

РАЗБИЕНИЕ АЛГОРИТМОВ СТАТИСТИЧЕСКОГО АНАЛИЗА ДЛЯ РАСПРЕДЕЛЁННЫХ СППР

МИЛОВ А. В.

кандидат технических наук

ЧУЙКО И. М.

Харьков

Одной из сложностей проектирования распределенных систем принятия решений систем является всё возрастающая сложность разработки прежде всего распределенных систем статистического анализа исходных данных, на результатах которого базируется генерация и анализ принимаемых решений [1, 4]. Эти системы состоят из асинхронно выполняемых компонентов, обменивающихся между собой сообщениями. Для таких систем требуются адекватные методы проектирования, непосредственно предназначенные для решения проблем разбиения систем анализа на части. В работе представлен возможный подход к проектированию распределенной системы статистического анализа. Ключевым пунктом этого подхода к проектированию является представление алгоритма разбиения компонентов распределенной обработки (КРО) для кластеризации функциональных модулей. Такой подход к проектированию направлен на создание распределенных СППР, использующих потенциальную параллельность между функциональными модулями, систем, в которых отсутствует невыгодный трафик сообщений.

Проектирование централизованных систем анализа данных является довольно трудоемким процессом; проектирование распределенных систем анализа является еще более сложным. Крайне важные соображения, касающиеся параллелизма и обмена сообщениями ставят перед проектировщиками распределенного статистического анализа большое число задач с взаимноисключающими решениями, которые необходимо тщательно сбалансировать.

Распределенные системы анализа состоят из асинхронно выполняемых компонентов, обменивающихся между собой сообщениями. На этапе проектирования распределенного статистического анализа выполняются две важные работы: разбиение на части и привязка. На стадии разбиения мы имеем дело с установление границ внутри анализирующей системы. На стадии привязки мы используем границы, установленные на стадии разбиения, с целью привязки компонентов системы статистического анализа к обрабатывающим модулям, составляющим архитектуру распределенной системы. В наиболее общем случае привязка может осуществляться, начиная со стадии выработки требований и заканчивая тогда, когда процессы распределяются по модулям динамически. Это дает возможность динамически пере-

конфигурировать системы, повышая тем самым невосприимчивость распределенной системы к сбоям.

В предлагаемых ранее стратегиях разбиения и привязки не обращалось внимания на то, как получаемые решения согласуются с решением общей задачи проектирования распределенной системы анализа [2, 3]. Главным в этих методах является минимизация затрат на связь в системе. Другие подходы ориентированы не столько на оптимизацию, сколько на прикладные задачи. В работе представлена стратегия разбиения для проектирования распределенных систем статистического анализа, основными целями которой является достижение иерархичности систем анализа, использование потенциальной параллельности между функциональными модулями и исключение невыгодного трафика сообщений. В результате структурирования системы в виде иерархии модулей можно ожидать повышения надежности системы и упрощения в обслуживании. Трафик сообщений невыгоден, если он не способствует использованию потенциальной параллельности. Так, если два модуля должны быть связаны между собой, но не могут выполняться параллельно, то они должны находиться в одном и тот же узле обработки распределенной системы анализа. В противном случае связь между ними будет невыгодной. На самом деле она даже будет стоить некоторого времени выполнения программы и, возможно, надежности. Конечной задачей разбиения является повышение параллелизма в работе.

Подход к проектированию включает в себя ряд последовательных этапов. Порядок следования этих этапов показан на рис. 1.

Дадим определение, используемое на этапе 2.

Граф структурной схемы $SCG = (N, A)$ – это направленный граф (полученный из структурной схемы), имеющий множество вершин N и множество направленных ребер A , таких, что каждый модуль t_j структурной схемы представлен одной вершиной n_i из множества N , а ребро (n_i, n_j) является элементом множества A , если модуль i в структурной схеме может вызывать модуль j .

Поскольку SCG выводится из структурной схемы, то ясно, что SCG является связным графом, имеющем одну «корневую» вершину r , то есть не существует ни одной вершины n_p такой, что ребро (n_p, r) является элементом множества A .

Перечислим этапы проектирования, а затем вкратце обсудим некоторые из этапов.

- 1) Выполнение функционального разбиения (декомпозиция) с использованием принципов структурного проектирования.
- 2) Получение графа структурной схемы для функциональных модулей.
- 3) Определение отношений для непосредственно подчиненных модулей.

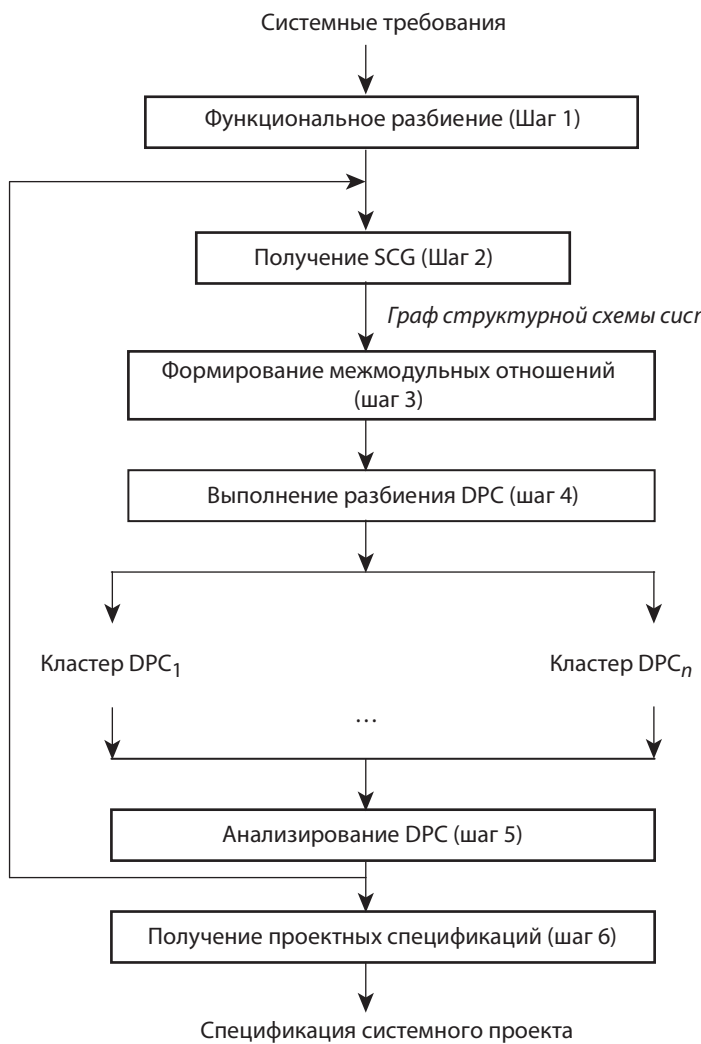


Рис. 1. Обзор подхода к проектированию

- 4) Выполнение разбиения DPC.
- 5) Выполнение анализа DPC.
- 6) Получение проектных спецификаций.

Предлагается сделать начальную фазу проектирования распределенного статистического анализа такой же, как и при проектировании централизованной системы. В частности, выберем для этой начальной фазы метод структурного проектирования. В нашем методе проектирования *этап 1* предназначен для гарантии того, что система состоит из большого числа слабо связанных модулей, организованных иерархически. Такие свойства проектирования находят свое отражение в системах, являющихся модульными в отношении системных функций. Системы такого типа считаются, в целом, более надежными и простыми в обслуживании, нежели иерархические. Наиболее важным итогом этого этапа является создание системной структурной схемы, которая графически иллюстрирует иерархию модулей и модули, вызываемые каждым из них.

Этап 2 – это, в основном, процесс трансляции. Определение описывает преобразование условной структурной схемы или диаграммы вызова модулей в граф структурной схемы.

На *этапе 3* нашего подхода к проектированию определяются соотношения между непосредственно подчиненными модулями. Следует заметить, что каждый модуль в функционально разделенной системе может иметь непосредственно подчиненные модули (то есть модули, которые он непосредственно вызывает), а может их и не иметь. Назначением этого этапа является определение для каждого модуля отношений всех непосредственно подчиненных ему модулей. Отношение между модулями связано с «порядком следования» модулей и может быть последовательным, избирательным или параллельным. Например, модуль *M* может вызывать модули *A*, *B* и *C*, из которых *A* и *B* могут выполняться параллельно, а модуль *C* должен ждать завершения работы как *A*, так и *B*. Тогда мы можем выразить отношение между этими модулями путем использования уравнения порядка следования для модуля *M*: $M \Rightarrow SEQ(CON(A, B), C)$. *SEQ* используется для обозначения последовательности модулей (слева направо), а *CON* – для обозначения параллельного выполнения модулей. Другим обозначением, которым мы будем пользоваться, является *ONE-OF*, применяемое для обозначения выбора одного модуля из нескольких.

Для того, чтобы точно определить эти отношения, проектировщик должен принять во внимание функции модулей и иметь представление о внутреннем строении этих модулей. Утверждение, что два модуля должны быть последовательными, зависит от того, используется ли выход одного из модулей непосредственно в качестве входа другого, или нет.

Если два модуля должны выполняться последовательно (избирательность может рассматриваться как частный случай последовательной работы, при которой один из модулей не выполняется), то эти два модуля не могут выполняться одновременно. Мы должны отметить, что проектировщик может полагаться на свой опыт и творчески подходить к определению межмодульных отношений, который он хочет передать на последующие этапы проектирования.

Наиболее важным этапом является *этап 4*, разбиение DPC. Целью разбиения DPC является перевод функциональных составляющих, полученных на этапе 1, в кластеры DPC, которые гораздо удобнее привязывать к модулям обработки распределенной системы. Можно просто допустить, что каждый функциональный модуль этапа 1 становится своим собственным кластером DPC. Это решило бы задачу потенциального межмодульного параллелизма, но в ущерб задаче избегания невыгодного трафика сообщений. Кроме этого, число кластеров DPC и, следовательно, необходимых узлов обработки было бы равно числу функциональных модулей. Это не привело бы к достаточно эффективному использованию ресурсов обработки. Вместо этого предлагается алгоритм разбиения

DPC, создающий набор кластеров DPC, использующий потенциальный межмодульный параллелизм, но не приводящий к невыгодному трафику сообщений.

Этап 5 включает в себя анализ результатов этапа разбиения DPC; мы представляем его, используя обозначения ряда руководств по проектированию.

Заключительный этап подхода, *этап 6*, предназначен для выработки проектных спецификаций, отражающих организационные и связанные требования к системе программного обеспечения.

Одним из *достоинств* предлагаемого подхода к проектированию является то, что он действует как дополнение к традиционным подходам к проектированию систем статистической обработки; архитектура разработанной системы основана на иерархии функциональных модулей. Что касается этой иерархии, то проектировщик должен дать информацию об отношениях модулей, являющихся подчиненными общего модуля. Это согласуется с принципом скрытия информации. Алгоритм DPC определяет отношения параллельности на системном уровне, основываясь на заданных проектировщиком отношениях на уровне отдельных модулей.

Алгоритм разбиения DPC имеет тот *недостаток*, что в нем не полностью рассматриваются результаты

«стоимости связей против прибыли от обработки», касающиеся размещения модулей в кластерах DPC. На практике, чтобы сделать такой выбор, приходится рассматривать большое число параметров (которые бывает трудно определить количественно), таких как системная топология и выполнение модулей. Именно это и явилось причиной включения в подход к проектированию этапа 5. В свете этого метод разбиения DPC лучше всего рассматривать как всего лишь один из этапов эвристического подхода. ■

ЛИТЕРАТУРА

1. **Баженова И. Ю.** Разработка распределенных приложений / И. Ю. Баженова. – М. : Изд-во МГУ им. М. В. Ломоносова, 2007. – 146 с.
2. **Миков А. И.** Распределенные системы и алгоритмы / А. И. Миков, Е. Б. Замятина. – М. : Изд-во INTUIT, 2008. – 344 с.
3. **Нечепуренко М. И.** Алгоритмы и программы решения задач на графах и сетях / М. И. Нечепуренко. – М. : Наука, 1990. – 428 с.
4. **Таненбаум Э.** Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Стеен. – СПб. : Питер, 2003. – 877 с.