

УДК 621.833.001.57

О.А. Усова, специалист,
Ю.В. Ляхов, магистр,
Одес. нац. политехн. ун-т

ПРОЕКТИРОВАНИЕ СЕТЕЙ ДЛЯ МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

О.А. Усова, Ю.В. Ляхов. **Проектування мереж для моделювання складних систем.** Запропоновано методи досягнення високої продуктивності системи управління базою даних за рахунок алгоритмів операцій вводу-виводу, обробки запитів до бази даних, засобів та утиліт.

O.A. Usova, Yu.V. Lyakhov. **Designing the complex systems simulation networks.** The paper deals with the present tasks of resource allocation in designing the networks for complex systems simulation. Some methods are proposed, with high efficiency of a data-base control system achieved with input-output operations algorithms, data-base query processing, means and utilities.

С развитием технологий распределенных вычислений изменяются и требования к вычислительным сетям. Основная задача разработки сетей распределенных систем при моделировании сложных систем заключается в перемещение больших объемов данных, необходимых для распределенных вычислений. Новые требования к производительности сетей, предъявляемые современными приложениями, такими как мультимедиа, распределенные вычисления, системы оперативной обработки транзакций, вызывают насущную необходимость расширения соответствующих стандартов.

Исследование параметров вычислительной системы при различных характеристиках отдельных компонентов позволяет выбрать сетевое и вычислительное оборудование с учетом производительности, качества обслуживания, надежности и стоимости [1]. Поскольку стоимость одного порта активного сетевого оборудования в зависимости от производителя оборудования, используемой технологии, надежности, управляемости может меняться от десятков долларов до десятков тысяч, моделирование позволяет минимизировать стоимость оборудования, предназначенного для использования в вычислительной системе. Однако ни одно из средств не способно охватить все задачи, поэтому если необходимо смоделировать сеть и проанализировать ее работу, придется воспользоваться несколькими продуктами. Имеются также заметные различия между продуктами, которые, как утверждается, решают одни и те же задачи.

В комплект поставки многих программных продуктов, для моделирования сетей сложных систем, входят библиотеки сетевых элементов, устройств и протоколов, отнюдь не все продукты способны моделировать одни и те же объекты. Следует обязательно выяснить, работу каких сетевых элементов способны рассчитывать то или иное средство. Большинство продуктов рассчитывает, как будут работать элементы сети, о которых имеются данные, однако и данные не всегда являются полными и избыточными. Кроме того, средства моделирования сетей имеют несколько ограниченные возможности учета воздействия на пропускную способность сети работы с приоритетами и уровнями обслуживания.

Предлагается решение задачи распределения ресурсов при проектировании таких сетей ЭВМ выполнить в несколько этапов: предпроектный; общий структурный синтез сети; разработка протоколов и средств административной системы сети; проектирование программно-технических средств сети; проектирование топологии сети; распределение ресурсов сети между абонентами.

Исходными для задачи распределения ресурсов в проектируемой сети и расчета объема вычислительных работ терминалов в ней являются следующие данные: T — заданный календарный период работы центрального вычислительного комплекса (ЦВК); A — множество заявок абонентов, где каждая заявка $a \in A$ характеризуется параметрами: $z(a)$ — тип

использования условного терминала, $r(a)$ — число терминал-часов, требуемых для $\delta(a) = \begin{cases} 0 \\ 1 \end{cases}$ заявки a , $s(a)$ — вид обработки, при $\delta(a) = 0$ имеет место обработка с локальных условных терминалов ЦВК, а при $\delta(a) = 1$ — с удаленных, $\tau(a) = (i, j)$ — пара, определяющая интервал времени, в течение которого необходимо предоставить ресурс для выполнения заявки a (i — начальный, j — конечный квант времени); $h(a)$ — вектор, определяющий в терминал-часах ограничение снизу на интенсивность предоставления ресурса заявке a по квантам календарного периода; $T_{\text{пр}}$ — время полезной работы ЦВК в часах.

В результате распределения ресурсов абонентов для каждой заявки $a \in A$ формируется вектор $\mathbf{a}(a_1 \dots a_T)$ размерности T , представляющий собой расписание работы системы над заявкой a .

Рассмотрим алгоритм для распределения ресурсов при создании терминалов в указанной сети. Пошаговая работа алгоритма может быть представлена последовательностью распределений

$$G = \gamma_0, \dots, \gamma_1, \dots, \gamma_n, \quad n = A,$$

где γ_i — распределение, полученное в результате работы алгоритма над i -й заявкой. Предложенный алгоритм достаточно эффективен. Как показывает проверка ряда расчетов [1], время выполнения алгоритма не превышает зависимость $n^2 T = |A|$, где $n = |A|$.

При моделировании сложных систем возникает проблема организации обработки распределенных данных. Под распределенной базой данных обычно понимают множество фрагментов логически единой базы данных, которые располагаются на различных узлах сети, и, возможно, управляются различными системами управления базами данных (СУБД).

Одним из способов достижения более высокой производительности СУБД является использование алгоритмов преобразования операций обработки данных. Можно выделить три типа таких алгоритмов: операции ввод/вывод, средства и утилиты администрирования, обработка запросов к базе данных.

В отличие от выполнения операций ввода/вывода и администрирования эффективность обработки запросов реализуется гораздо сложнее. Обработку запроса можно представить как множество атомарных операций, состав и последовательность которых, т.е. план запроса, определяются после просмотра нескольких вариантов.

Выполнение запроса определяет строгую последовательность выполнения каждой операции, результат которой служит исходным набором данных для выполнения следующей. Для обеспечения эффективности необходимо выбрать такой набор атомарных операций, с помощью которых можно выполнить часть задачи, и результаты работы каждой операции можно было бы объединить для общего представления.

Упрощения при вертикальной фрагментации можно достичь путем удаления вертикальных фрагментов, которые не имеют общих атрибутов с условием проекции.

Пусть база данных разделена на вертикальные фрагменты

$$V_1 : \pi_{1D,A,B}(R), \quad V_2 : \pi_{1D,C,D}(R).$$

Пусть необходимо выполнить запрос вида SELECT A, B FROM R .

Схематически такой запрос можно представить как последовательность операций $\pi_{A,B}((\pi_{1D,A,B}(R))JOIN(\pi_{1D,C,D}(R)))$. После применения правила коммутативности проекции и соединения, т.е. раскрытия скобок, получен результат операции проекции V_2 — пустое отношение и, следовательно, выполнение запроса можно свести к операции $\pi_{A,B}(\pi_{1D,A,B}(R))$.

Для горизонтальной фрагментации базы данных характерна избыточность некоторых операций соединения, и их исключение может значительно снизить затраты на выполнение запроса в целом.

Пусть база данных разделена на горизонтальные фрагменты

$$G_1 : \sigma_{ID=1 \& A=a}(R), G_2 : \sigma_{ID=1 \& B=b}(R_1), G_3 : \sigma_{ID=1}(R_1), G_4 : \sigma_{ID=1}(R_2), G_5 : \sigma_{ID=1}(R_2).$$

Необходимо выполнить запрос $SELECT * FROM R1 k, R2 M WHERE k.ID = m.ID AND k.B = b$
Выполнение запроса определяется следующей последовательностью:

$$(\sigma_{B=b}((\sigma_{ID=1 \& A=a}(R_1)) \cup (\sigma_{ID=1 \& B=b}(R_1)) \cup (\sigma_{ID=1}(R_1)))) JOIN ((\sigma_{ID=1}(R_2)) \cup (\sigma_{ID=1}(R_2))).$$

При более детальном разложении последовательности операций можно заметить, что последовательность $\sigma_{B=b}(R_1) = \sigma_{B=b}(\sigma_{ID=1 \& A=a}(R_1))$ является избыточной, т.е. не вносит в результат ни одного кортежа. Кроме этого, предикат операции селекции является подмножеством фрагмента G_2 .

После упрощения операции соединения и объединения последовательность операций можно переписать в виде

$$\begin{aligned} & ((\sigma_{ID=1 \& B=b}(R_1)) JOIN (\sigma_{ID=1}(R_2))) \cup ((\sigma_{ID=1 \& B=b}(R_1)) JOIN (\sigma_{ID=1}(R_2))) \cup \\ & \cup ((\sigma_{B=b}(\sigma_{ID=1 \& A=a}(R_1))) JOIN (\sigma_{ID=1}(R_2))) \cup \\ & \cup ((\sigma_{B=b}(\sigma_{ID=1 \& A=a}(R_1))) JOIN (\sigma_{ID=1}(R_2))). \end{aligned}$$

Поскольку второе и третье соединения не вносят в результат ни одного кортежа, их можно исключить. Таким образом, окончательный результат можно записать в виде последовательности

$$((\sigma_{ID=1 \& B=b}(R_1)) JOIN (\sigma_{ID=1}(R_2))) \cup ((\sigma_{B=b}(\sigma_{ID=1 \& A=a}(R_1))) JOIN (\sigma_{ID=1}(R_2))).$$

Основная проблема, возникавшая при модификации запроса, заключается в том, что модель стоимости, которая использовалась при оптимизации, со временем становится неточной как из-за изменения размера фрагментов, так и из-за реорганизации базы данных. В связи с этим решение вопросов о том, какие данные и на каких узлах сети должны храниться, является достаточно сложным. Эти вопросы относятся к проблемам физического проектирования баз данных для распределенных систем и во многом зависят от решения задачи размещения фрагментов.

При проектировании сетей для моделирования сложных систем необходимо применение концептуальной модели, в основе которой лежит проектирование действий и конфигураций структуры системы, определение и задание начальных состояний системы, формирование множества допустимых и аварийных состояний системы (объекта) менеджмента.

Литература

1. Конноли Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Конноли Т., Бегг К., Страчан А.: Пер. с англ. — М.: Вильямс, 2000. — 364 с.

Поступила в редакцию 19 мая 2007 г.