

## ВЫБОР СТРАТЕГИИ РАСПРЕДЕЛЕНИЯ ЛОГИЧЕСКИХ ФРАГМЕНТОВ БАЗ ДАННЫХ

к.т.н. Г.А. Кучук, к.т.н. Б.В. Остроумов, А.В. Пашнев  
(представил проф. А.В. Королев)

Предложен подход к анализу различных стратегий размещения логических фрагментов баз данных в среде распределенной вычислительной сети.

Рассмотрим распределенную сеть  $U$  неоднородных ЭВМ, состоящую из  $m$  локальных вычислительных сетей (ЛВС)  $u_i$  ( $U = \{u_i | i = \overline{1, m}\}$ ), связанных между собой каналами с относительно низкой пропускной способностью  $v_{i_1, i_2}$  ( $V = \{v_{i_1, i_2} | i_1, i_2 = \overline{1, m}\}$ ). В среде сети  $U$  предполагается обеспечить функционирование базы данных  $B$ , состоящей из  $n$  непересекающихся логических фрагментов  $f_j$  ( $B = \{f_j | j = \overline{1, n}\}$ ). Стратегии распределения фрагментов базы данных (БД) классифицируются в зависимости от количества задействованных ЛВС и наличия дублирования информации [1].

Для описания множества глобальных стратегий распределения баз данных  $\Omega = \{w_l | l \in N\}$  введем отображение  $\varphi$  множества логических фрагментов БД на множество групп ЛВС ( $\varphi : B \rightarrow \tilde{U}$ , где  $\tilde{U} = \{U_k \subset U\}$ ) [2].

При централизованной стратегии  $w_1$  единственная копия БД располагается на сервере одной из ЛВС сети, которую обозначим как  $U_0$ . Тогда:

$$\varphi(f_{j_1}) = \varphi(f_{j_2}) = U_0 \subset \tilde{U}, \quad \forall j_1, j_2 \in \overline{1, n};$$

$$\text{card}U_0 = 1.$$

При стратегии расчленения данных  $w_2$  логические фрагменты единственной копии базы данных распределяются по разным ЛВС, следовательно:

$$\text{card } \varphi(\mathbf{f}_j) = 1, \forall j \in \overline{1, n};$$

$$\exists j_1, j_2 \in \overline{1, n} \mid \varphi(\mathbf{f}_{j_1}) \neq \varphi(\mathbf{f}_{j_2}).$$

Стратегия дублирования  $\mathbf{w}_3$  характеризуется наличием нескольких полных копий БД на различных ЛВС, поэтому ее формализованное описание выглядит следующим образом:

$$\varphi(\mathbf{f}_{j_1}) = \varphi(\mathbf{f}_{j_2}) = U' \subset \tilde{U}, \forall j_1, j_2 \in \overline{1, n};$$

$$\text{card } U' > 1.$$

Для данной стратегии мощность подмножества  $U'$  определяет число полных копий базы данных.

Смешанная стратегия распределения логических фрагментов базы данных  $\mathbf{w}_4$  позволяет иметь несколько копий фрагментов на различных ЛВС, причем каждая ЛВС может содержать произвольный фрагмент. При этом на отображение  $\varphi$  не накладываются дополнительные ограничения, а число копий фрагмента  $\mathbf{f}_i$  определяется как мощность подмножества  $\varphi(\mathbf{f}_i)$ .

Для оценки вышеописанных стратегий определим над множеством  $\Omega$  множество показателей оценки качества функционирования  $\mathbf{G} = \{\mathbf{g}_l\}$  так, что  $\mathbf{g}_l: \Omega \rightarrow \mathbf{R}$  [3]. В частности, множество  $\mathbf{G}$  должно включать следующие элементы:  $\mathbf{g}_1$  - оценка надежности функционирования БД;  $\mathbf{g}_2$  - оценка степени локализации ссылок;  $\mathbf{g}_3$  - оценка среднего времени доступа, учитывающая для рассматриваемой стратегии архитектуру сети и пропускные способности каналов  $\mathbf{v}_{i_1, i_2} \in \mathbf{V}$ ;  $\mathbf{g}_4$  - оценка затрат на реализацию проекта;  $\mathbf{g}_5$  - оценка затрат ведения базы данных. Для каждого  $\mathbf{g}_i \in \mathbf{G}$  введем весовой коэффициент  $\alpha_i$ , который определяется исходя из требований, предъявляемых к системе управления базой данных.

Тогда выбор стратегии распределения логических фрагментов БД можно осуществить при помощи

$$\xi(\mathbf{w}_i) = \sum_{l \in \mathbf{L}} \alpha_l \mathbf{g}_l(\mathbf{w}_i),$$

где  $\mathbf{L} \subset \mathbf{G}$  – подмножество показателей, актуальных для данной БД.

Предложенный подход был использован для анализа функционирования различных баз данных административной вычислительной сети (АВС). Однозначный выбор глобальной стратегии размещения логических фрагментов без предварительного анализа был невозможен, так как в каждом

конкретном случае каждая стратегия обладала как преимуществами, так и недостатками перед остальными.

Основным преимуществом стратегий  $W_1$ , безусловно, является простота реализации, хотя по остальным показателям она уступает стратегиям  $W_2 \div W_4$ . Стратегия  $W_3$  превосходит остальные по надежности функционирования системы, по доступности и эффективности выборки данных, однако имеет место очень большая избыточность данных и тенденция к нарушению согласованности копий. Стратегия  $W_2$  позволяет более равномерно распределить нагрузку между ЛВС по сравнению со стратегиями  $W_1$  и  $W_3$ , однако для БД с малой локализацией ссылок резко увеличивается время обработки транзакций и уменьшается надежность системы.

Стратегия  $W_4$  объединяет подходы, связанные с расчленением и дублированием данных с целью приобретения преимуществ, которыми они обладают. Но к сожалению, эта стратегия приобретает сложности каждого из объединяемых подходов. Она является общей в том, что любая часть БД может быть дублирована произвольное количество раз и в каждом узле может содержаться желаемая часть БД. Недостатком стратегии является необходимость хранить информацию о том, где находятся данные в сети, и согласовать произвольное количество хранимых документов, связанных с каждым логическим фрагментом (ЛФ). Обработка и оптимизация запросов являются при использовании смешанной стратегии нетривиальными задачами. Однако, ключевым преимуществом смешанной стратегии является гибкость. Например, можно установить компромисс между объемом памяти, используемой в целом и в каждом отдельном узле, обеспечиваемым уровнем надежности и различными мерами эффективности. Так, архивные данные необходимо запоминать только в одном месте, напротив, более критические данные могут быть дублированы, если требуется достичь требуемого уровня надежности. При дублировании ЛФ, запоминании более одного фрагмента, стоимость согласования, включая стоимость связи, возрастает, однако большее количество данных становится локально доступным, что ведет к снижению качества пересылок и стоимости связи при выполнении запросов.

Хотя распределенная СУБД, реализующая стратегию  $W_4$ , и является предельно гибкой, остается проблема взаимозависимости различных факторов, влияющих на производительность системы, ее надежность и требования к памяти. Изолировать один фактор от другого весьма трудно.

Таким образом, учитывая свойства рассмотренных стратегий распределения фрагментов, можно сделать вывод, что стратегия  $W_1$ , допускающая лишь централизованное распределение, является простейшей и применяется для небольших БД с невысокими требованиями к надежности системы и не-

большими объемами информации. Стратегия  $W_2$  наиболее подходит для случая, когда либо внешняя память ограничена по сравнению с объемом БД, либо недостаточна надежность централизованной БД, либо когда должна быть повышена эффективность функционирования программных средств обработки транзакций. Стратегия  $W_3$  применяется для тех ситуаций, когда фактор надежности является критическим, база небольшая, а интенсивность обновления может быть невысокой, например, БД с интенсивными запросами справочного типа. И наконец, стратегия  $W_4$  является приемлемой тогда, когда ни одна из более простых стратегий не является удовлетворительной.

## ЛИТЕРАТУРА

1. Тиоми Т., Фрай Дж. Проектирование структур баз данных // М.: Мир, 1985. – 348 с.
2. Rothnie I.B. Introduction to a System for Distributed Databases // ACM Trans. Database Syst. – 1980. – № 5. – Р. 1 – 17.
3. Костюк В.И., Дешко А.И., Игнатенко Б.В. Проектирование информационных моделей в гибких системах // К. – Вища школа, 1987. – 194 с.

---

УДК 681.31

## ПРОЕКТИРОВАНИЕ ВСТРОЕННОГО ДИАГНОСТИРОВАНИЯ АДАПТИВНЫХ САМООРГАНИЗУЮЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ

к.т.н. О.И. Богатов  
( представил д.т.н., проф. Г.А. Поляков )

В статье проведен анализ нисходящего проектирования и определены задачи проектирования встроенных средств диагностирования адаптивных самоорганизующихся вычислительных комплексов (АСВК).

Одной из важнейших характеристик АСВК являются показатели надежности. Достижение высокого уровня надежности обеспечивается рядом технологических, эксплуатационных и организационных мероприятий.

Особое место среди них занимают методы диагностирования.

Лучшие результаты в обеспечении высоких показателей надежности могут быть получены применением как функциональных, так и тестовых средств диагностирования.

Использование традиционных методов тестового диагностирования для таких сложных систем, как АВСК, затруднено, а иногда и невозможно. Это связано в первую очередь с большой трудоемкостью разработки диагностического обеспечения и особенно построения тестов. Построение тестов является NP – полной задачей.

Реальный путь упростить решение задач тестового диагностирования – это применение тестопригодного проектирования (ТПР). Тестопригодное проектирование – это в общем случае способ проектирования логических систем по возможности с минимальными дополнительными затратами без изменения функций и параметров исходной схемы и с возможностью производить тестирование небольшим числом тестов, полученных простыми способами.

Для того чтобы ТПР было эффективным, его необходимо сделать частью единого процесса разработки АВСК. Средства ТПР должны войти как составная часть в состав традиционных средств нисходящего проектирования и быть совместимыми с уже существующими системами автоматизации проектирования.

Используемая в настоящее время методология нисходящего проектирования предполагает единую модель представления объекта проектирования, в которой используется три иерархических домена описания. Каждый из доменов служит для отдельного задания поведенческой, структурной и физической информации об объекте проектирования .

Описания в этих трех доменах иерархически декомпозированы на такие уровни абстракции как: архитектурный, алгоритмический, функционально-блочный, логический и схемный.

Каждый из перечисленных доменов может содержать как статический, так и динамический компоненты. Статический компонент служит для задания не зависящей от параметра времени части описания (операции, структуры и т.д.). Динамический компонент описывает порядок следования операций и время между операциями.

На основе данной модели процесс синтеза объекта проектирования

может быть определен следующим образом:

многовариантное преобразование информации структурного домена в информацию физического;

многовариантный переход от более высокого уровня абстракции в некотором домене к более низкому уровню абстракции в том же самом домене или в другом домене.

В общем случае нисходящее проектирование представляет собой процесс планирования разработки с учетом неопределенностей в исходных данных.

В процессе планирования должна существовать возможность исправления плана, если при проектировании становятся очевидными существенные его отличия от предполагаемых характеристик. Незначительные различия могут не учитываться, а будут исправлены в конце путем применения процедур восходящего проектирования.

Исходя из такой общей модели процесса проектирования, определим основные задачи по разработке встроенных средств тестирования АСВК и способы их решения для каждого из этапов проектирования .

Очевидно, что проектирование средств тестирования должно вестись параллельно разработке основных средств АСВК. Кроме того, эти два направления должны находиться в постоянном взаимодействии.

На *архитектурном* уровне проектирования производится выбор стратегии диагностирования. Исходными данными являются технические требования и ограничения, определяющие основной режим функционирования, требования надежности и ограничения, связанные с ее обеспечением, а также архитектура проектируемого комплекса.

Стратегия диагностирования включает в себя виды диагностирования (функциональное и тестовое) и алгоритм взаимодействия между ними, а также глубину диагностирования для каждого из видов. Глубина диагностирования - минимальная структурная единица, неисправность в которой может быть локализована.

Формальных методов решения задачи выбора стратегии диагностирования не существует, поэтому для получения решения должны быть использованы экспертные оценки.

На *алгоритмическом* уровне проектирования разрабатывается алгоритм встроенного самотестирования и средства его реализации.

Исходной информацией для этого является стратегия диагностирования, выбранная на предыдущем этапе. Алгоритм диагностирования определяет условия инициализации тестового режима, последовательность управления средствами диагностирования и выдачу сообщений о результатах проверки. Для реализации алгоритма необходимо иметь специальные средства управления, называемые тестовым монитором.

На *функционально - блочном* уровне проектирования решаются следующие задачи:

- выбор методов встроенного самотестирования для каждого блока;
- разработка средств встроенного самотестирования с выбранными методами;
- построение планов тестирования;
- разработка устройств управления тестовыми средствами и (или) модификация имеющихся БУ.

Исходные данные :

- статическая структура блоков в виде соединения таких функциональных узлов, как регистры, сумматоры, схемы сравнения, схемы И, ИЛИ и др. ;
- ограничения на тестовые средства, такие как дополнительные аппаратные затраты, количество дополнительных входов, выходов, время тестирования, полнота проверки, ухудшение динамических характеристик и другие;
- алгоритмы встроенного самотестирования, полученные на предыдущем уровне проектирования.

Разработка средств встроенного самотестирования заключается в размещении в проверяемом блоке генераторов тестовых воздействий и схем анализа выходных реакций.

Решение этой задачи зависит от соотношения времени тестирования и аппаратных затрат, а также от выбранного способа подачи тестов. Построение планов тестирования заключается в формировании последовательности шагов, обеспечивающих подачу тестовых воздействий с генератора на проверяемый функциональный узел и передачу реакций на схему анализа.

Данная задача решается формальными методами.

В случае, когда имеется возможность подачи воздействий с генератора

непосредственно на вход каждого функционального узла и съема реакций непосредственно с выхода на схему анализа, нет необходимости в решении данной задачи.

Разработка устройства управления тестовыми средствами производится аналогично разработке блока управления решением основной задачи [2,3]. Если же предусмотрено управление тестовыми средствами с помощью основного БУ, то при его проектировании добавляются эти функции.

На *логическом* уровне все средства АСВК проектируются как единое целое. Каждый блок разрабатывается с учетом конкретного элементного базиса, нагрузочной способности элементов и т.д.

Здесь же оцениваются дополнительные аппаратурные и временные затраты, связанные со встроенным диагностированием. Производится также вычисление показателей надежности, обеспечиваемых средствами встроенного диагностирования.

Таким образом, в статье проведен анализ нисходящего проектирования, определены задачи проектирования встроенных средств диагностирования АСВК и алгоритм проектирования встроенного диагностирования АСВК на вербальном уровне. Алгоритм может быть использован при построении автоматизированной системы проектирования.

## ЛИТЕРАТУРА

1. Богатов О.И., Горовой С.И. Активная защита параллельных процессоров. Сборник научных трудов “Информационные системы”, Выпуск 1(5), НАНУ, ХВУ, Харьков, 1997.

2. Поляков Г.А., Богатов О.И. и др. Принципы построения и основные результаты разработки адаптивных самоорганизующихся вычислительных комплексов АСУ ВС Украины. Сборник научных трудов, ХВУ, вып. N 2, Харьков, 1994.

3. Богатов О.И., Поляков Г.А., Проектирование параллельных декомпозиционных процессоров для специализированных АСУ. Труды международной конференции “Проблемы передачи и обработки данных” под ред. Королева А.В. НАНУ, ПАНИ, Харьков, 1994.