

БАЗЫ ДАННЫХ

часть II

Параллельные архитектуры баз
данных

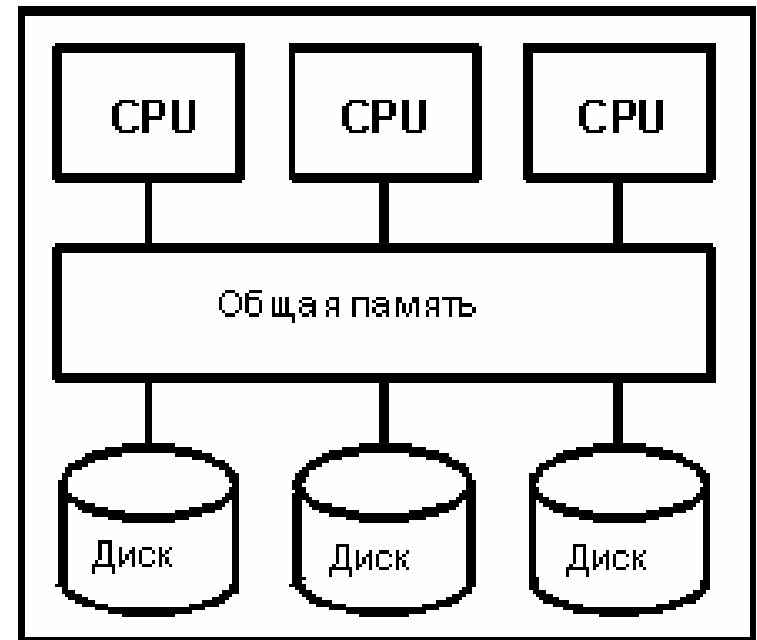
Параллельные архитектуры БД

Основные параллельные архитектуры

Параллельные архитектуры БД

Фактически определились три архитектурных направления:

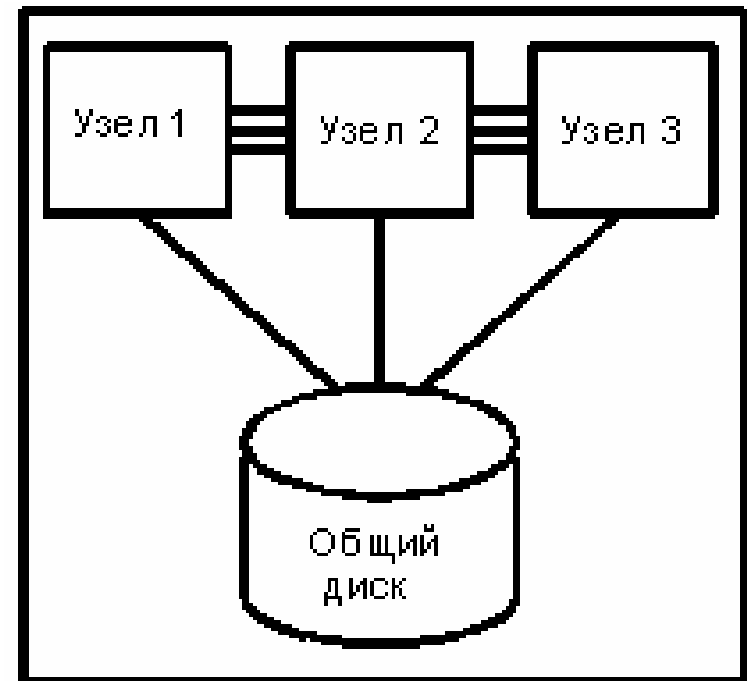
1. Симметричные многопроцессорные системы (SMP) - наиболее часто используемая форма сильносвязанных многопроцессорных систем, т.е. систем, разделяющих единую оперативную память и наиболее часто - дисковую подсистему.



SMP система

Параллельные архитектуры БД

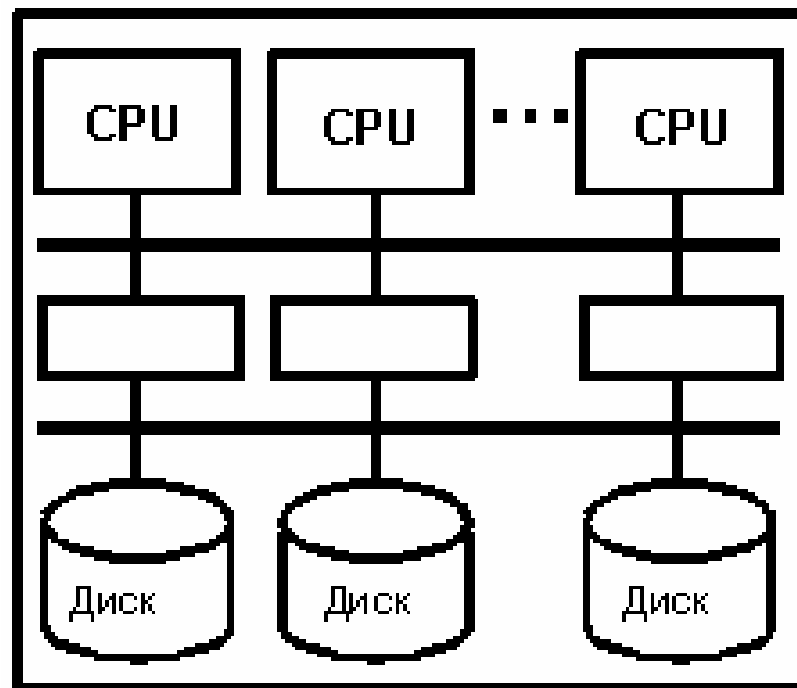
2. Слабосвязанные многопроцессорные системы - совокупность самостоятельных компьютеров, объединенных в единую систему быстродействующей сетью и, возможно, имеющих общую дисковую подсистему, как, например, кластерные инсталляции;



Кластер

Параллельные архитектуры БД

3. Системы с массовым параллелизмом (MPP) - системы с сотнями и даже тысячами процессоров, детали их реализации могут значительно различаться.



MPP система

Параллельные архитектуры БД

Наиболее оптимальными с точки зрения стоимости и прозрачности наращивания можно считать симметричные многопроцессорные платформы (SMP). Добавление процессоров в них обходится относительно дешево, и при использовании соответствующих программ не требует изменения программного обеспечения или принципов администрирования, причем, начиная уже с однопроцессорных систем. Для более дорогостоящих и ответственных систем необходимый уровень резервирования может быть достигнут с помощью кластеров, в т.ч. состоящих из SMP-систем.

Параллельные архитектуры БД

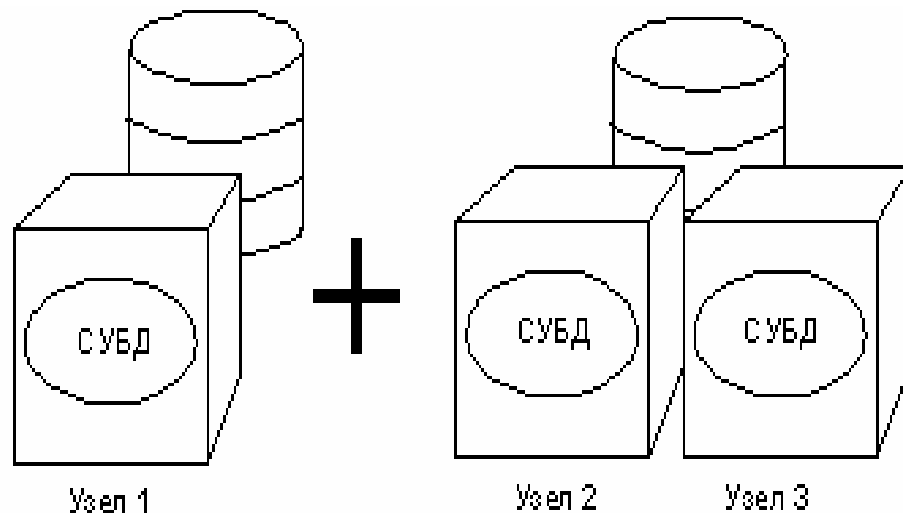
Можно выделить четыре группы требований, определяющих с технической точки зрения потребительские качества современной СУБД:

- 1.масштабируемость;**
- 2.производительность;**
- 3.возможность смешанной загрузки разными типами задач;**
- 4.обеспечение постоянной доступности данных.**

Параллельные архитектуры БД

Масштабируемость - такое свойство вычислительной системы, которое обеспечивает предсказуемый рост системных характеристик при добавлении к ней вычислительных ресурсов. В случае сервера СУБД можно рассматривать два способа масштабирования - **вертикальный** и **горизонтальный**.

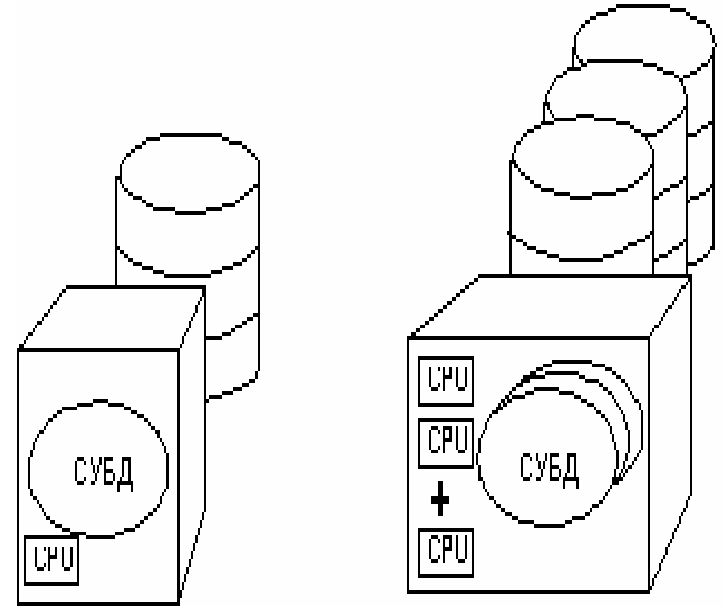
При **горизонтальном** подходе увеличивается число серверов СУБД, возможно, взаимодействующих друг с другом в прозрачном режиме, разделяя таким образом общую загрузку системы.



Горизонтальное масштабирование

Параллельные архитектуры БД

Вертикальное масштабирование подразумевает увеличение мощности отдельного сервера СУБД. Хорошим примером может служить увеличение числа процессоров в симметричных многопроцессорных (SMP) платформах. При этом программное обеспечение сервера не должно изменяться, например, требовать дополнительных модулей, т.к. это увеличило бы сложность администрирования и ухудшило предсказуемость системы.



Вертикальное масштабирование

Параллельные архитектуры БД

Для вертикального масштабирования все чаще используются симметричные многопроцессорные системы (SMP), поскольку в этом случае не требуется смены платформы, т.е. операционной системы, аппаратного обеспечения, а также навыков администрирования.

При оценке сервера СУБД на базе SMP платформы стоит обратить внимание на две основные характеристики расширяемости архитектуры: **адекватность** и **прозрачность**.

Параллельные архитектуры БД

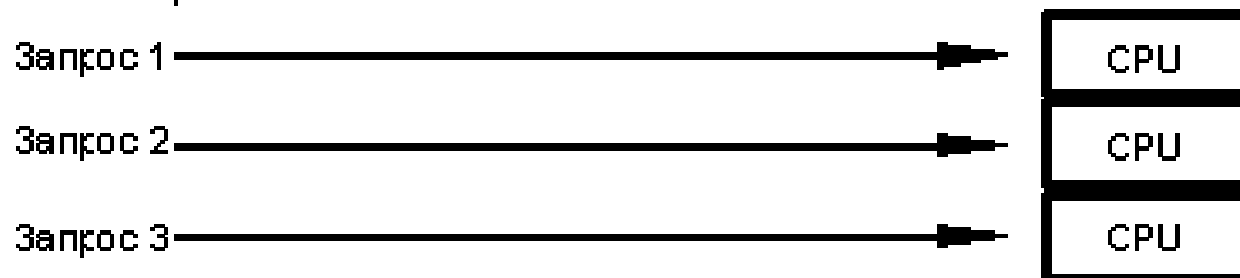
Свойство адекватности требует, чтобы архитектура сервера равно поддерживала один или десять процессоров без переинсталляции или существенных изменений в конфигурации, а также дополнительных программных модулей.

Приложение не должно учитывать подробности реализации аппаратной архитектуры - способы манипулирования данными и программный интерфейс доступа к базе данных обязаны оставаться одинаковыми и в равной степени эффективными (прозрачность).

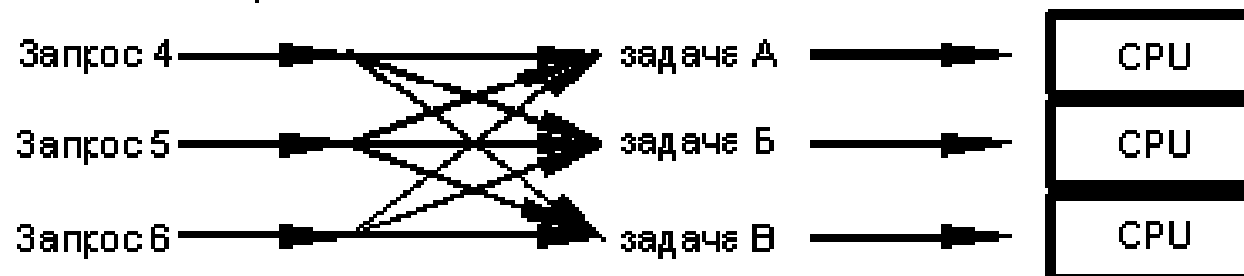
Параллельные архитектуры БД

Запросы могут обрабатываться последовательно несколькими задачами или разделяться на подзадачи, которые в свою очередь могут быть выполнены параллельно.

Параллельные транзакции:



Параллельные запросы:



Параллельные архитектуры БД

Среди многих факторов, влияющих на **производительность СУБД**, рассмотрим два, имеющие прямое отношение к нынешним параллельным вычислительным платформам:

1. **поддержка параллелизма;**
2. **реализация многопоточковой архитектуры**

Параллельные архитектуры БД

Одним из способов достижения более высокой производительности является использование алгоритмов распараллеливания заданий.

В СУБД существует три области применения таких алгоритмов:

1. параллельный ввод/вывод;
2. параллельные средства и утилиты администрирования;
3. параллельная обработка запросов к базе данных.

Параллельные архитектуры БД

Распараллеливание ввода/вывода в сочетании с оптимальным планированием заданий позволяет осуществить крайне эффективный одновременный доступ к фрагментированным таблицам и индексам, расположенным на нескольких физических дисках.



Параллельные архитектуры БД

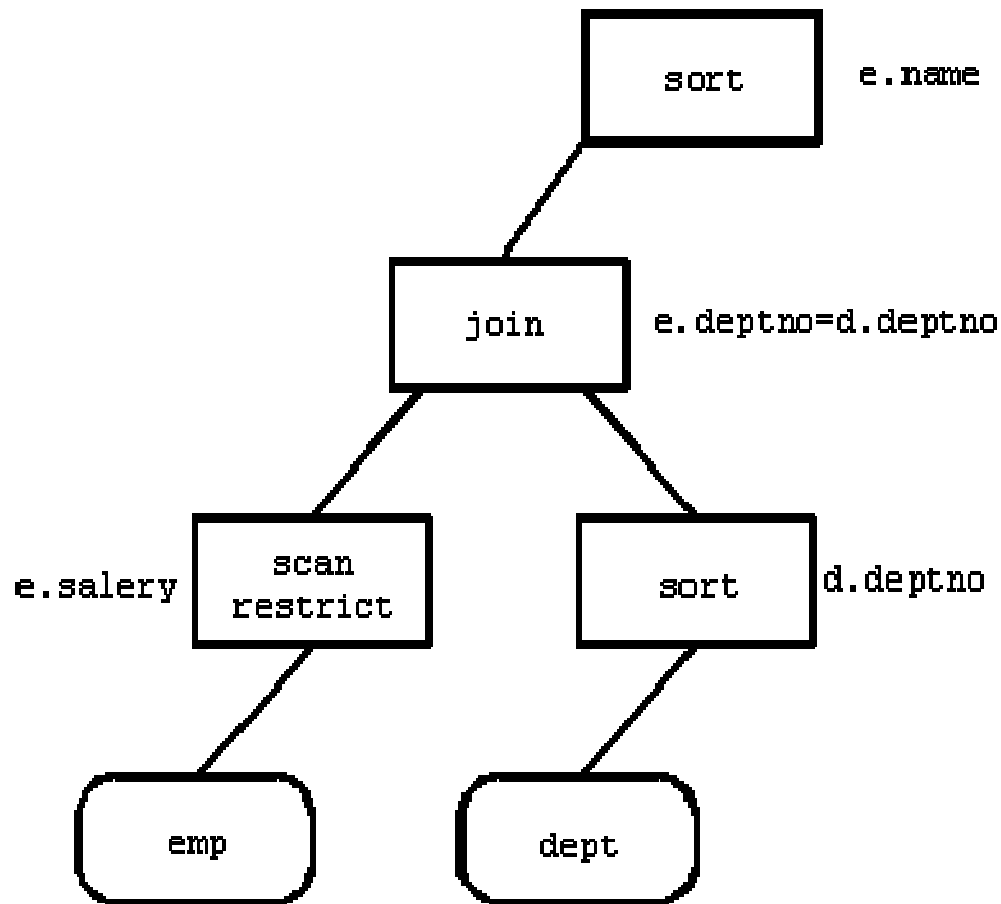
Выполнение административных утилит (реорганизация данных, в т.ч. сортировки, построение индексов, загрузка и выгрузка данных, архивирование и восстановление баз данных) также должно быть полностью параллельным, включая распараллеливание операций ввода/вывода, хотя на практике это требует доработки лишь небольшого числа механизмов.

Параллельные архитектуры БД

Теоретическим обоснованием возможности распараллеливания запросов в реляционной СУБД является свойство реляционного замыкания.

Параллелизм в обработке запросов подобен идее "разделяй и властвуй" или коллективному выполнению задания.

```
select * from emp e dept d
where e.salary>2000
and e.deptno=d.deptno
order by e.name
```



Параллельные архитектуры БД

Достижение этой цели требует, чтобы декомпозиция и параллельное выполнение запроса были осуществимы независимо от средств межзадачного взаимодействия, принятых в конкретной вычислительной системе.

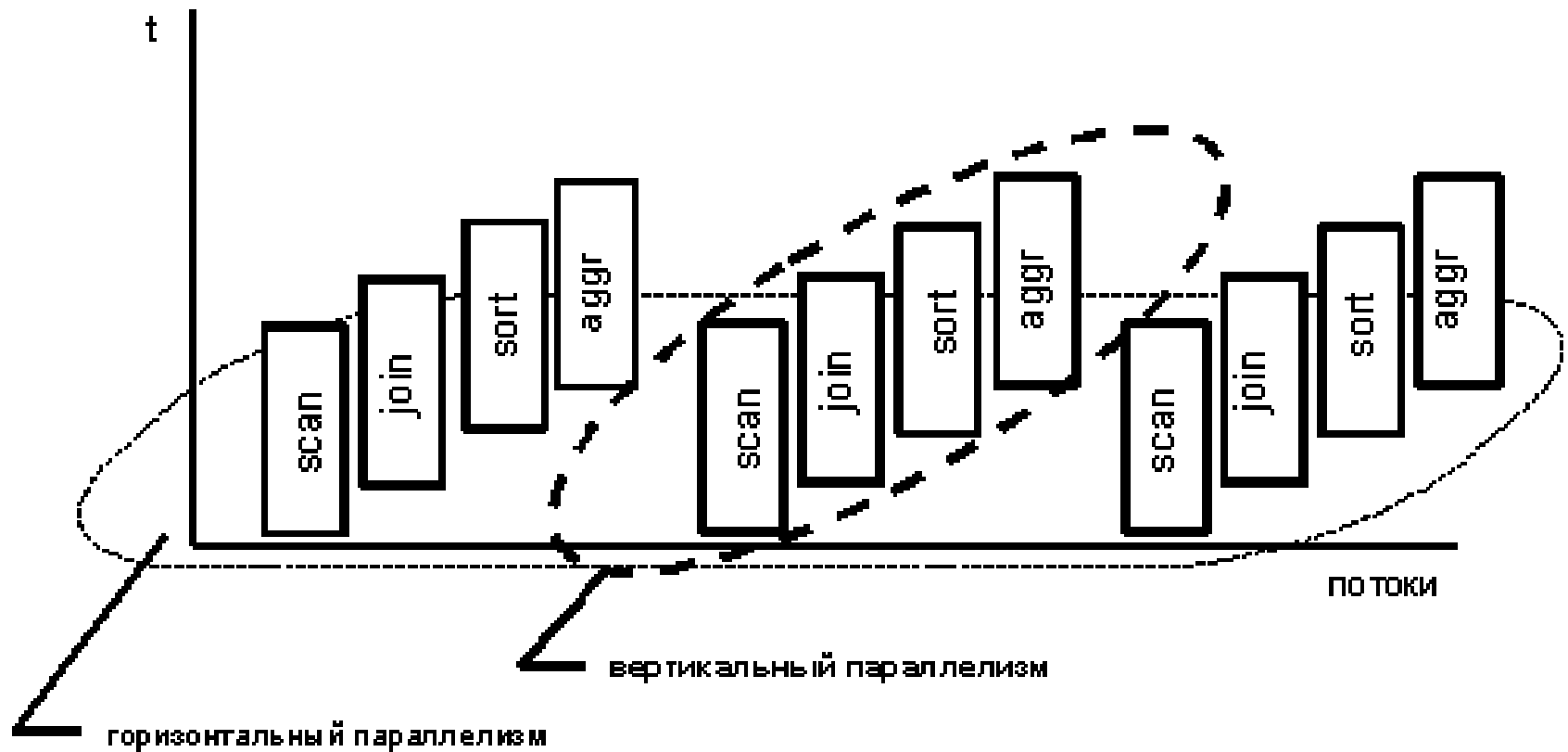
Создание множества тиражируемых копий одной атомарной операции, одновременно выполняющих одну и ту же задачу, можно назвать **горизонтальным параллелизмом**.

Параллельные архитектуры БД

Дополнительная степень параллелизма - **вертикальный параллелизм** - может быть достигнута при параллельном запуске нескольких, в общем случае различных атомарных операций, которые обычно должны были бы выполняться последовательно.

Такой подход, во многом сходный с конвейерной обработкой в современных процессорах, носит название "потока данных" (data flow) или "управления по требованию" (demand driven).

Параллельные архитектуры БД

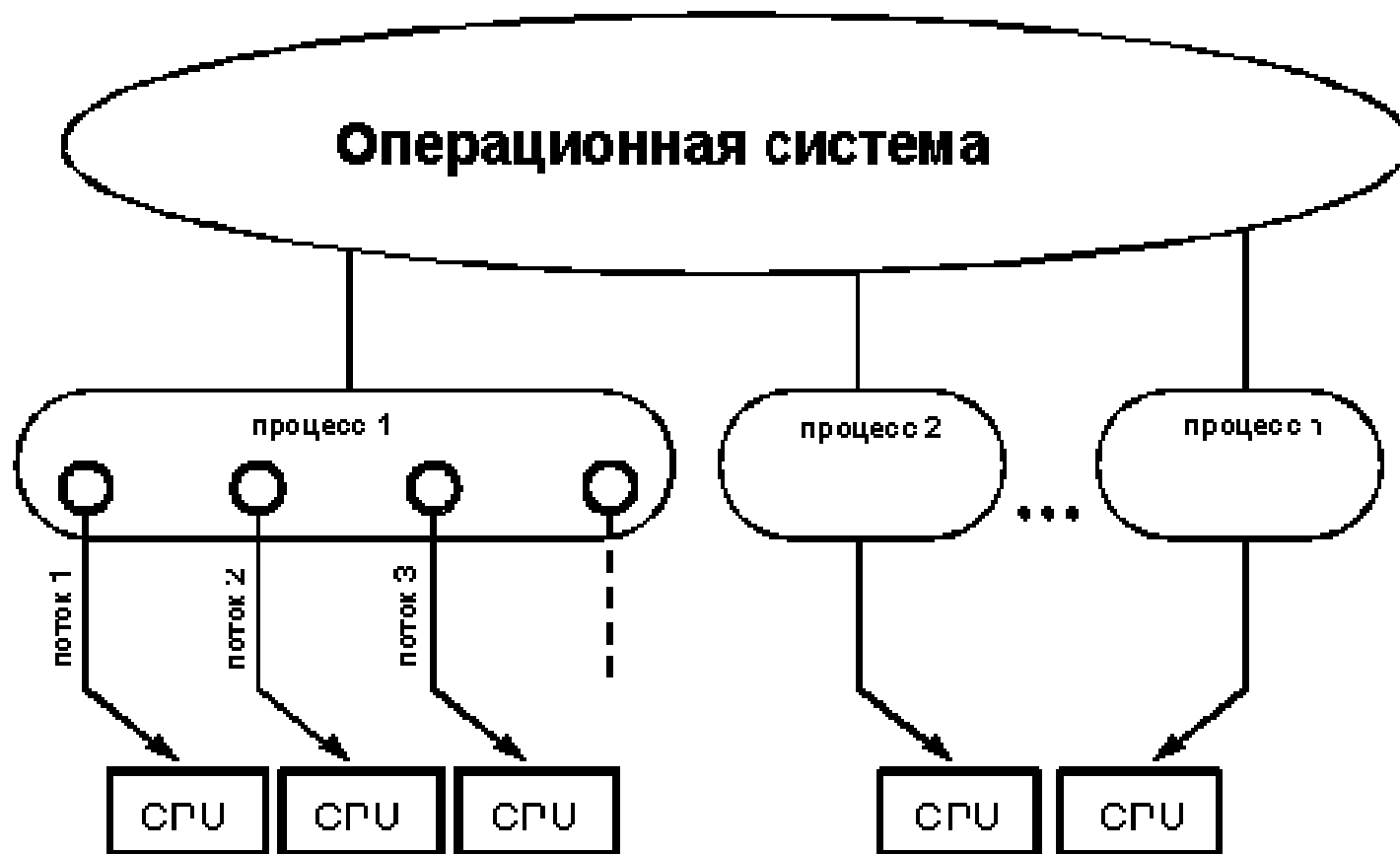


Параллельные архитектуры БД

Способ организации одновременной обработки множества запросов внутри сервера баз данных при минимальных накладных расходах на переключение контекста и максимальном разделении вычислительных ресурсов между ними называется истинно многопоточковой архитектурой .

Поток (thread) - это фрагмент контекста одного процесса, включающий только те данные (стек рабочих переменных и текущего состояния), которые необходимы для реализации выполняемых потоком функций.

Параллельные архитектуры БД



Параллельные архитектуры БД

Эволюция в области информационных систем все отчетливее направлена в сторону объединения трех видов задач: оперативной обработки транзакций (OLTP), поддержки принятия решений (DSS) и пакетной обработки (batch processing).

Одновременное исполнение задач смешанного характера, разделяющих общие вычислительные ресурсы и базы данных, называется "оперативной сложной обработкой данных" (OLCP - On Line Complex Processing).

Параллельные архитектуры БД

Основные факторы реализации универсальной системы со смешанной загрузкой можно определить как:

1. оптимизация запросов;
2. эффективное управление ресурсами;
3. параллельная обработка запросов.

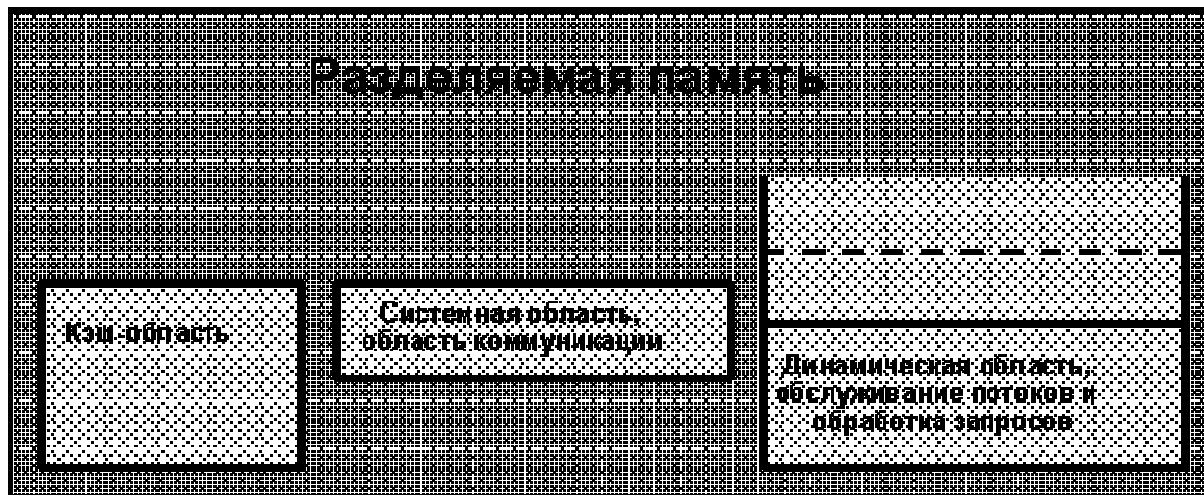
Параллельные архитектуры БД

Оптимальное управление ресурсами требует поддержки прозрачности доступа к ресурсам в целом и эффективного использования каждого ресурса в отдельности.

Один из наиболее важных ресурсов - оперативная память. Оптимальное управление распределением памяти настолько же критично для производительности системы, как и уменьшение загрузки ОС.

Параллельные архитектуры БД

Приложения СУБД используют разделяемую память для сохранения контекста подключения, а также кэширования запросов и результатов.



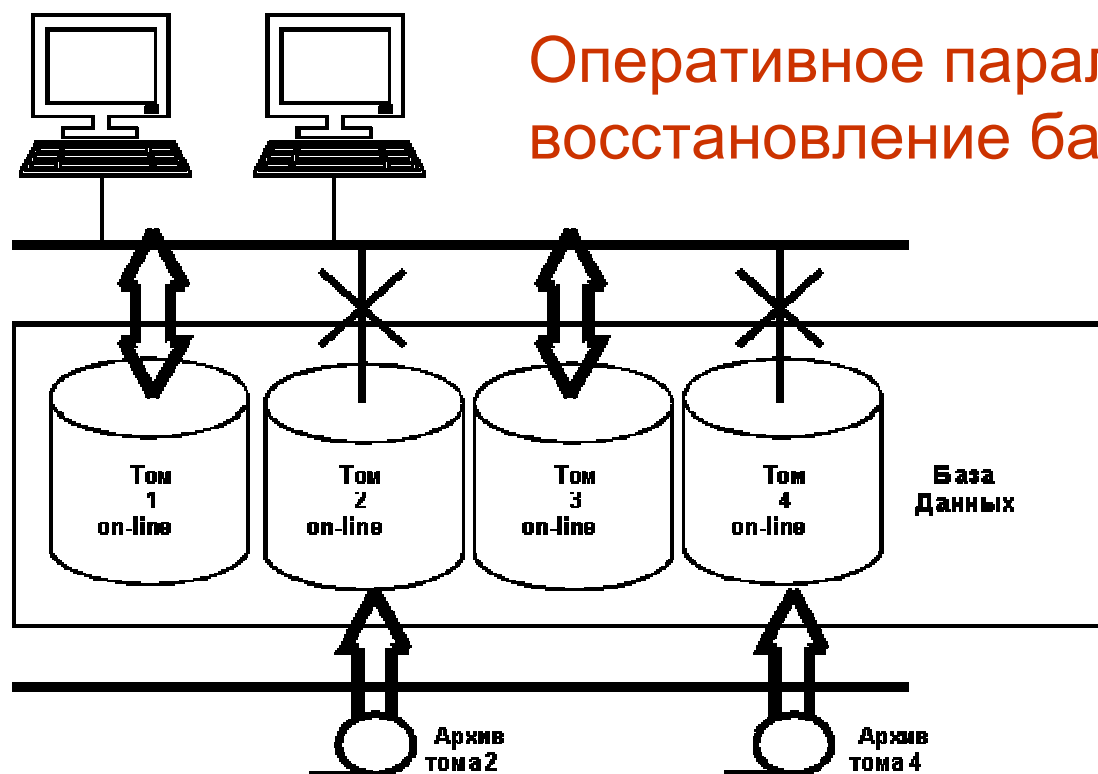
Параллельные архитектуры БД

Основные моменты, обеспечивающие постоянную готовность современных СУБД можно определить, как:

1. оперативное администрирование;
2. функциональная насыщенность СУБД.

Параллельные архитектуры БД

Утилиты администрирования в идеале призваны поддерживать бесперебойное функционирование СУБД, что подразумевает сведение к минимуму планируемых или сбойных простоев системы.



Оперативное параллельное восстановление базы данных.

Параллельные архитектуры БД

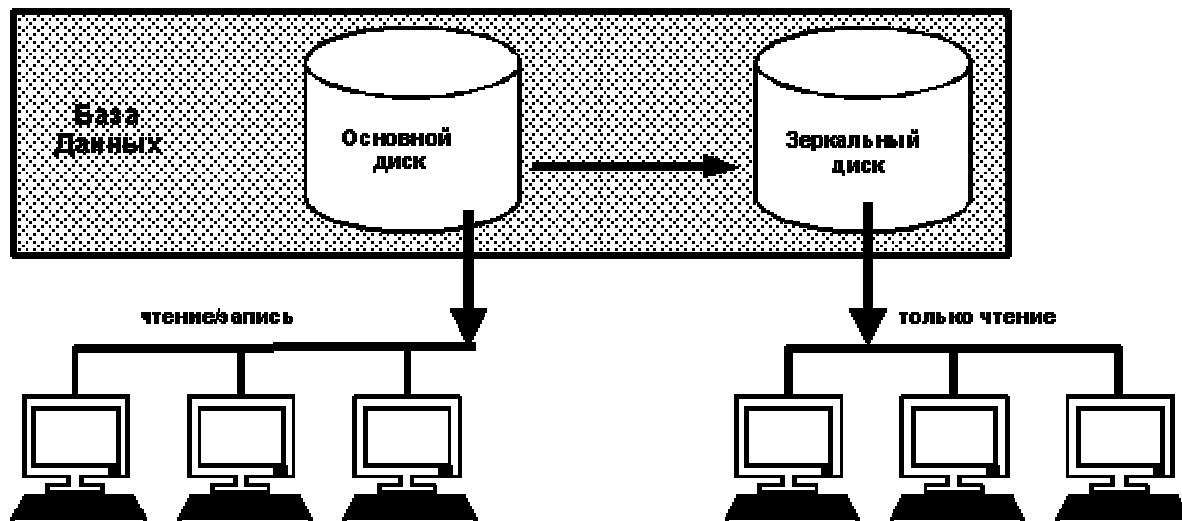
Общим термином "функциональная насыщенность" можно определить множество механизмов, используемых серверами баз данных (а) для уменьшения последствий системных сбоев и (б) для повышения прозрачности доступа к дублированным данным при нормальном функционировании.

Теоретически отказоустойчивая СУБД обязана обеспечивать доступ к данным по чтению и записи независимо от обстоятельств, будь то сбой аппаратной платформы или какого-то компонента СУБД.

Параллельные архитектуры БД

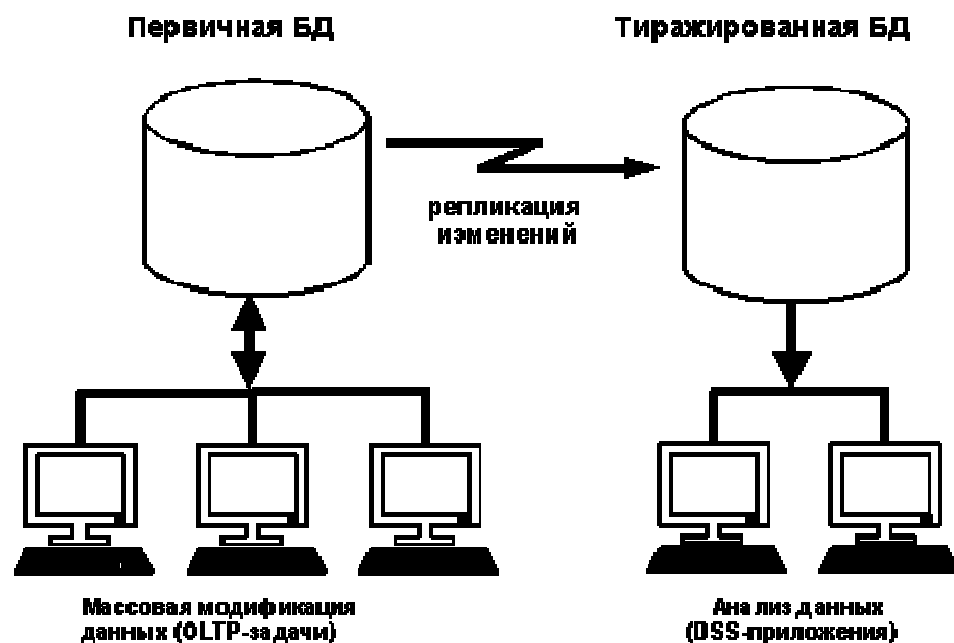
Системы, обеспечивающие непрерывный доступ к данным (fault tolerant) или почти непрерывный (high availability), обычно опираются на различные формы избыточности. Как правило, это системы дублирования аппаратного обеспечения и контролируемой избыточности данных.

Один из вариантов - зеркалирование дисков:



Параллельные архитектуры БД

Тиражирование в системах, требующих в первую очередь повышенной надежности, в целом подобно зеркалированию, но здесь копия данных может поддерживаться удаленно.



ВОПРОСЫ ?

СОВЕТ ДНЯ:

Из «нормативной» лексики студентов:

«Смени свой интерфейс!»

«Я тебя отдисконекчу!»

«Щас дам по монитору – твой дисплей погаснет...»

Люди, изучайте русский язык – это поможет сдать экзамен!