

Архитектура систем баз данных

Основные концепции

Определение архитектуры

Архитектура – это совокупность **существенных** решений, касающихся
- **организации** программной системы;

- выбора **структурных** элементов, составляющих систему и их интерфейсов;

- **поведения** этих элементов, определенного в процессе взаимодействия с другими элементами;

- **составления** из этих структурных и поведенческих элементов прогрессивно больших систем;

- **архитектурного стиля**, определяющего организацию системы, её элементы и их интерфейсы, их взаимодействия и способ соединения.

Классическое определение (Буч, Рембо, Джекотсон)

Виды архитектур баз данных

Централизованная архитектура

Компоненты системы базы данных располагаются в **единой среде** (ПК) и исполняются внутри **одной программы**

Реализует режим работы системы

однопользовательский

с централизованной БД

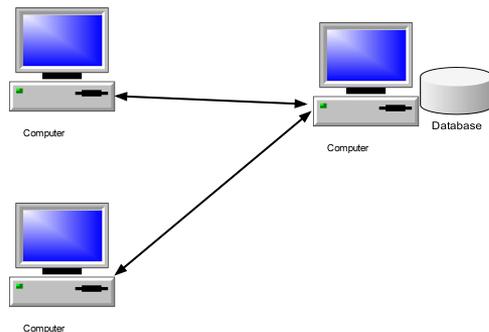
Децентрализованная архитектура

Компоненты системы базы данных располагаются в **разных средах** (ПК) и исполняются внутри **различных программ**

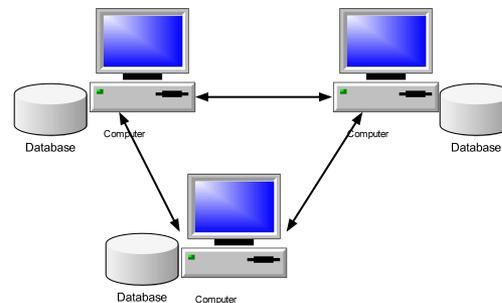
многopользовательский с параллельным доступом к БД

с распределенной БД

(системы с распределенной обработкой)



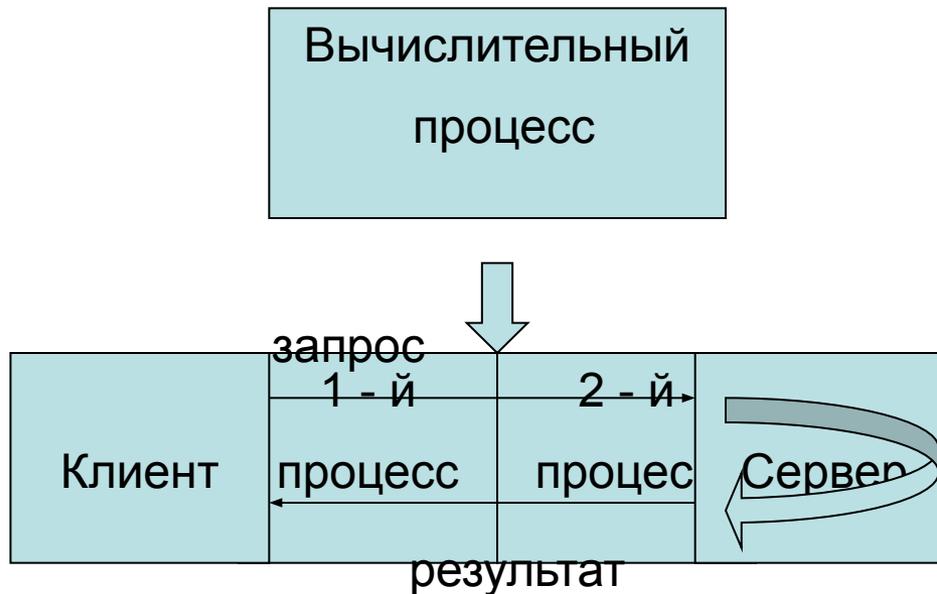
(системы распределенных баз данных)



Технология «клиент-сервер»

Реализация децентрализованной архитектуры основана на технологии «клиент-сервер»

Суть технологии «клиент-сервер»



Технология «клиент-сервер» в БД

Реализация технологии «клиент-сервер» в системах баз данных

состоит в **разделении** функций, выполняемых системой баз данных, и размещении их на **клиентской** либо **серверной** части:

- функции ввода и отображения данных (**Presentation logic**)
- функции решения прикладной задачи (**Business logic**)
- функции реализации модели данных (**Database logic**)
- функции управления ресурсами (**Database Manager system**)
- функции связи (**Service function**)

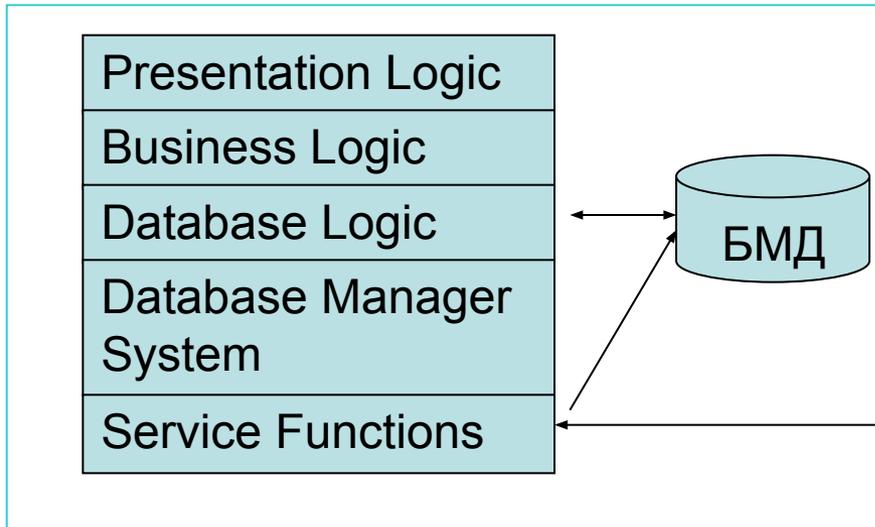
Модели архитектур баз данных

	Presentation logic	Business logic	Database logic	Database manager	
Модель сервера БД и модель активного сервера приложений					клиент
					сервер
Модель сервера БД					клиент
					сервер
Модель удаленного доступа					клиент
					сервер
Модель удаленного доступа					клиент
					сервер
Модель файлового сервера					клиент
					сервер
Модель файлового сервера					клиент
					сервер

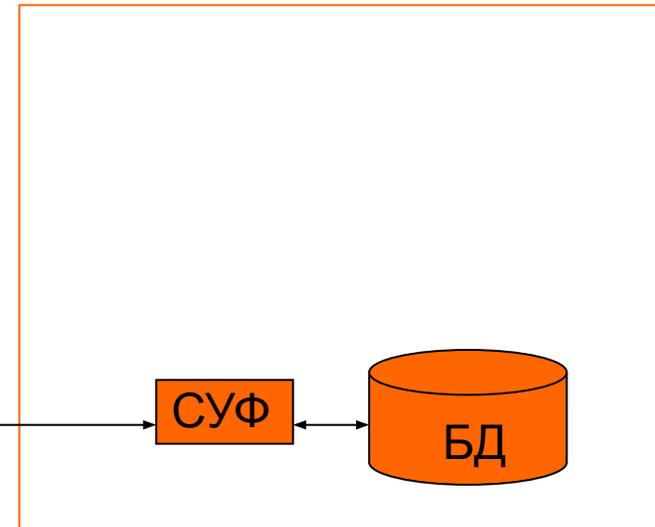
Характеристика моделей

Модель файлового сервера

Клиент



Сервер



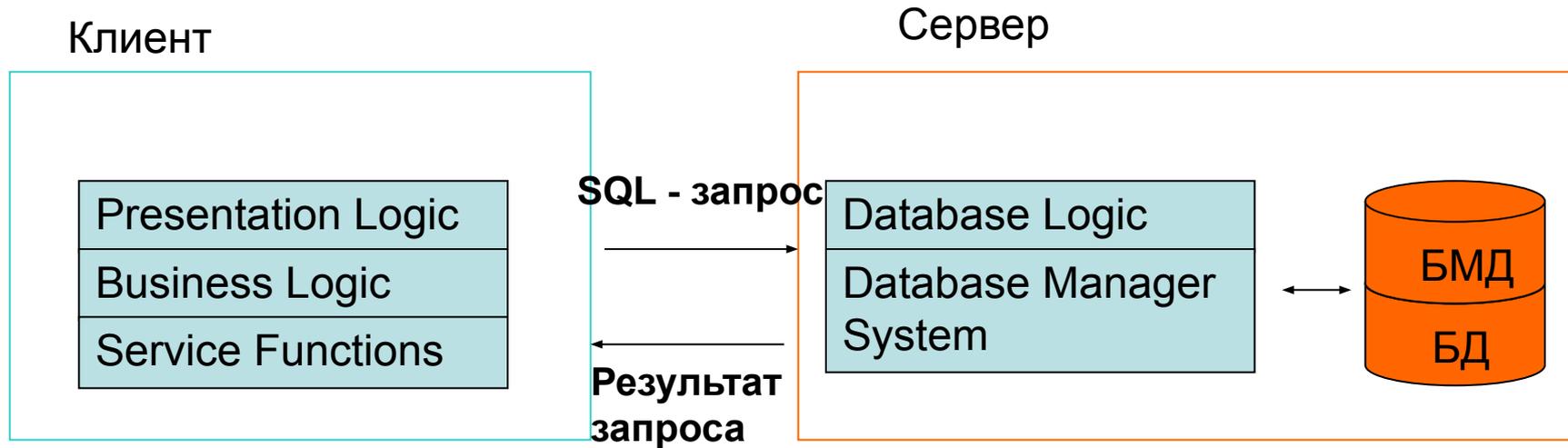
Достоинства: разделение монопольного процесса на 2 процесса

Основные недостатки:

1. Большая нагрузка на сеть;
2. Слабая защита данных.

Характеристика моделей

Модель удаленного доступа к данным



Достоинства:

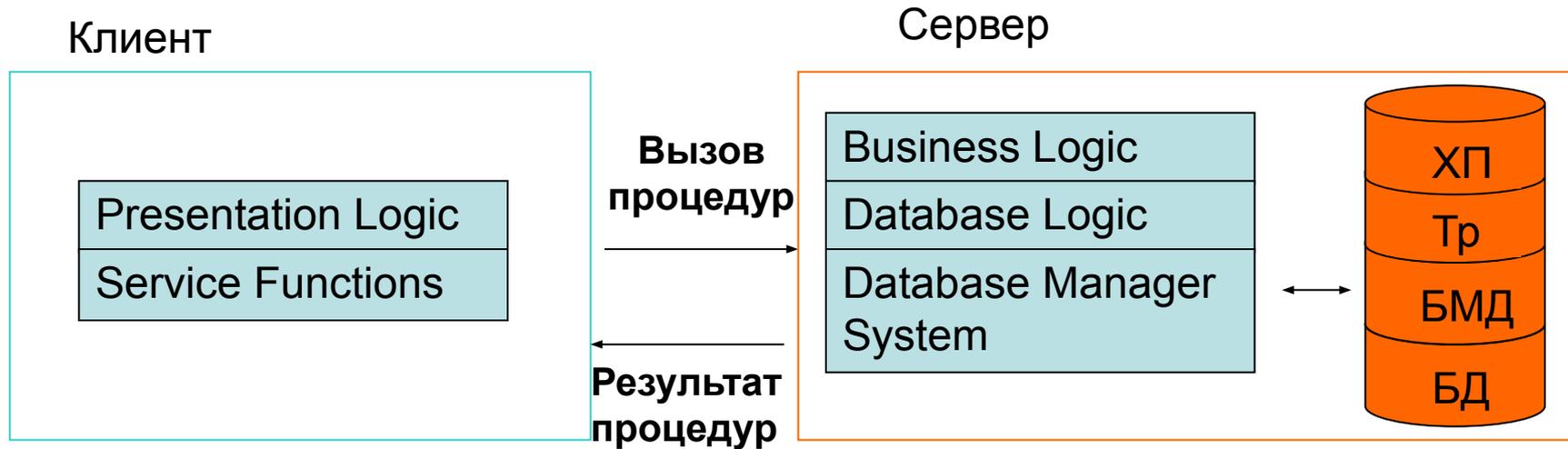
1. Разгрузка сетевого трафика;
2. Повышение производительности

Основные недостатки:

1. Дублирование кода приложения;
2. Пассивная роль сервера.

Характеристика моделей

Модель сервера баз данных



Достоинства:

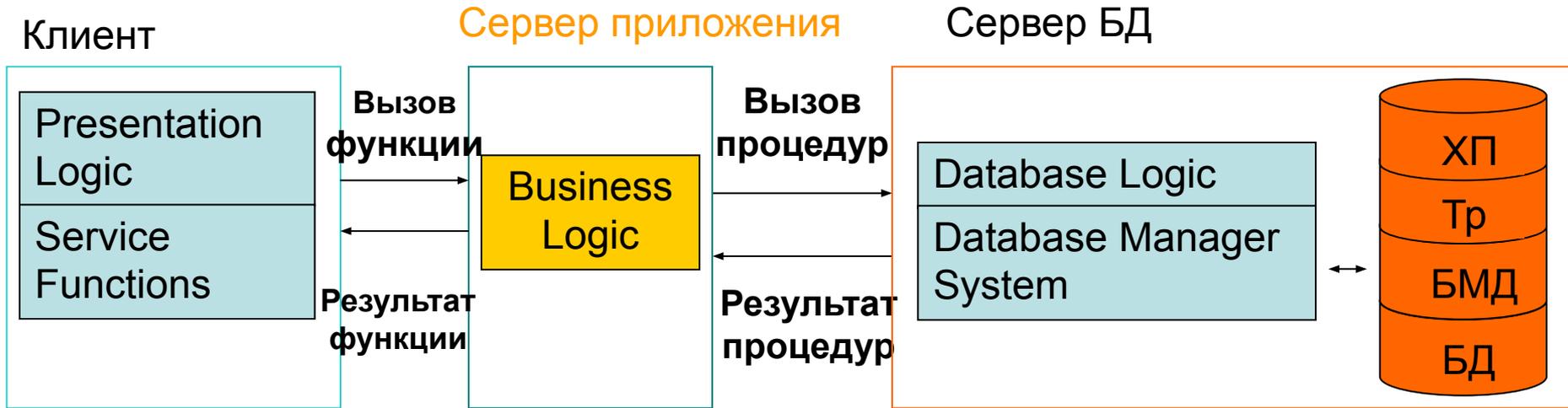
1. Технологичность разработки программного обеспечения;
2. Высокий уровень защиты базы данных из-за применения ХП

Основные недостатки:

1. Большая нагрузка на сервер при большом числе клиентов;
2. Сложности в проектировании и планировании разработки системы

Характеристика моделей

Модель сервера приложений



Достоинства:

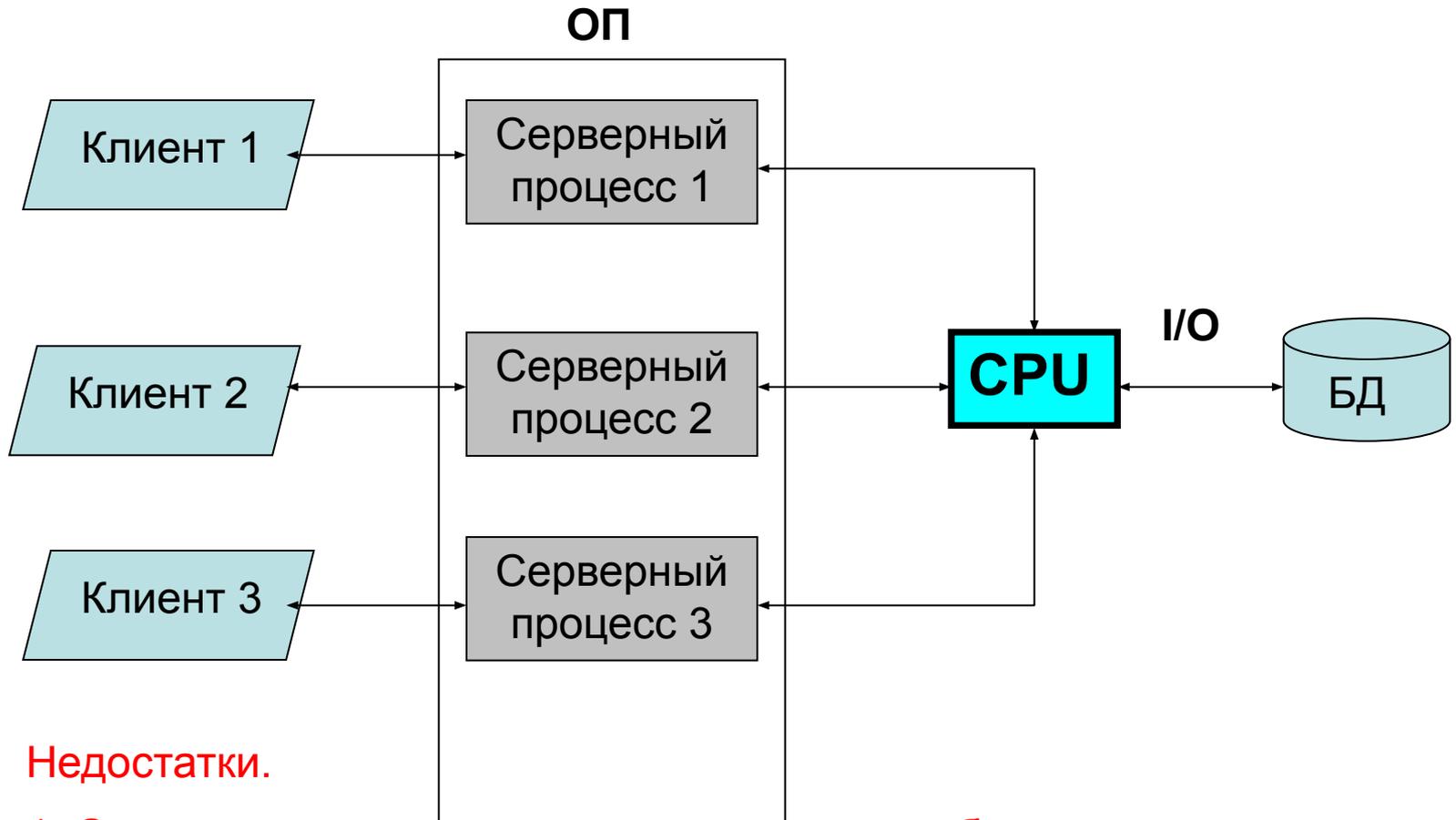
1. Высокая производительность системы базы данных при большого количестве клиентов при невысокой стоимости.
2. Более высокая технологичность разработки и сопровождения программного обеспечения

Основные недостатки:

1. Дополнительные издержки на отладку и тестирование
2. Реализация дополнительных общих функций на сервере приложений

Модели серверов баз данных

Модель один-к-одному

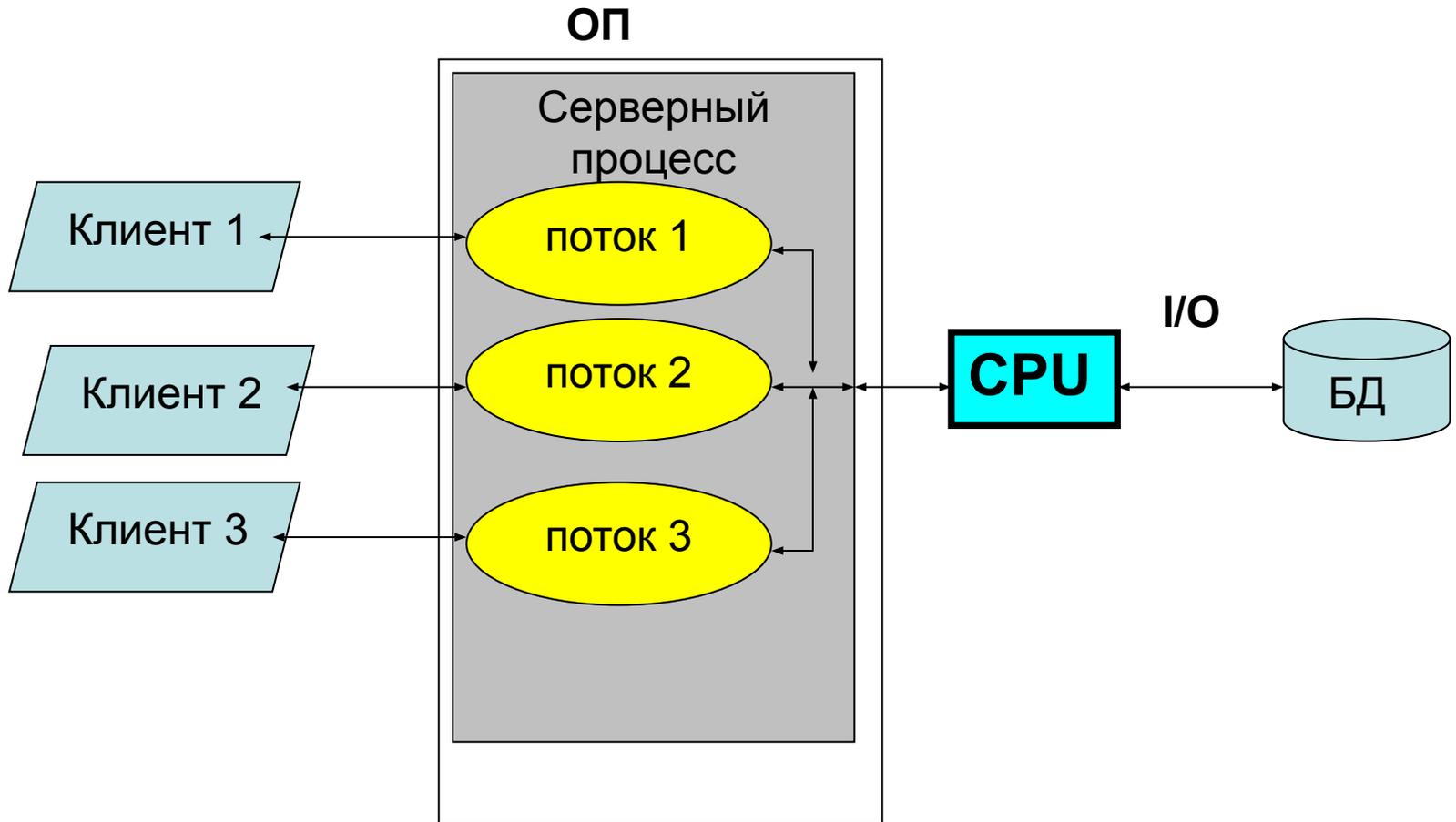


Недостатки.

1. Ограниченное количество одновременно обслуживаемых клиентов.
2. Нерациональное использование ресурсов.

Модели серверов баз данных

Многопоточковая односерверная модель



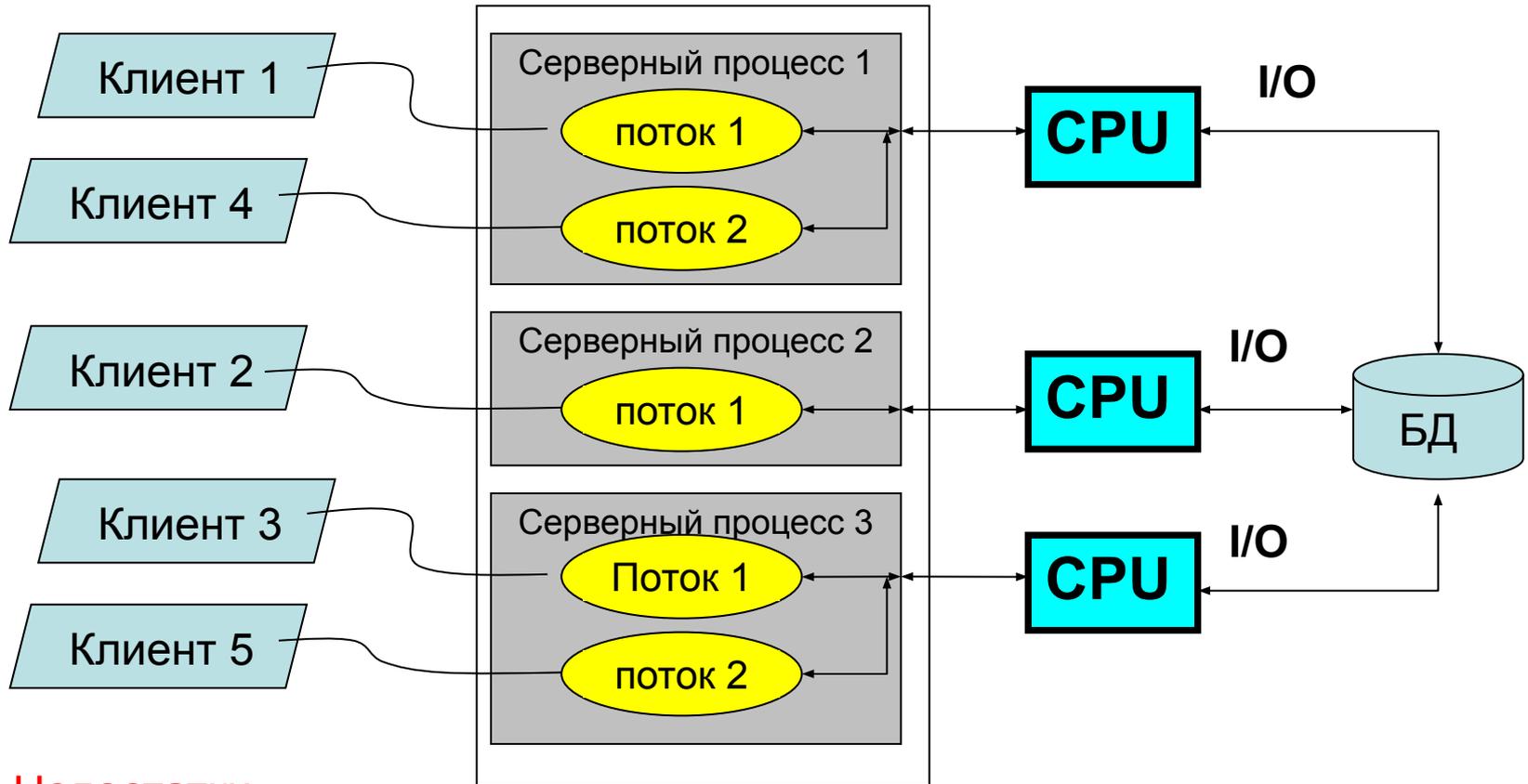
Недостатки.

1. Модель не эффективна для многопроцессорной системы.

Модели серверов баз данных

Многопоточковая мультисерверная модель

ОП



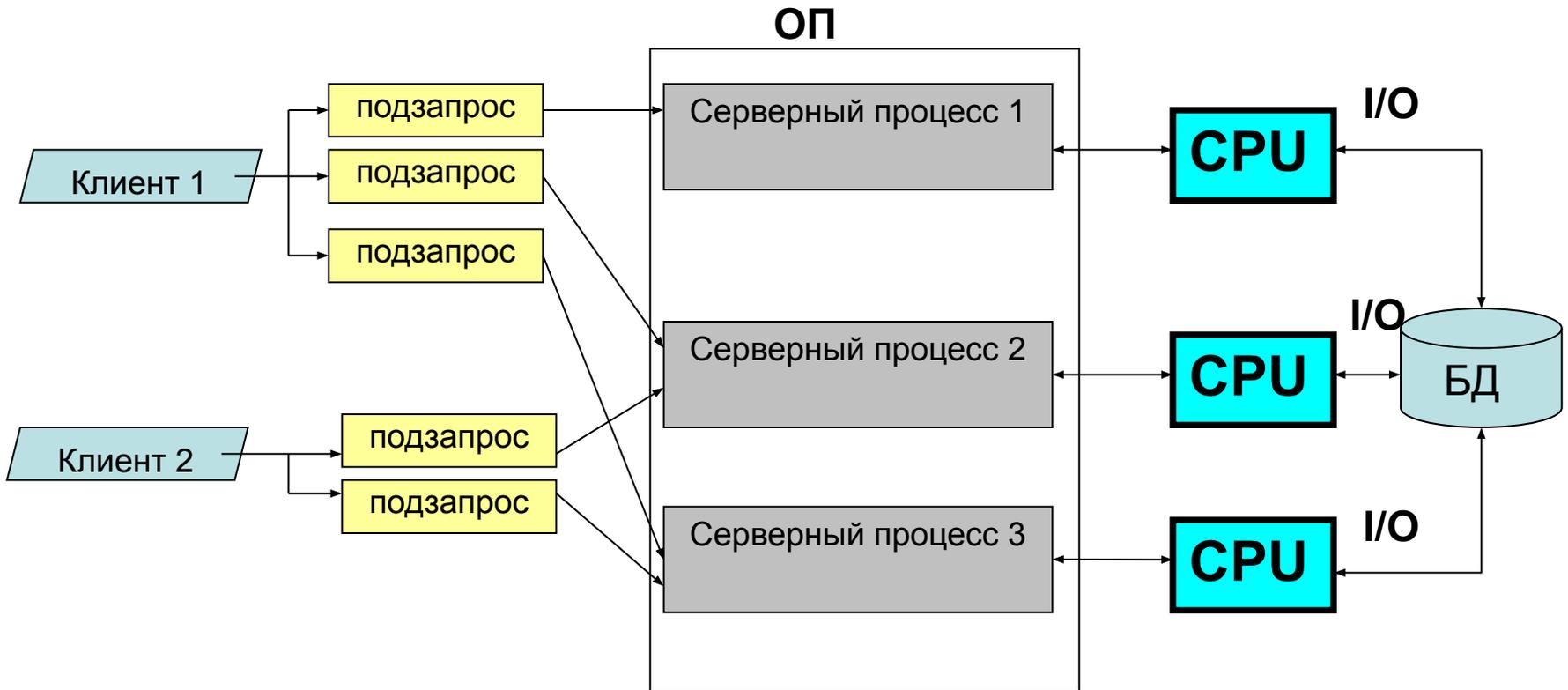
Недостатки.

1. Снижение производительности при выполнении операций I/O

На базе этой модели строятся архитектуры с распараллеливанием выполнения запросов

Модели серверов баз данных

Выполнение запроса при вертикальном параллелизме



Декомпозиция запроса выполняется при условии независимости полученных подзапросов. Тип серверного процесса здесь – нить (thread).

Модели серверов баз данных

Выполнение запроса при гибридном параллелизме

