

# Распределенные вычисления

Косяков Михаил Сергеевич

к.т.н., доцент кафедры ВТ

Кузичкина Анастасия Олеговна

Григорьев Александр Николаевич

инженеры ORC Group

[http://vk.com/distributedclass\\_2016\\_1](http://vk.com/distributedclass_2016_1)

[ifmo.distributedclass.bot@gmail.com](mailto:ifmo.distributedclass.bot@gmail.com)



# Мотивация



# О чем речь?

- RAM model vs Concurrency model
- Share everything vs Share nothing

## Разделы дисциплины

- Предмет распределенных вычислений
- Модель распределенного вычисления
- Логическое время
- Алгоритмы взаимного исключения

# Литература

М.С. Косяков

ВВЕДЕНИЕ В РАСПРЕДЕЛЕННЫЕ  
ВЫЧИСЛЕНИЯ

Учебное пособие



Санкт-Петербург  
2014

## Введение в распределенные вычисления. Учебное пособие.

Косяков М.С. СПб: НИУ ИТМО, 2014 – 155с.

### ... и литература из списка в пособии

# Раздел 1. Предмет распределенных вычислений



# Определения распределенной системы

«Вы понимаете, что пользуетесь распределенной системой, когда поломка компьютера, о существовании которого вы даже не подозревали, приводит к останову всей системы, а для вас – к невозможности выполнить свою работу»

*Лесли Лэмпорт (Leslie Lamport)*



# Определения распределенной системы

«Распределенная система – набор независимых компьютеров, представляющий их пользователям единой объединенной системой»

*Эндрю Таненбаум (Andrew Tanenbaum)*





# Определения распределенной системы


- С аппаратной точки зрения:

Совокупность взаимосвязанных автономных компьютеров или процессоров

- С программной точки зрения:

Совокупность независимых процессов, взаимодействующих посредством передачи сообщений для обмена данными и координации своих действий

# Автономные узлы и независимые процессы

- Автономные узлы = независимое управление = точно не SIMD архитектура
- Каждый процесс имеет свое собственное состояние
- Состояние процесса закрыто для других процессов
- Скорости выполнения процессов различны и неизвестны
- Задержки обмена сообщениями случайны и непредсказуемы: точность координации действий процессов ограничена этими задержками 

# Точность координации действий процессов: масштаб бедствия

Table 2.2 Example Time Scale of System Latencies

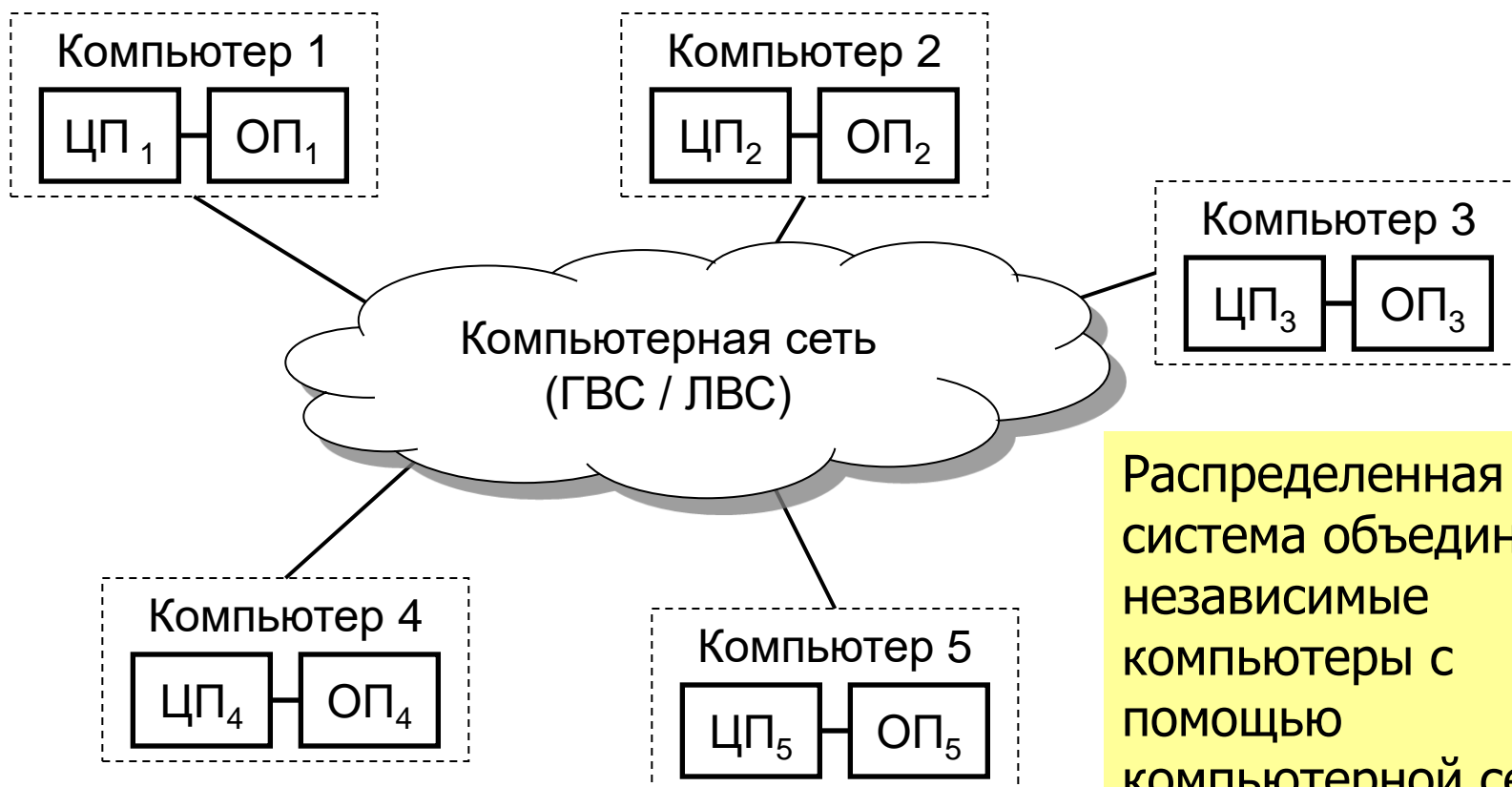
Event	Latency	Scaled
1 CPU cycle	0.3 ns	1 s
Level 1 cache access	0.9 ns	3 s
Level 2 cache access	2.8 ns	9 s
Level 3 cache access	12.9 ns	43 s
Main memory access (DRAM, from CPU)	120 ns	6 min
Solid-state disk I/O (flash memory)	50–150 µs	2–6 days
Rotational disk I/O	1–10 ms	1–12 months
Internet: San Francisco to New York	40 ms	4 years
Internet: San Francisco to United Kingdom	81 ms	8 years
Internet: San Francisco to Australia	183 ms	19 years
TCP packet retransmit	1–3 s	105–317 years
OS virtualization system reboot	4 s	423 years
SCSI command time-out	30 s	3 millennia
Hardware (HW) virtualization system reboot	40 s	4 millennia
Physical system reboot	5 m	32 millennia



# Признаки распределенной системы

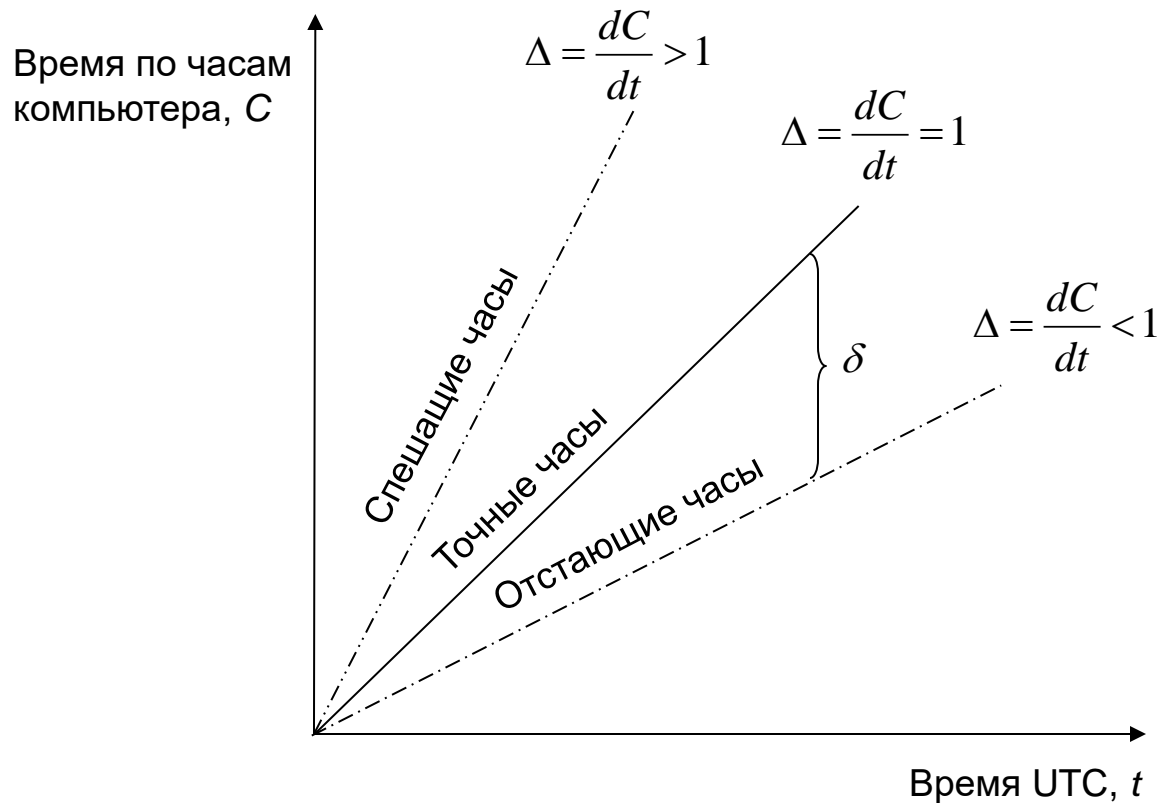
- Отсутствие общей памяти
- Отсутствие единого времени
- Географическое распределение
- Независимость и гетерогенность

# Определения распределенной системы



Распределенная система объединяет независимые компьютеры с помощью компьютерной сети

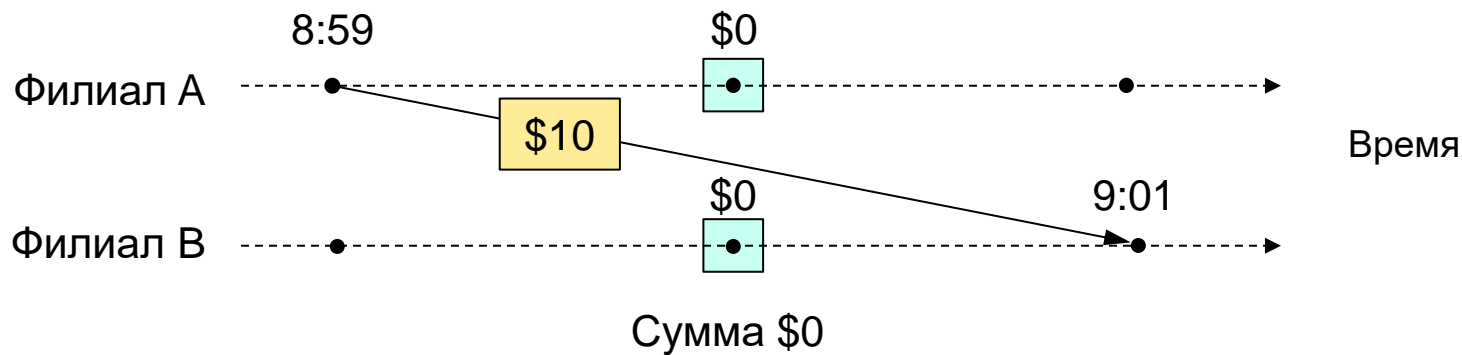
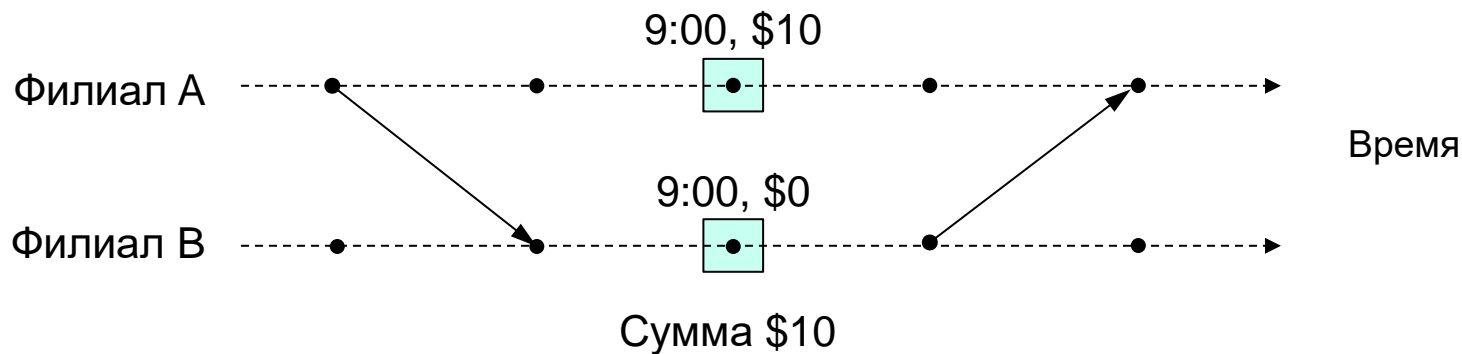
# Физическое время: Часы в независимых компьютерах



Соотношение времени по часам компьютеров и времени UTC;  
 **$\delta$**  – рассинхронизация часов,  **$\Delta$**  – скорость дрейфа

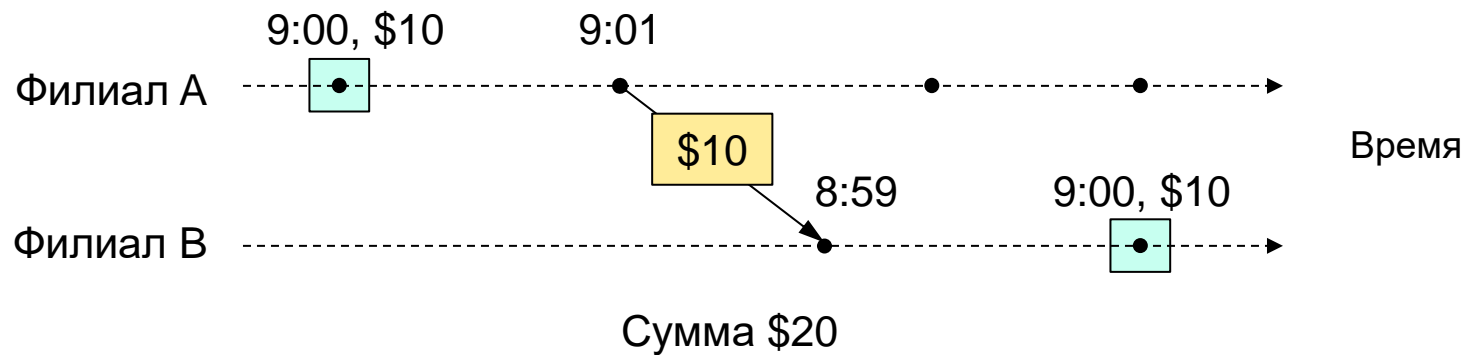
- С течением времени показания часов станут существенно отличаться
- Приложения ожидающие, что временная отметка, ассоциируемая с тем или иным событием, корректна и не зависит от компьютера, на которой она регистрировалась (то есть часы которого использовались), могут работать неправильно

# Банковская система






# Банковская система



# Синхронные распределенные системы

- Время выполнения каждого отдельного действия любого процесса ограничено снизу и сверху известными значениями
- Задержка доставки каждого сообщения от одного процесса к другому не превышает известный предел
- Каждый процесс имеет свои локальные часы со скоростью отклонения от точных показаний, не превышающей известное значение
- *Нужно гарантировать эти пределы!* 

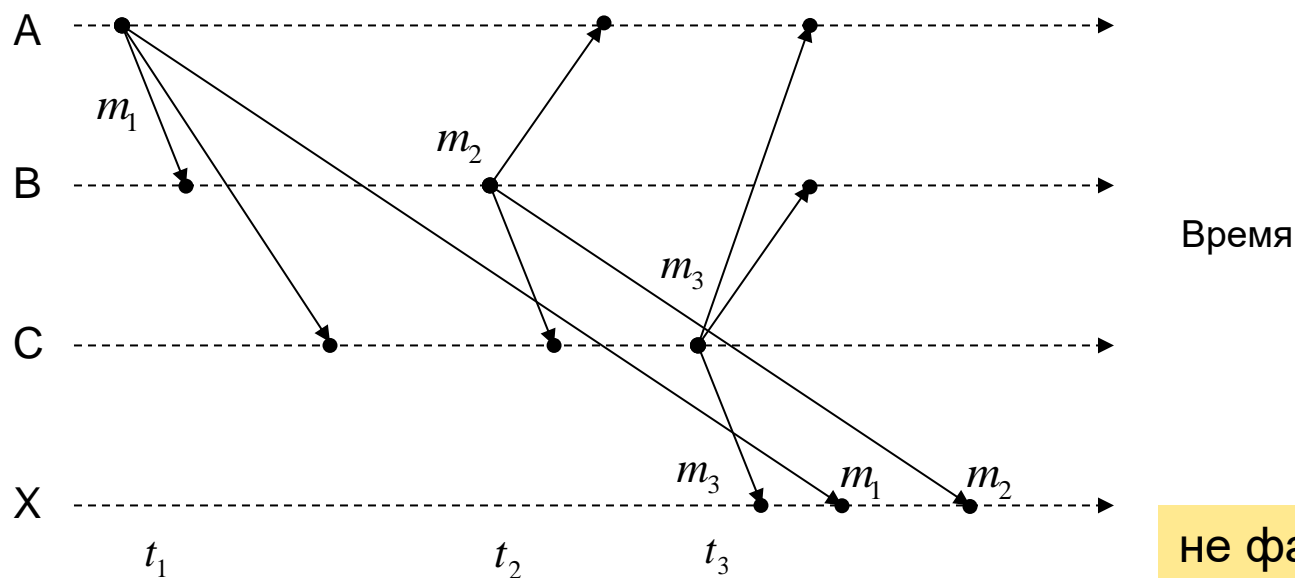
# Асинхронные распределенные системы

- Скорость выполнения операций произвольна
- Задержка доставки сообщений произвольна
- Скорость отклонения часов произвольна
- *Причина асинхронности – в совместном использовании аппаратных ресурсов*
- *Множество проблем не имеет решения для этой модели!*



# Упорядочивание событий

Часто важно знать не время событий, а их порядок: до или после



Порядок сообщений, наблюдаемый различными пользователями, может быть различным



## Восемь заблуждений П. Дейча

«По существу, каждый, кто впервые создает распределенное приложение, делает следующие предположения. Все они, в конце концов, оказываются ложными, и все вызывают большие неприятности. Вот эти восемь заблуждений:

1. Сеть является надежной
2. Задержки передачи сообщений равны нулю
3. Полоса пропускания неограничена
4. Сеть является безопасной
5. Сетевая топология неизменна
6. Систему обслуживает только один администратор
7. Издержки на транспортную инфраструктуру равны нулю
8. Сеть является однородной»



# Централизованные vs распределенные алгоритмы

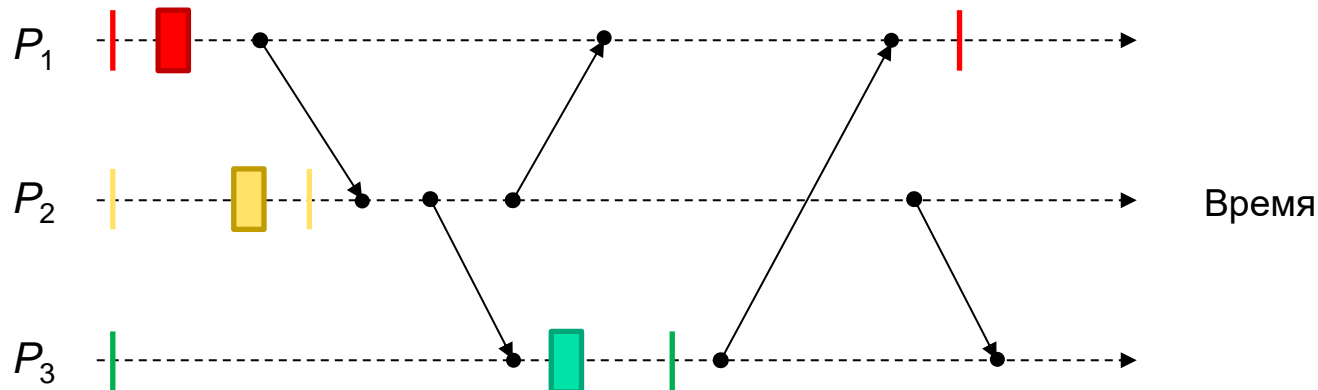
- Отсутствие знания глобального состояния
  - Доступ только к локальной информации
  - Информация о состоянии других процессов – только через поступающие сообщения, как следствие – устаревшая
  - «Астрономические» проблемы
- Отсутствие общего единого времени
  - События НЕ полностью упорядочены

# Централизованные vs распределенные алгоритмы

- Отсутствие детерминизма
  - Независимое исполнение процессов и случайные задержки передачи сообщений
  - Выполнение может быть описано разными последовательностями глобальных состояний
  - Следствие: нет смысла говорить, что то или иное состояние достигается по ходу выполнения распределенного алгоритма
- Устойчивость к отказам

# Отсутствие детерминизма

Вопрос: могут ли процессы определить, какие фонари светили одновременно?



Точно известно, что красный и зеленый фонари одновременно находились во включенном состоянии;

Был ли включен красный свет одновременно с желтым процессы не смогут определить





## Масштабируемость РС

Способность вычислительной системы эффективно справляться с увеличением числа пользователей или поддерживаемых ресурсов без потери производительности и без увеличения административной нагрузки на ее управление

# Масштабируемость РС

- Масштабируемость:
  - Нагрузочная масштабируемость (вертикальная, горизонтальная)
  - Географическая масштабируемость
  - Административная масштабируемость
- Сложности масштабирования:
  - Централизованные службы
  - Централизованные данные
  - Централизованные алгоритмы

- Частое использование синхронной связи при быстром взаимодействии
- Предположение высоконадежной локальной связи с возможностями широковещательных сообщений

# Технологии масштабирования

Цель – уменьшить нагрузку на каждый компонент РС (узлы и связи)

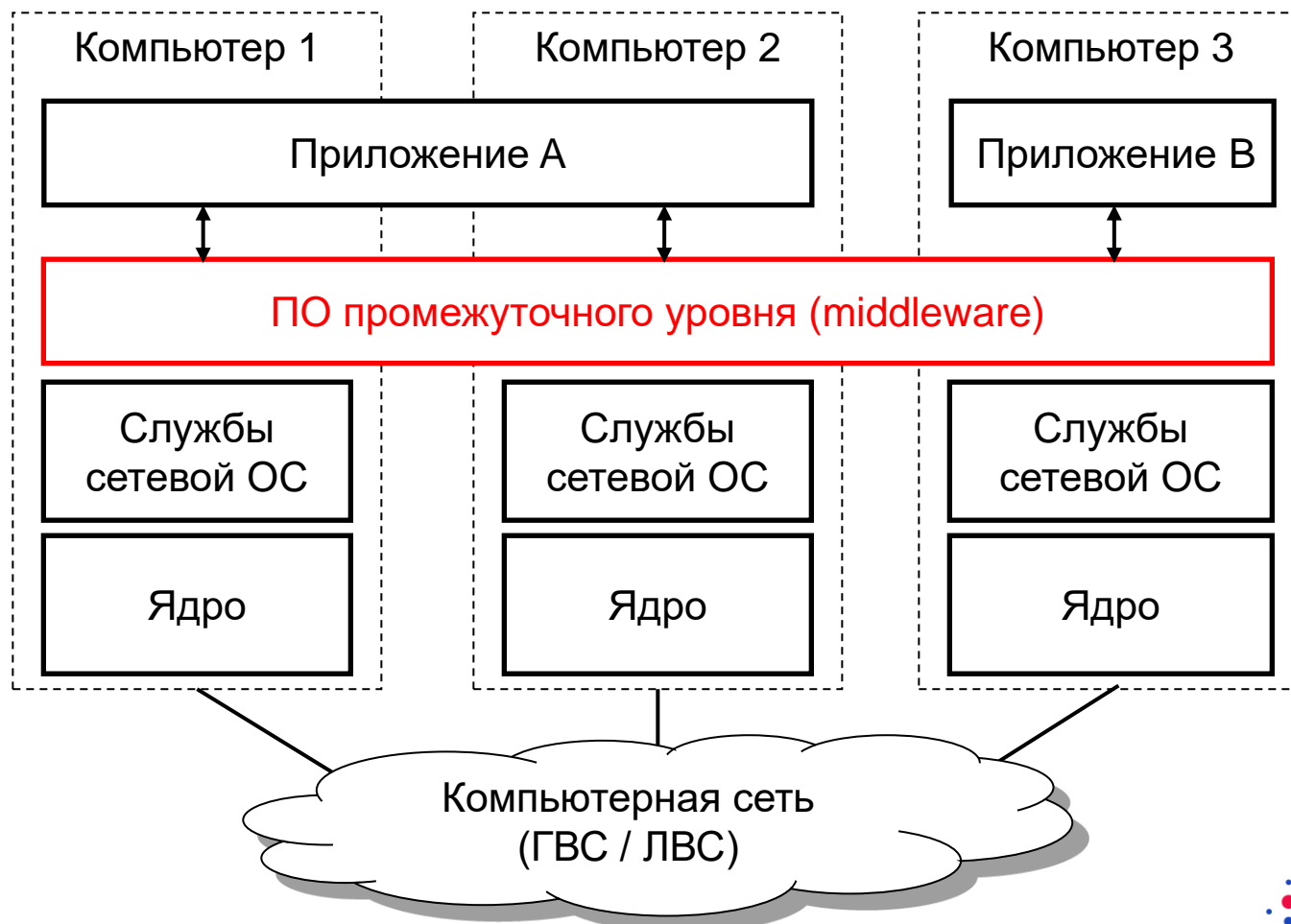
- Распространение (distribution)
- Репликация (replication)
- Кэширование (caching) – снижает нагрузку и скрывает задержки обращения
- Но (!) обеспечение непротиворечивости (consistency) ограничивает масштабируемость
- Асинхронная связь – нельзя для интерактивных приложений
- Перенос кода (апплеты Java, JavaScript)



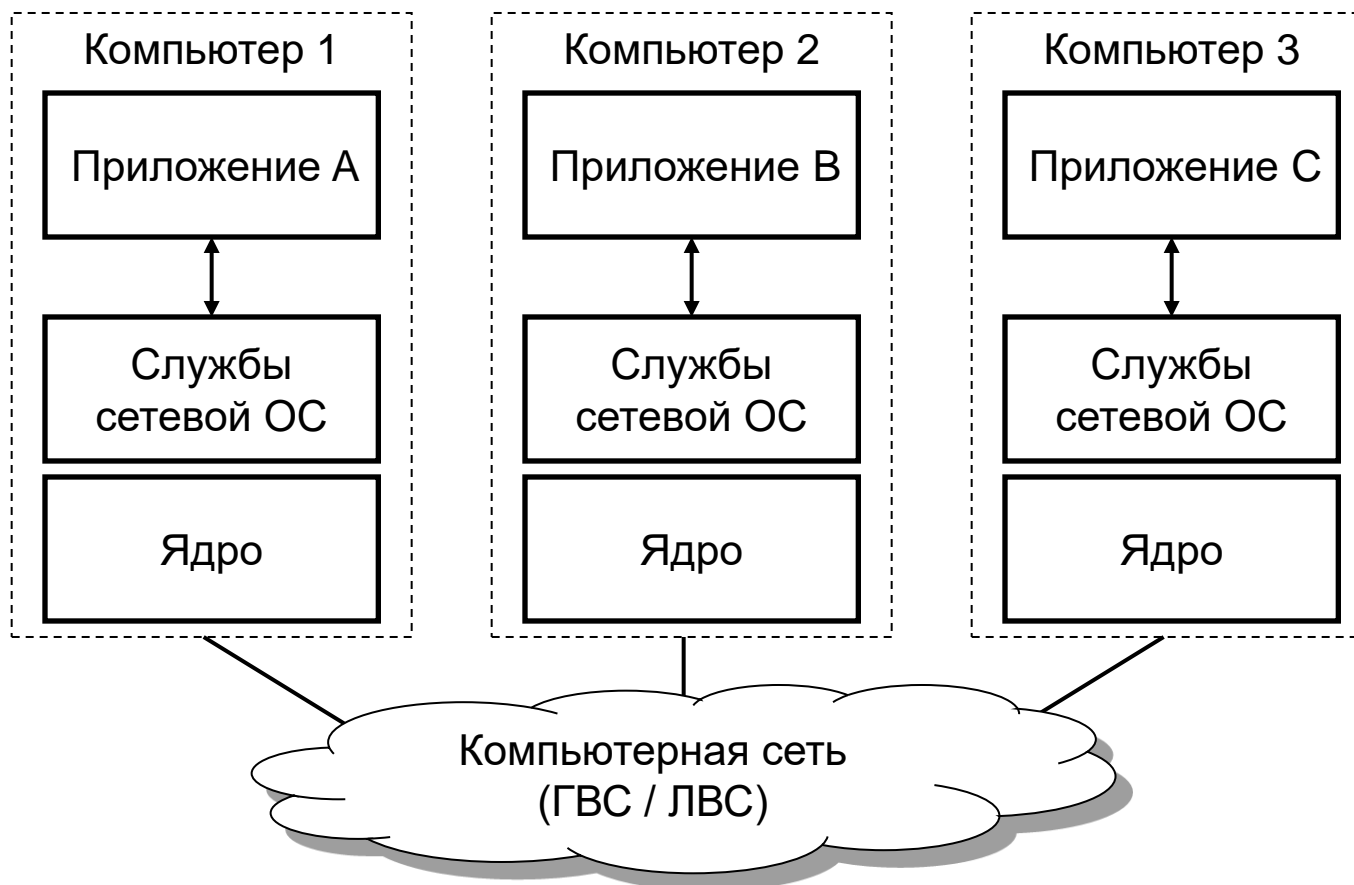
# ПО промежуточного уровня (middleware)

Логический программный уровень, призванный обеспечивать дополнительное абстрагирование приложений от базовых платформ и скрывать их неоднородность от пользователей и приложений, а также предоставлять подходящую модель программирования для разработчиков

# ПО промежуточного уровня (middleware)



# Сетевые ОС



# ПО промежуточного уровня (middleware)

- Интерфейс программирования транспортного уровня сетевой ОС полностью заменяется другими средствами
- Поддерживаемые абстракции определяют соответствующую модель (систему) программирования
- Множество служб, чтобы не пользоваться `send()` и `receive()`