. Petersburg: BHV-Petersburg. 2002. 608 p.
2. Ledyankin I.A., Nesterenko O.E., Platonov S.A. Modified simplex-method for optimizing information structures of computational tasks implemented on the means of automated systems for monitoring space and ballistic objects // Proceedings of the Military Space Academy named after A.F. Mozhaisky. 2020. № 672. P. 17–24.
3. Nesterenko O.E. Model of processes of organization of processing of operational information by distributed computing complexes of special purpose // T-Comm: Telecommunications and transport. 2016. Vol. 10. № 12. P. 48–54.
4. Burenin A.N., Legkov K.E. Modern infocommunication systems and networks of special purpose. Fundamentals of building management. — Moscow: LLC «ID Media Publisher». 2015. 348 p.
5. Toporkov V.V. Models of distributed computing. — Moscow: FIZMATLIT. 2004. 320 p.
6. Kleyrock L. Queueing Theory. — Moscow: Mashinostroenie. 1979. 432 p.
7. Aliev T.I. Basics of modeling discrete systems. — St. Petersburg: SPbSU ITMO. 2009. 363 p.