

УДК 681.324

И.А. Скатков, ст. преп., канд. техн. наук; Д.Ю. Воронин, аспирант

Севастопольский национальный технический университет

ул. Университетская 33, г. Севастополь, Украина, 99053

E-mail: kvt@sevgtu.sebastopol.ua

ОПТИМИЗАЦИЯ ОБРАБОТКИ РАСПРЕДЕЛЕННЫХ ЗАДАЧ ПРИ ИСПОЛЬЗОВАНИИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ ДИСПЕТЧЕРИЗАЦИОННЫХ РЕШЕНИЙ

Предлагается система поддержки принятия решений по организации диспетчеризации в распределенных вычислительных системах (РВС). Рассматривается структурный и функциональный синтез диспетчеризационных подсистем.

Введение. Распределенную систему принято представлять в виде совокупности автономных средств обработки данных, которые функционируют совместно, на основе информационных обменов при помощи среды коммутации. В качестве среды коммутации в большинстве случаев выступает компьютерная сеть [1].

Можно обозначить множество преимуществ распределенных систем: распараллеливание вычислений, относительно хорошая масштабируемость, совместная работа пользователей с общими ресурсами и множество других. Однако существует и ряд проблем, связанных с организацией диспетчеризации выполняемых заданий в условиях разнородности программного обеспечения, необходимости соблюдения стандартов безопасности и т.д. [2].

Причинами, обуславливающими функциональную распределенность, являются следующие.

1. Источники информации, необходимой для решения задач, разнесены в пространстве и во времени. Примером может служить банковская компания, имеющая множество филиалов с распределенными сервисами. База данных должна быть единой. Встает проблема синхронизации, обеспечения достоверности, целостности и безопасности. Необходим своевременный отклик системы – чтобы клиенту не приходилось ждать запроса о состоянии его счета слишком долго.

2. Высокая стоимость большинства вычислительных ресурсов требует организации мультитежимо. При использовании уникального оборудования, необходимо минимизировать количество его экземпляров и максимизировать коэффициент использования. Примером такого оборудования может быть мощный вычислительный сервер. Например, используется распределенная система обработки данных, которая занимается сбором информации с кассовых аппаратов (терминальных устройств), и организацией их удаленной обработки на мощном вычислительном сервере. При этом сервер и терминальные устройства могут быть значительно распределены (находится даже в разных странах). Связь между сервером и такой распределенной системой осуществляется при помощи широкополосного доступа к среде Internet. Такой подход позволяет значительно снизить материальные затраты без ущерба для производительности всей системы в целом.

3. Многие задачи являются многофазными (как во времени, так и в пространстве) по своей постановке. Например, задачи управления, прогнозирования, мониторинга и т.п.

Технология обработки задач в РВС весьма специфична и требует введения управляющих методов, способных учесть эту специфику. В частности, критерии качества работы РВС весьма противоречивы и вследствие этого при принятии управленческих решений часто возникают конфликты. Существуют определенные способы разрешения такого рода противоречий. Многие из них сводятся к смещению акцента между нагрузкой на каналы связи и степенью автономности узлов в РВС. Однако во многих случаях наиболее оправданным считается введение множества управляющих воздействий, организующее диспетчеризацию в РВС на должном уровне оптимальности.

Согласно [3], диспетчеризация представляет собой совокупность управляющих воздействий, оптимизирующих работу всей системы в целом, путем оперативного планирования и перераспределения доступа к ресурсам в ней. Под критериями оптимальности в данном случае принято понимать согласованность, ритмичность, экономичность и своевременность выполнения задач в системе.

При проектировании системы поддержки принятия решений по организации диспетчеризации в распределенных вычислительных системах следует учитывать, что задача диспетчеризации не может быть эффективно решена аналитически даже в случае полной информации. Это происходит в силу того, что ее решение приводит к многостадийной задаче составления расписаний и многоиндексной оптимизационной задаче булевого программирования (задаче о назначении). Следовательно, разработки алгоритмического характера в этом направлении являются крайне актуальными.

Целью данной статьи является рассмотрение ряда вопросов, связанных со структурным и функциональным синтезом диспетчеризационных подсистем в РВС.

Постановка задачи. Задача синтеза подсистемы организации диспетчеризации в РВС имеет структурную и функциональную составляющие. Используя математический аппарат теории сетевого

анализа [4] можно сформулировать задачу синтеза функциональной составляющей этой системы в следующем виде. На основе использования модели «узел – работа» («дуга – событие») можно построить граф $\xi (A, B)$, который характеризует функциональную взаимосвязь между задачами, решаемыми распределенной вычислительной системой. При этом, множество вершин $A: \{a_i\}, i = \overline{1, |A|}$ соответствует задачам, а множество дуг $B: \{b_{ij}\} \equiv (i, j)$ задает отношения предшествования между ними. Таким образом, число ограничений на предшествование в системе равно $|B|$.

Все задачи множества A расклассифицированы по типам. Элементы множества $M: \{M_1, \dots, M_k\}$ соответствуют задачам различных типов, где $|M|$ – число типов задач, а $|M_i|$ – число задач i -ого типа. Внутри множества M_i элементы ранжированы таким образом, что $\forall i, i: \{(i_0, i_1), (i_1, i_2) \dots (i_k, i_{k-1})\}$ справедливо $\tau_{01} > \tau_{12} > \dots \tau_{(k-1)k}$, где $\tau_{(k-1)k}$ – трудоемкость решения k -ой задачи после $(k-1)$ -ой. Для решения k -ой задачи в i -ом узле выделяется r_{ik} единиц ресурсов. При выделении ресурсов существуют определенные правила, обусловленные структурными особенностями РВС.

1) Необходимо обеспечить ограничение на минимальный объем выделяемых ресурсов ($\min r_{ij} = r_{ij}^{\min}$) узлу системы. Это происходит, например, в силу того, что платы оперативной памяти, при помощи которых наращивается дополнительная память в узле системы, имеют ограниченный минимальный объем, т.е. платы памяти с меньшим объемом сняты с промышленного производства.

2) Необходимо обеспечить ограничение на максимальный объем выделяемых ресурсов ($\max r_{ij} = r_{ij}^{\max}$) узлу системы. Это происходит, например, в силу того, что процессоров с большей тактовой частотой не производят в силу технологических ограничений производства.

3) В системе строго задано количество ресурсов, которые могут быть произвольным образом распределены между ее узлами $\sum^{MAX} = R^o$, где R^o – общее количество ресурсов (включая резервные), которые могут быть произвольным образом распределены между узлами системы, с целью более оптимального ее функционирования.

В [5] излагаются методы формализации синтеза структуры сложных систем. Проинтерпретируем задачу распределения ресурсов по узлам при помощи графов G_i и G_j .

Граф $G_i = (I, \Gamma_i)$ задает отношение связности между различными задачами, здесь I – множество вершин графа, отображающих различные задачи, а Γ_i – множество дуг, отображающих взаимосвязь между ними.

Граф $G_j = (J, \Gamma_j)$ задает особенности выделения ресурсов в распределенной системе, здесь J – множество вершин графа, отображающие различные узлы РВС, а Γ_j – множество дуг, отображающих процесс перераспределения ресурсов между ними.

Множество отображений графа G_i на G_j представляют собой различные варианты перераспределения ресурсов между узлами РВС в процессе решения задач. Итак, под решением задачи оптимального перераспределения ресурсов в РВС будем понимать такое отображение графа задач на граф перераспределения ресурсов, при котором достигается экстремум критерия качества отображения при выполнении заданных ограничений.

Критериями оптимальности функциональных решений могут служить:

- $\min \max t_{ijk}$, где t_{ij} – время решения i -ой задачи j -ым узлом при использовании k -ого ресурса;
- $\min \max r_{ijk}$;
- $\min \sum t_{ijk}$ и т.д.

Таким образом, синтез функциональной составляющей системы поддержки принятия диспетчеризационных решений сводится к нахождению такой системы правил f , которая используя информацию о трудоемкостях задач T и граф взаимообусловленностей задач $\xi (A, B)$, могла бы оптимальным образом распределить множество ресурсов R^o между узлами РВС. Иными словами, под разработкой функциональной составляющей следует понимать выбор оптимальной управляющей стратегии организации диспетчеризации.

Задача синтеза структурной составляющей, решаемая при помощи системы поддержки принятия диспетчеризационных решений, состоит в следующем. Пусть задана структура распределенной вычислительной системы (совокупность узлов, объединенных между собой при помощи среды коммутации). Каждый узел состоит из процессорного блока и буферного накопителя. Узлы в системе имеют различное количество процессоров в процессорном блоке, а также различный объем буферного накопителя. Для заданной структуры распределенной системы необходимо синтезировать структуру подсистемы диспетчеризации.

Входными данными для структурного синтеза являются описание структуры распределенной системы (в соответствии с классификацией, предложенной в [1]) и требования пользователя диспетчеризационной подсистеме. Выходными данными является полностью сформированная структура диспетчеризационной системы, с выбором оптимальной диспетчеризационной стратегии.

Декомпозиция задачи на основе классификации распределенных систем.

Алгоритмическое решение задачи синтеза подсистемы диспетчеризации потребовало выделения определенных классов структур РВС для того, чтобы для каждого такого класса предложить наилучшую стратегию диспетчеризации. Структурное многообразие РВС потребовало четкой и единой классификации, способной выделить основные структурные параметры РВС. Используя классификацию, предложенную в [1], можно однозначно выделить различные параметры РВС и рассмотреть специфику организации диспетчеризации с учетом ограничений, связанных с каждым из них. В качестве математической модели структуры РВС удобно воспользоваться графологическим преобразованием, предлагаемым в [1]. В [1] классификация основывается на следующих структурных признаках: топология среды коммутации, тип распределенных ресурсов, способ передачи информации в среде коммутации, масштабируемость, назначение распределенной системы и ряде других. Таким образом, код конкретной структурной реализации РВС составляется из последовательности групп идентификаторов распределенных кривой чертой. Каждая группа соответствует определенному классификационному признаку. В общем виде следует записывать таким образом: A/B/C/D/E/F/G/H/I/J. Например, для РВС с полносеточной топологией среды коммутации, с распределенной оперативной памятью, с использованием радиоканала для передачи данных в среде коммутации, с предусмотренной масштабируемостью, предназначенная для параллельной работы нескольких пользователей с общими ресурсами, обладающую прозрачностью месторасположения, многопользовательской, с использованием сетевой ОС классификационный код следует записать следующим образом: Fgr/ Mem/ R/ MrE/ MgE/ U/ Loc/ Mu/ nOs.

Конструктивным преимуществом предложенной классификации является то, что она описывает конкретные различия в структуре РВС, вместо использования абстрактных классов. Именно этот факт позволяет учесть факторы, определяющие структурные решения при синтезе подсистем диспетчеризации.

Решение поставленной задачи на основе реализации системы поддержки принятия решений по синтезу подсистемы диспетчеризации.

Процесс синтеза подсистемы диспетчеризации для распределенной вычислительной системы будем решать при помощи теории принятия решений. Можно разделить эту задачу на три основных этапа: формирование общей структуры подсистемы диспетчеризации, расчет количественных характеристик функциональных блоков, определение функциональной составляющей разрабатываемой подсистемы (рисунок 1).



Рисунок 1 – Этапы формирования диспетчеризационной подсистемы

На первом этапе (рисунок 2) формируется общий вид структуры подсистемы диспетчеризации. Определяются цели и задачи, возложенные на эту подсистему, а также структурные блоки и их состав,

например, буферный накопитель, информационные фильтры, запоминающие структуры. Общая структура подсистемы диспетчеризации формируется в процессе диалога с пользователем.

На втором этапе (рисунок 3) определяются параметры составных элементов подсистемы диспетчеризации, такие как быстродействие и число процессоров в процессорном блоке, ёмкость буферного накопителя и т.д. Второй этап выполняется с использованием аналитической модели подсистемы диспетчеризации. Критерии оценки и стоимостные функции задаются пользователем.

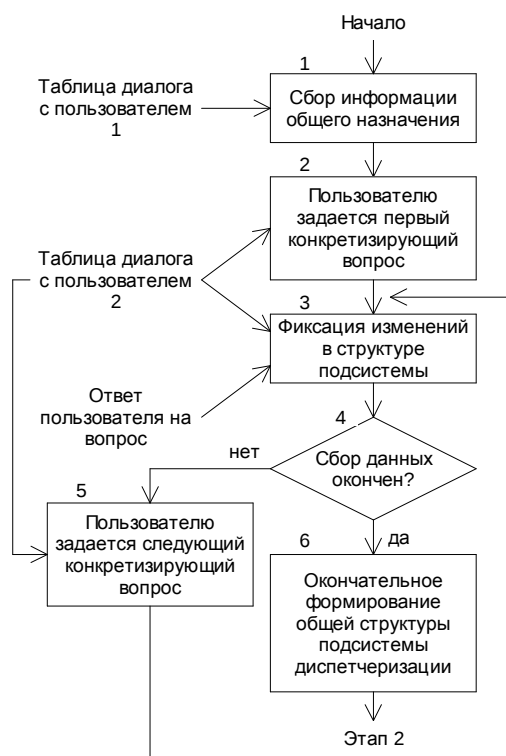


Рисунок 2 — Граф-схема этапа формирования общей структуры подсистемы диспетчеризации (этап 1)

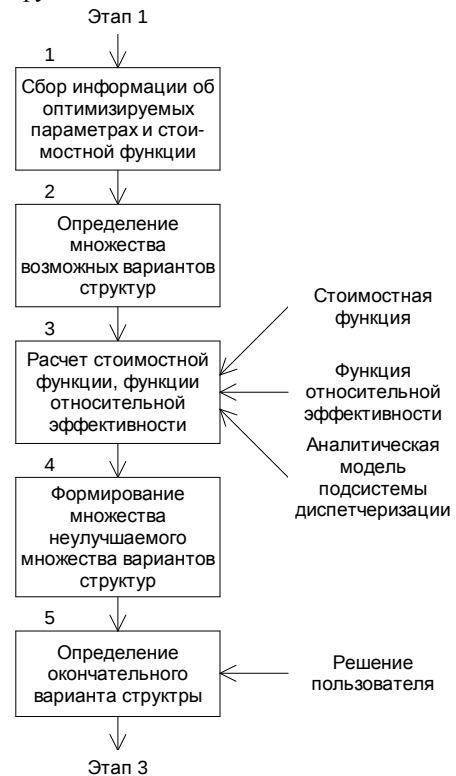


Рисунок 3 – Граф-схема этапа определения параметров структуры подсистемы диспетчеризации (этап 2)

На третьем этапе (рисунок 4) для разработанной структуры происходит определение оптимальной диспетчеризационной стратегии. Имеются шесть возможных стратегий: три базовых и три производных от них, носящих адаптивный характер. Производится имитационное моделирование работы подсистемы диспетчеризации с использованием каждой из стратегий, с последующей оценкой эффективности функционирования каждой из них, и соответственно выбора наиболее эффективной. Результатом проведения этого этапа является решение использовать конкретную стратегию в конкретном случае для заданной структуры диспетчеризационной подсистемы.

Базовые стратегии диспетчеризации состоят в следующем:

- 1) наиболее трудоемкие задачи необходимо решать в наиболее производительных узлах;
- 2) наименее производительные узлы следует занять наименее трудоемкими задачами;
- 3) приоритетным является использование узлов с наибольшей производительностью.

Кроме трех базовых дисциплин опроса система предполагает использовать комбинированные диспетчеризационные стратегии на базе адаптивных алгоритмов выбора вариантов и алгоритма обучения (самообучения). В разрабатываемой системе используются следующие адаптивные алгоритмы: беспроекционный алгоритм Варшавского-Воронцовой, проекционный алгоритм стохастической аппроксимации.

Адаптивный алгоритм Варшавского-Воронцовой относится к классу беспроекционных адаптивных алгоритмов. Данный алгоритм описан в [6] и предназначен для решения задачи адаптивного выбора вариантов среди N вариантов в случае с бинарными потерями $\xi_n \in \{0, 1\}$.

Алгоритм стохастической аппроксимации относится к классу проекционных адаптивных алгоритмов. Данный алгоритм также описан в [6].

Алгоритм обучения не имеет непосредственного отношения к адаптивным алгоритмам выбора вариантов, а принадлежит к теории вероятностных автоматов. Алгоритм исчерпывающе изложен в [7].

Заключение. Предложенная система поддержки принятия структурных и функциональных решений, связанных с задачей организации диспетчеризации в распределенных вычислительных системах, может активно быть использована при «пилотном» проектировании. Система имеет

интуитивно-понятный интерфейс и обеспечивает удобную среду для проведения исследований, направленных на повышение эффективности функционирования распределенной системы в целом. В рамках проведенных исследований, были изучены зависимости между задаваемыми пользователем требованиями и структурой подсистемы диспетчеризации.

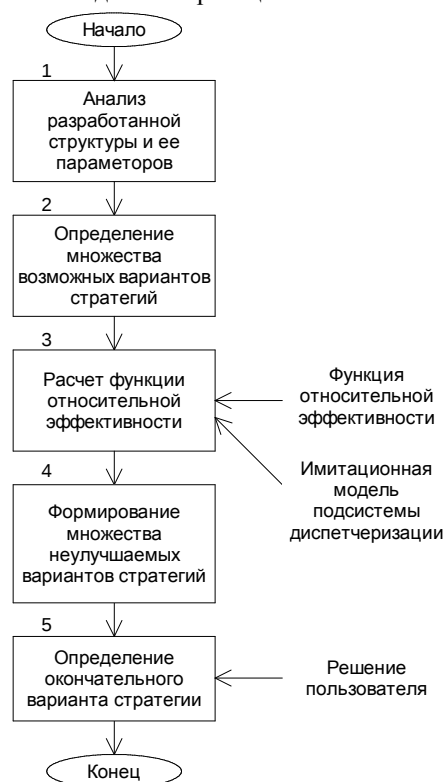


Рисунок 4 – Граф-схема этапа определения функциональной составляющей подсистемы диспетчеризации (этап 3)

По результатам проведенных исследований можно сделать ряд выводов.

1) существует значительная корреляция между выбранной структурой и оптимальной диспетчеризационной стратегией для нее;

2) использование адаптивного подхода для выбора диспетчеризационной стратегии позволяет значительно повысить значение функции эффективности;

3) высокий квалификационный уровень пользователя при выборе параметров подсистемы диспетчеризации позволяет значительно сократить вычислительные затраты.

Разработка эффективного адаптивного алгоритма, способного учитывать особенности функционирования распределенных систем – является первоочередным направлением дальнейших исследований.

Библиографический список

1. Скاتков А.В. Распределенные системы: структурный анализ, классификация, экстремальные задачи на графах / А.В. Скатков, Д.Ю. Воронин, Д.Н. Данильчук // Вестник Хмельницкого национального университета. — 2007. — № 3. — Т. 1 (93). — С. 77–82.
2. Таненбаум Э.С. Распределенные системы. Принципы и парадигмы / Э.С. Таненбаум, М. Ван Стеен. — СПб.: Изд-во «Питер», 2003. — 877 с.
3. Большая советская энциклопедия. 3-е изд. [Электронный ресурс]. – Электрон. текстовые данные (34578 bytes). — М.: Рубрикон, 2004. — Режим доступа: <http://bse.sci-lib.com/article029126.html> Monday, 14 April 2008 12:10:35.
4. Филлипс Д. Методы анализа сетей / Д. Филлипс, А. Гарсиа-Диас. — М.: Мир, 1984. — 496 с.
5. Цвиркун А.Д. Основы синтеза структуры сложных систем / А.Д. Цвиркун. — М.: Наука, 1982. — 200 с.
6. Назин А.В. Адаптивный выбор вариантов: рекуррентные алгоритмы / А.В. Назин, А.С. Позняк. — М.: Наука, 1986. — 288 с.
7. Поспелов Д.А. Вероятностные автоматы / Д.А. Поспелов. — М.: Энергия, 1970. — 88 с.

Поступила в редакцию 28.08.2008 г.