ITIVITI

LET IT TRADE

Распределенные вычисления

Косяков Михаил Сергеевич
к.т.н., доцент кафедры ВТ

Тараканов Денис Сергеевич
ассистент кафедры ВТ

http://vk.com/distributedclass_2019_2

ifmo.distributedclass.bot@gmail.com

УНИВЕРСИТЕТ ИТМО

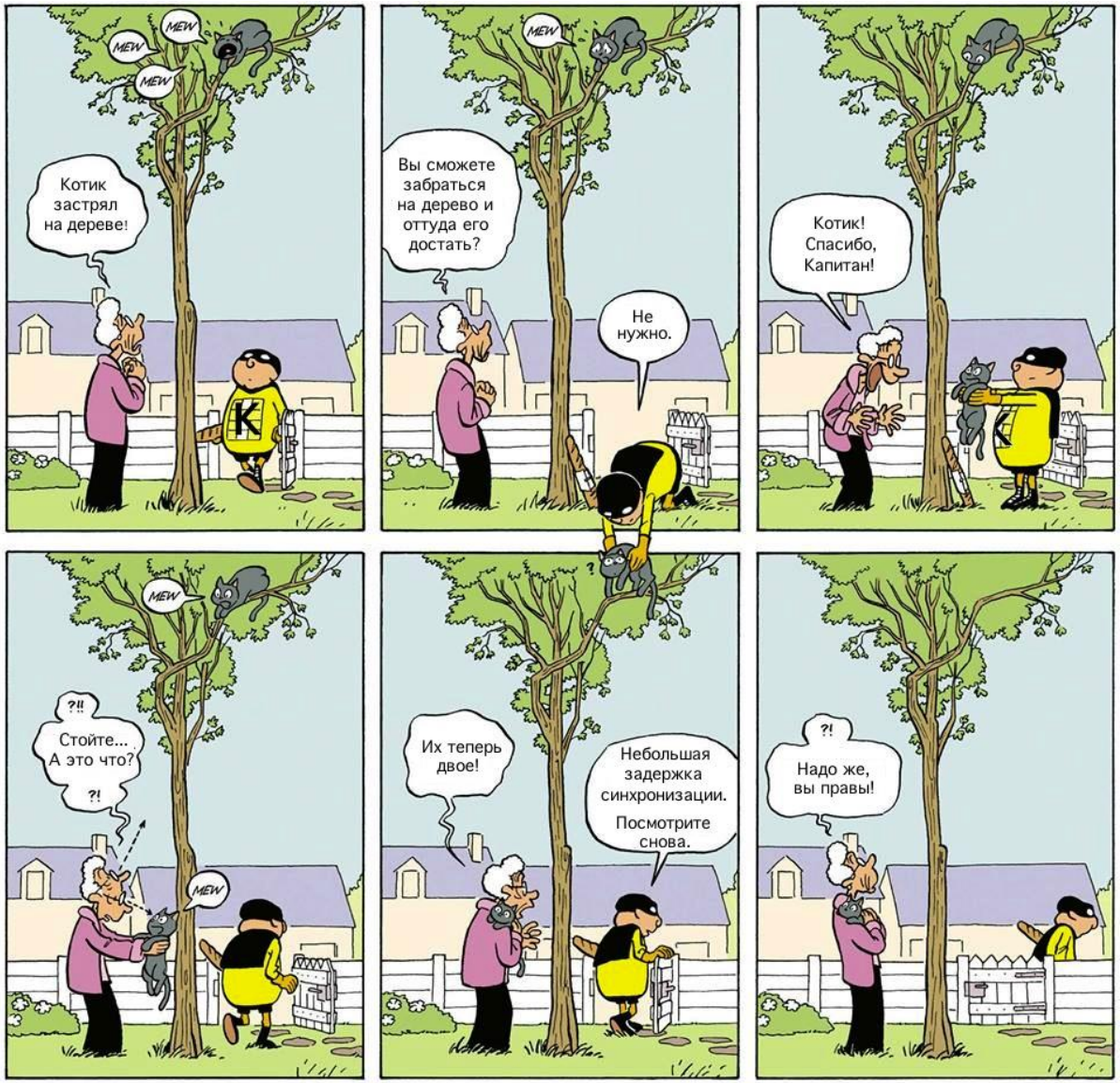
Мотивация

LET IT TRADE

ITIVITI

УНИВЕРСИТЕТ ИТМО

О чем речь?



ITIVITI

LET IT TRADE

УНИВЕРСИТЕТ ИТМО

О чем речь?

ITIVITI

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

LET IT TRADE

УНИВЕРСИТЕТ ИТМО

Синхронизация

ITIVITI

- Что такое синхронизация?
  - Отношение между событиями (до, во время, после)
- Требования к синхронизации
  - Сериализация: A должно произойти до B
  - Взаимное исключение: A и B не должны происходить одновременно
- Модель выполнения
  - Кто обедал раньше Вася или Петя?
  - Отсутствие единого времени

LET IT TRADE

УНИВЕРСИТЕТ ИТМО

Сериализация с помощью сообщений

ITIVITI

Вася (Thread A)	Петя (Thread B)
1. Завтракает 2. Работает 3. Обедает 4. Звонит Пете	1. Завтракает 2. Ждет звонка 3. Обедает

Утверждения:

- $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4$; $b_1 \rightarrow b_2 \rightarrow b_3$
- $a_3 \rightarrow a_4 \rightarrow b_2 \rightarrow b_3$
- $a_1 \ || \ b_1$

LET IT TRADE

УНИВЕРСИТЕТ ИТМО

Взаимное исключение

- Возможные исходы? `count=0;`
- 2 потока исполнения выполняют код:

```
count++;
```

- 10 потоков исполнения выполняют код:

```
count++;
```

- 10 потоков исполнения выполняют код:

```
for (i=0; i<100; ++i) count++;
```

- Взаимное исключение с помощью сообщений

Разделы дисциплины

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

Литература

Введение в распределенные вычисления. Учебное пособие.

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

… и литература из списка в пособии

ITIVITI

Раздел 1. Предмет распределенных вычислений

LET IT TRADE

УНИВЕРСИТЕТ ИТМО

Определения распределенной системы

ITIVITI

LET IT TRADE

«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)

УНИВЕРСИТЕТ ИТМО

Определения распределенной системы

ITIVITI

LET IT TRADE

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

Эндрю Таненбаум (Andrew Tanenbaum)

УНИВЕРСИТЕТ ИТМО

Определения распределенной системы

ITIVITI

- 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

LET IT TRADE

УНИВЕРСИТЕТ ИТМО

Автономные узлы и независимые процессы

ITIVITI

- 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

LET IT TRADE

Точность координации процессов: масштаб бедствия

ITIVITI

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

LET IT TRADE

УНИВЕРСИТЕТ ИТМО

Признаки распределенной системы

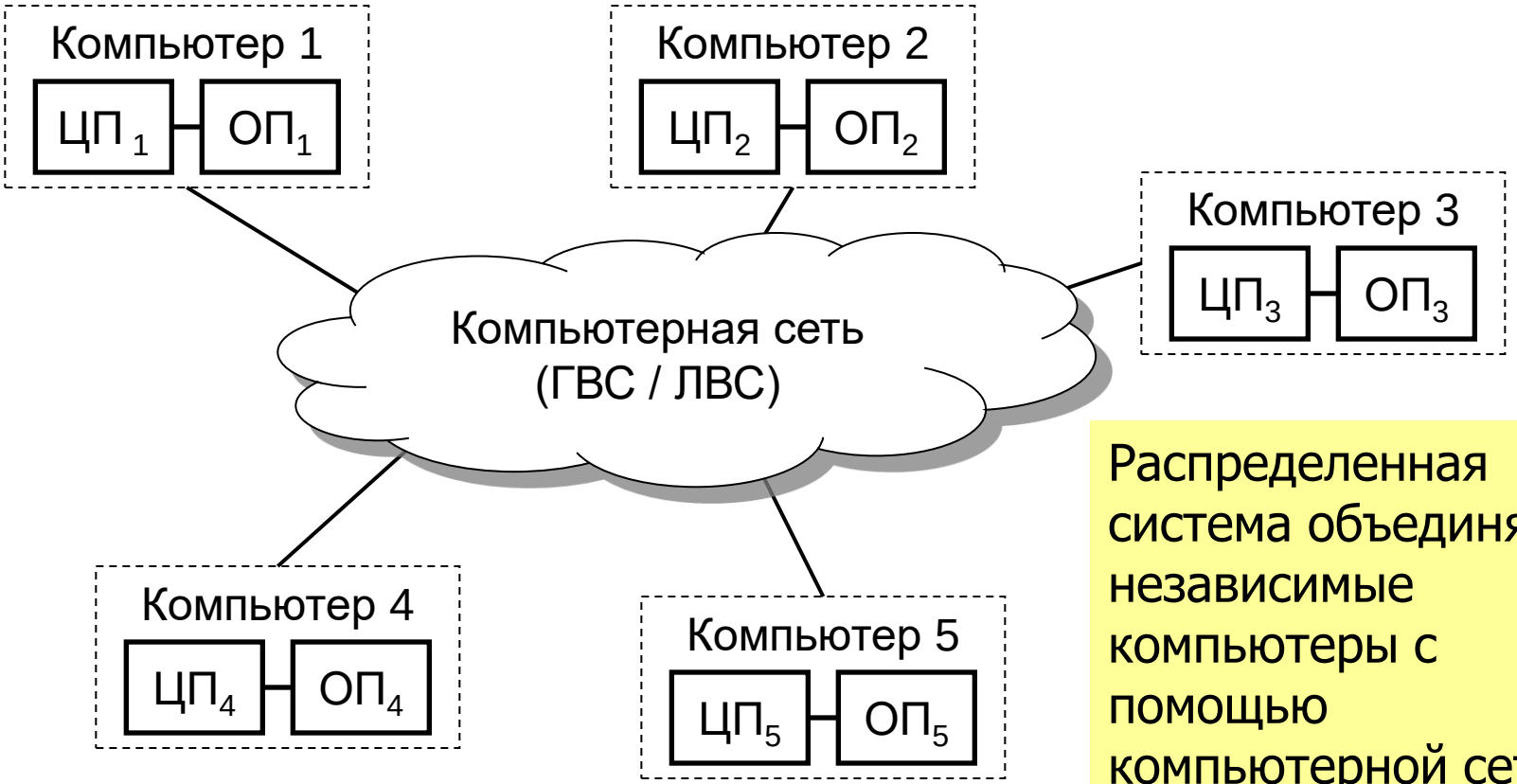
ITIVITI

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

LET IT TRADE

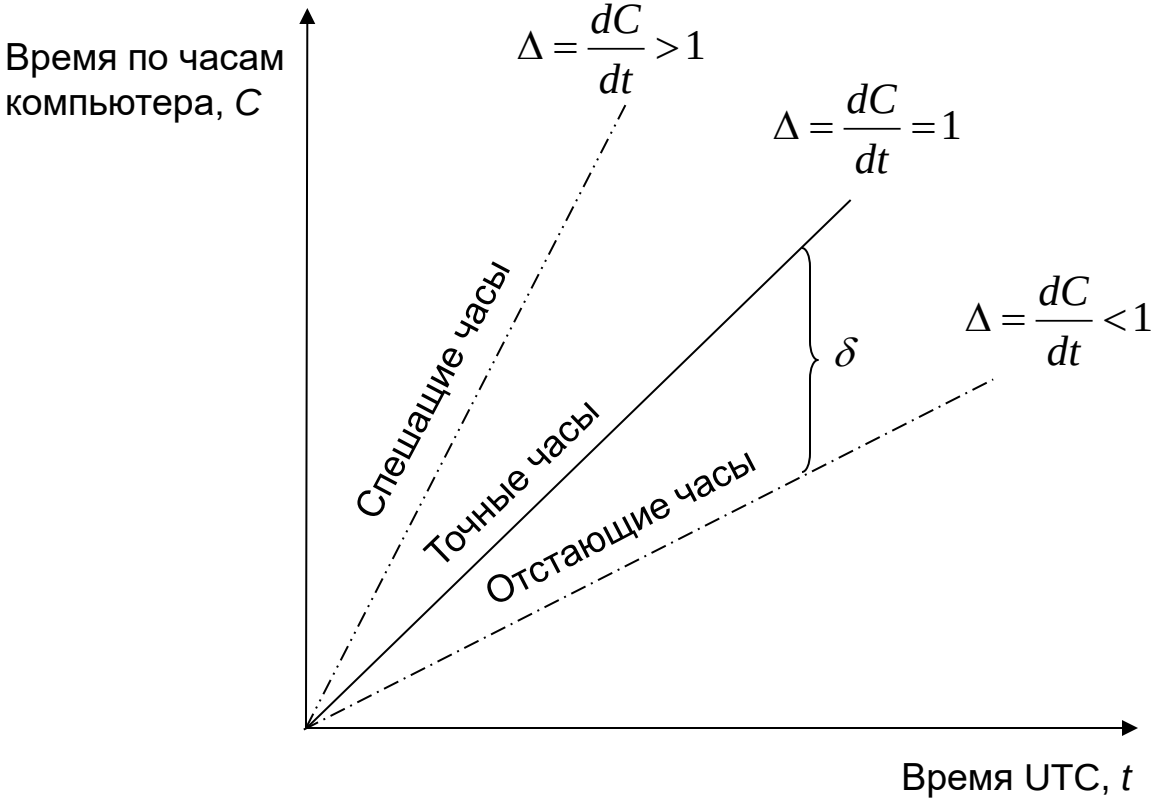
УНИВЕРСИТЕТ ИТМО

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;
δ – рассинхронизация часов, Δ – скорость дрейфа

Физическое время: Часы в независимых компьютерах

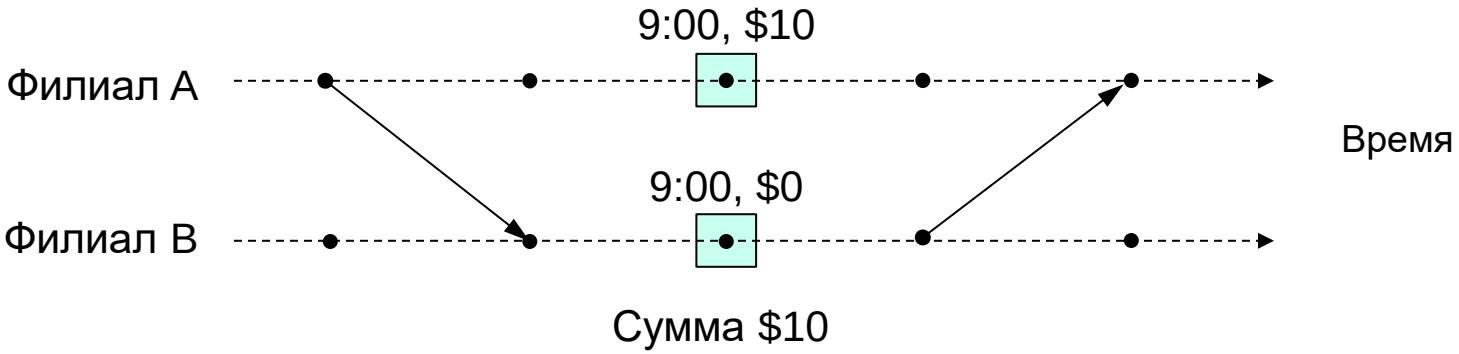
ITIVITI

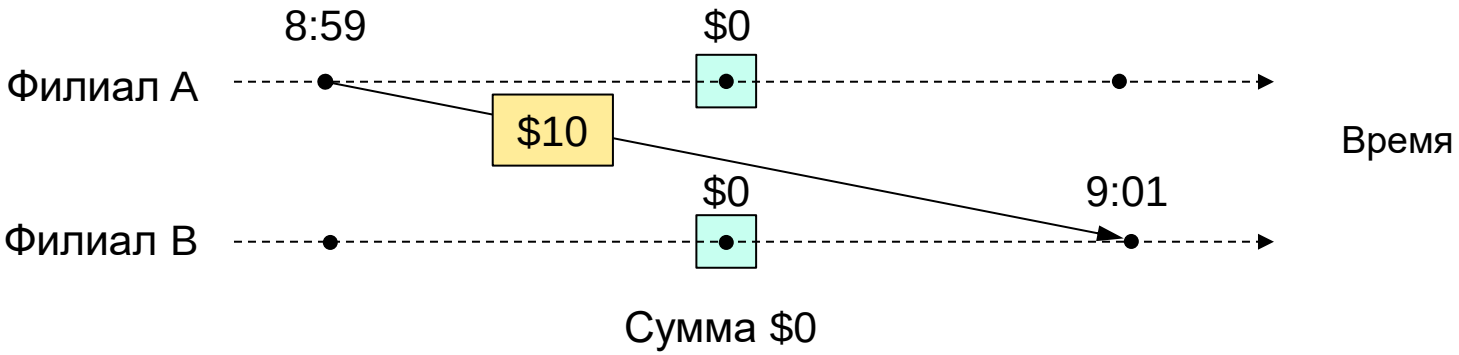
- С течением времени показания часов станут существенно отличаться
- Приложения ожидающие, что временная отметка, ассоциируемая с тем или иным событием, корректна и не зависит от компьютера, на которой она регистрировалась (то есть часы которого использовались), могут работать неправильно

LET IT TRADE

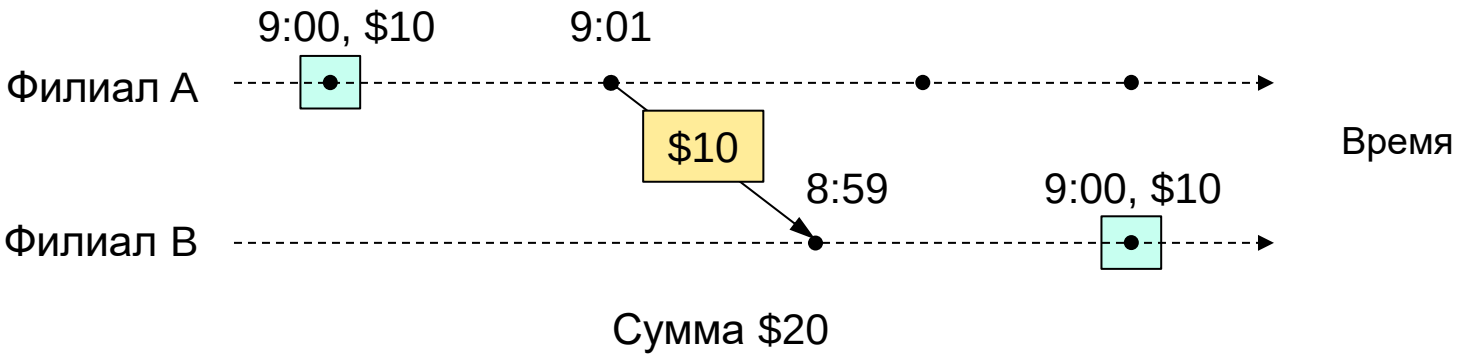
УНИВЕРСИТЕТ ИТМО

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!*

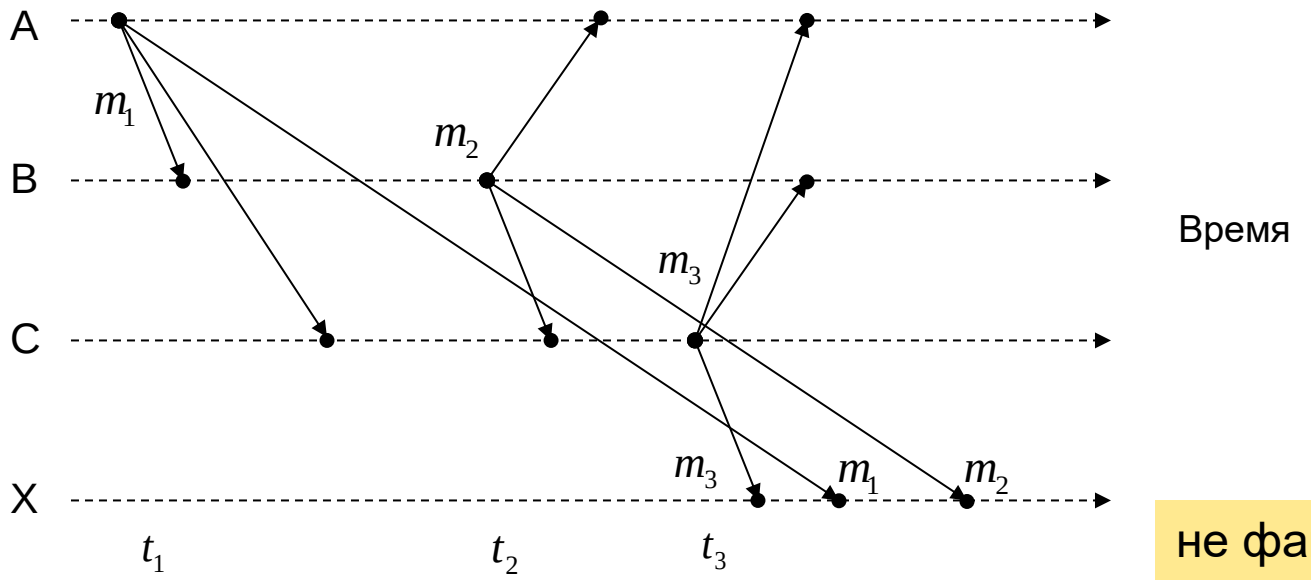
Асинхронные распределенные системы

- 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



не факт, что t1<t2<t3

Порядок сообщений, наблюдаемый различными пользователями, может быть различный

Упорядочивание событий

ITIVITI

LET IT TRADE

Изначально A == B == 0

Thread 1	Thread 2	Thread 3
A = 1;	if (A == 1) B = 1;	if (B == 1) X = A;

X == ? при условии, что B == 1 в Thread 3

УНИВЕРСИТЕТ ИТМО

Упорядочивание событий

ITIVITI

LET IT TRADE

Изначально A == B == 0

Thread 1	Thread 2	Thread 3
A = 1;	if (A == 1) B = 1;	if (B == 1) X = A;

X == ? при условии, что B == 1 в Thread 3

Ожидаем, что запись A = 1 становится видимой одновременно для всех потоков?

Упорядочивание событий

ITIVITI

LET IT TRADE

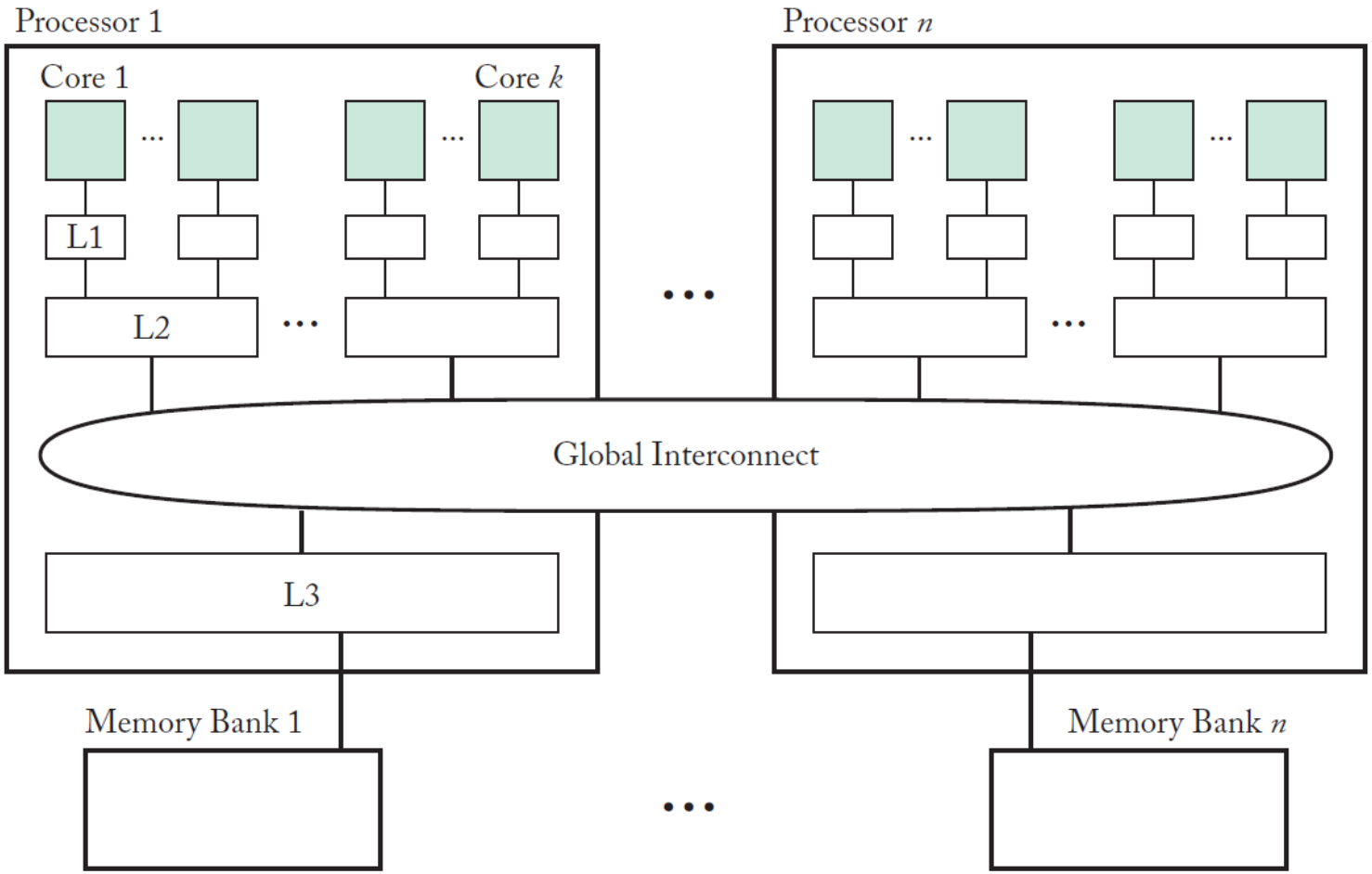
«The network is the computer»

John Gage, Sun Microsystems

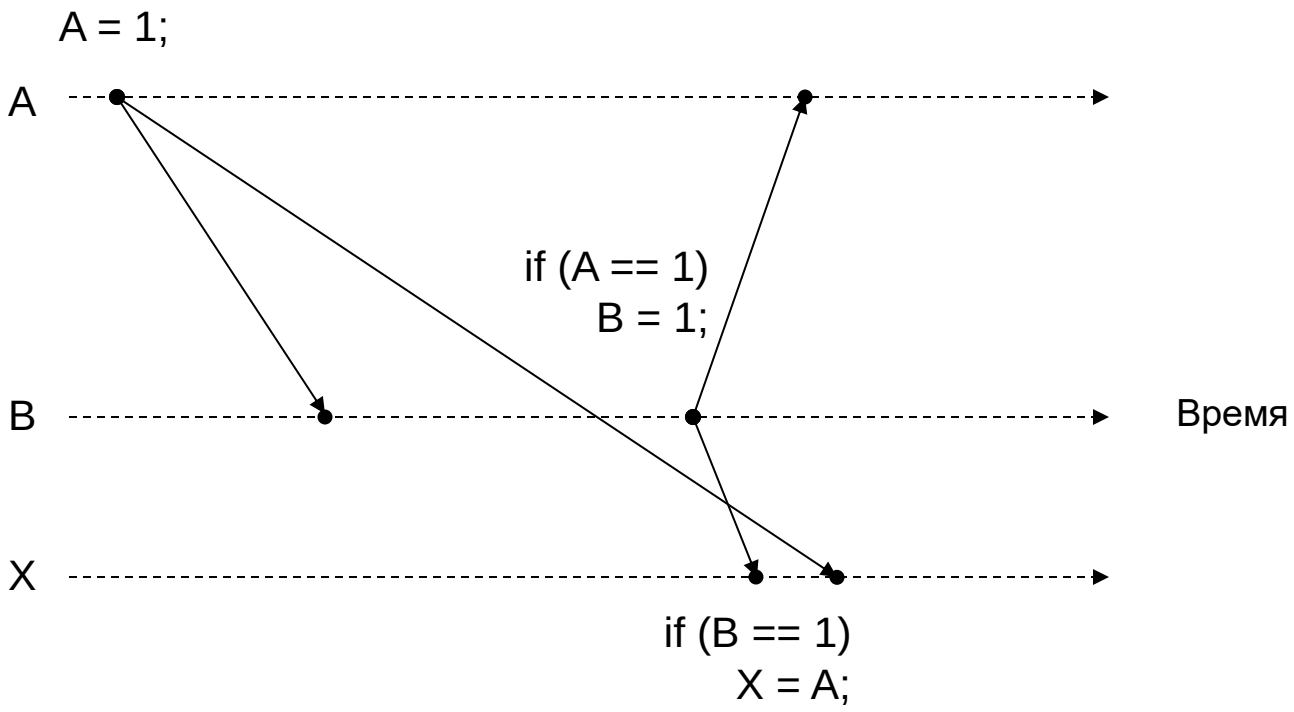
«… but the computer is also the network»

УНИВЕРСИТЕТ ИТМО

Упорядочивание событий

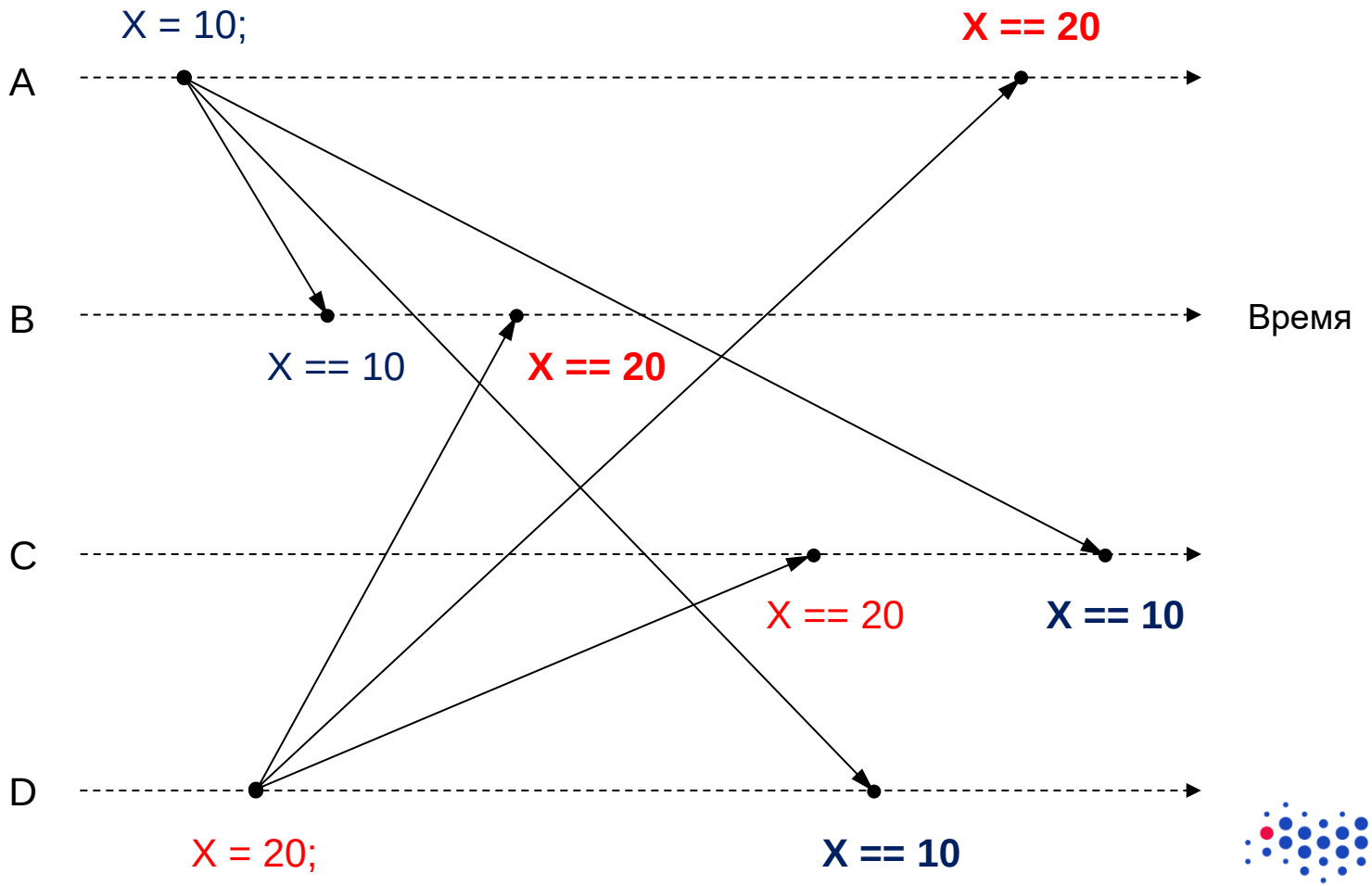


Упорядочивание событий



Порядок событий, наблюдаемый различными потоками, может быть различный

Когерентность кэша



Когерентность кэша

ITIVITI

LET IT TRADE

Изначально X == flag1 == flag2 == 0

Thread 1	Thread 2	Thread 3	Thread4

X = 10;
flag1 = 1;
	X = 20;
	flag2 = 1;
		while(flag1 != 1);
		while(flag2 != 1);
		
		r1 = X;
			while(flag1 != 1);
			while(flag2 != 1);
			
			r2 = X;

r1 == 10 r2 == 20 ?

УНИВЕРСИТЕТ ИТМО

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

ITIVITI

LET IT TRADE

- 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

УНИВЕРСИТЕТ ИТМО

Централизованные vs распределенные алгоритмы

ITIVITI

LET IT TRADE

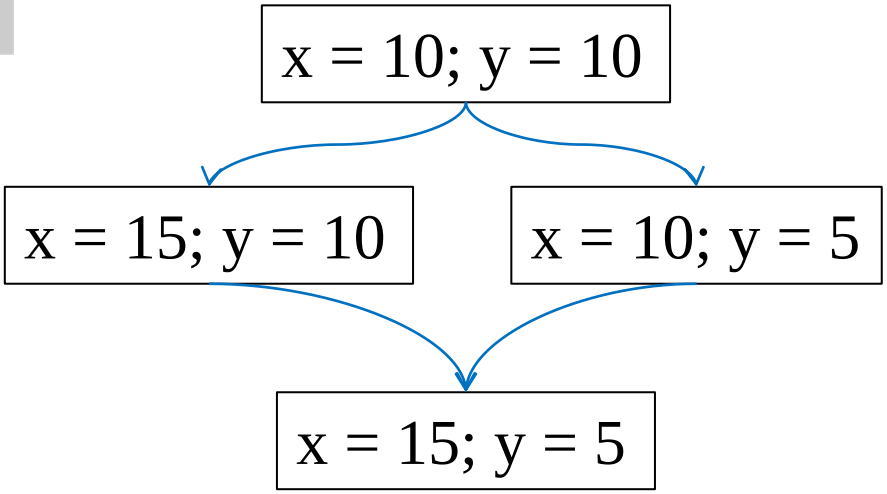
- Отсутствие детерминизма
  - Независимое исполнение процессов и случайные задержки передачи сообщений
  - Выполнение может быть описано разными последовательностями глобальных состояний
  - Следствие: нет смысла говорить, что то или иное состояние достигается по ходу выполнения распределенного алгоритма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

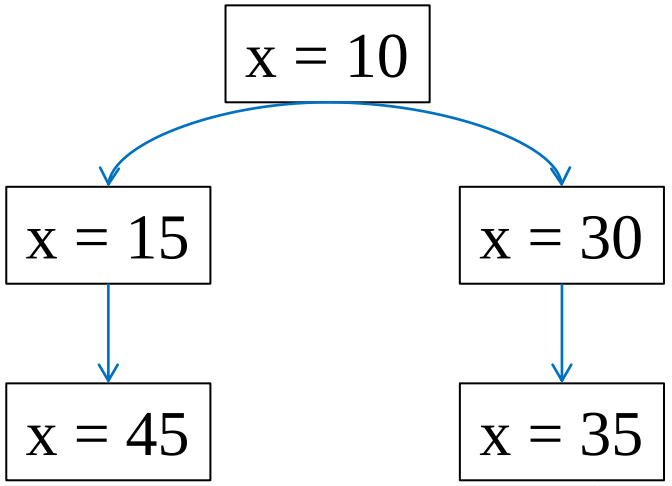
УНИВЕРСИТЕТ ИТМО

Недетерминизм

$x = 10; y = 10$

$x = x+5; y = y - 5$	$x = x+5; x = 3x$
Чередование	Состязание





Масштабируемость PC

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

Масштабируемость PC

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

Сложности географической масштабируемости

ITIVITI

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

LET IT TRADE

УНИВЕРСИТЕТ ИТМО

Технологии масштабирования

ITIVITI

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)

LET IT TRADE

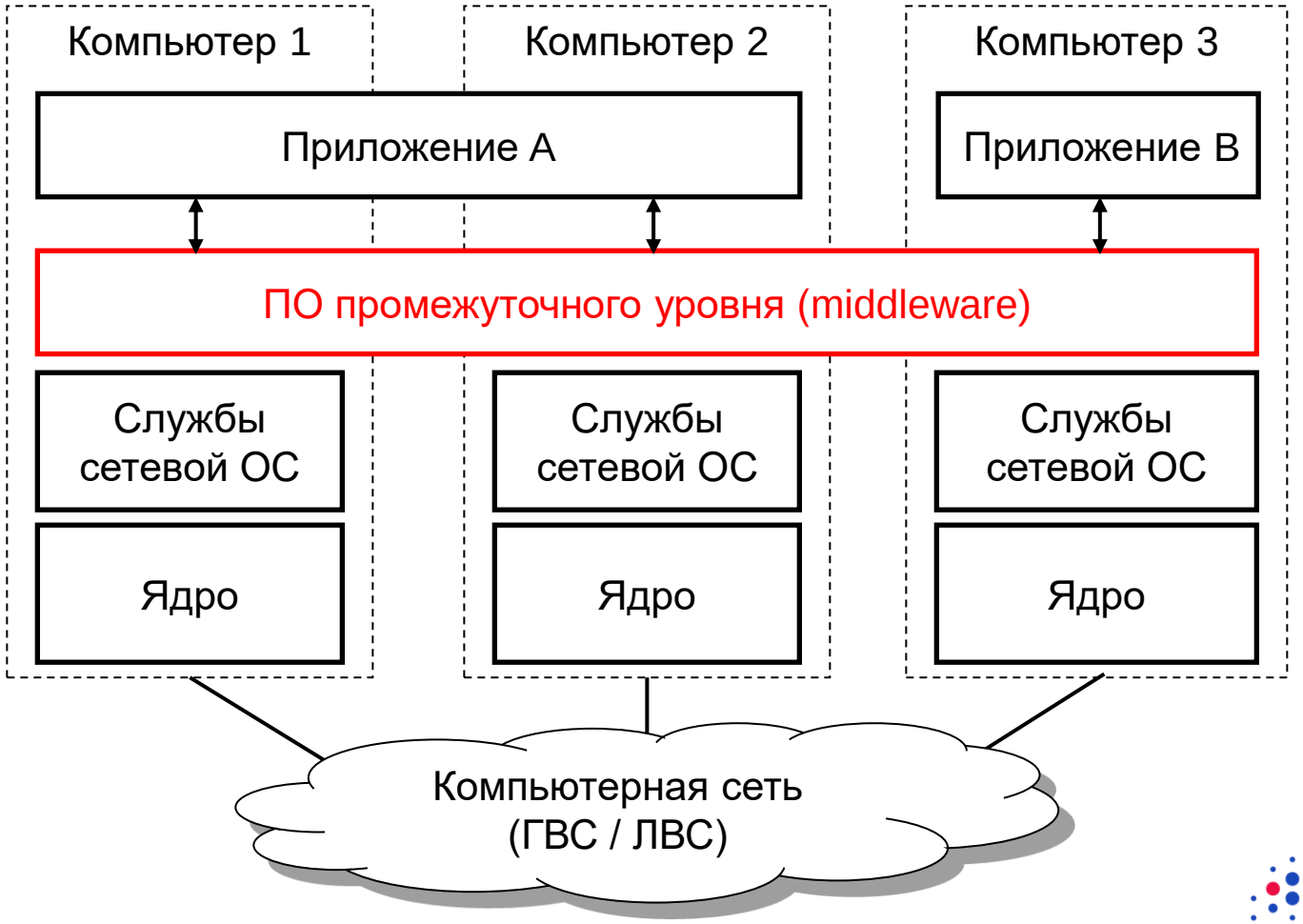
ПО промежуточного уровня (middleware)

ITIVITI

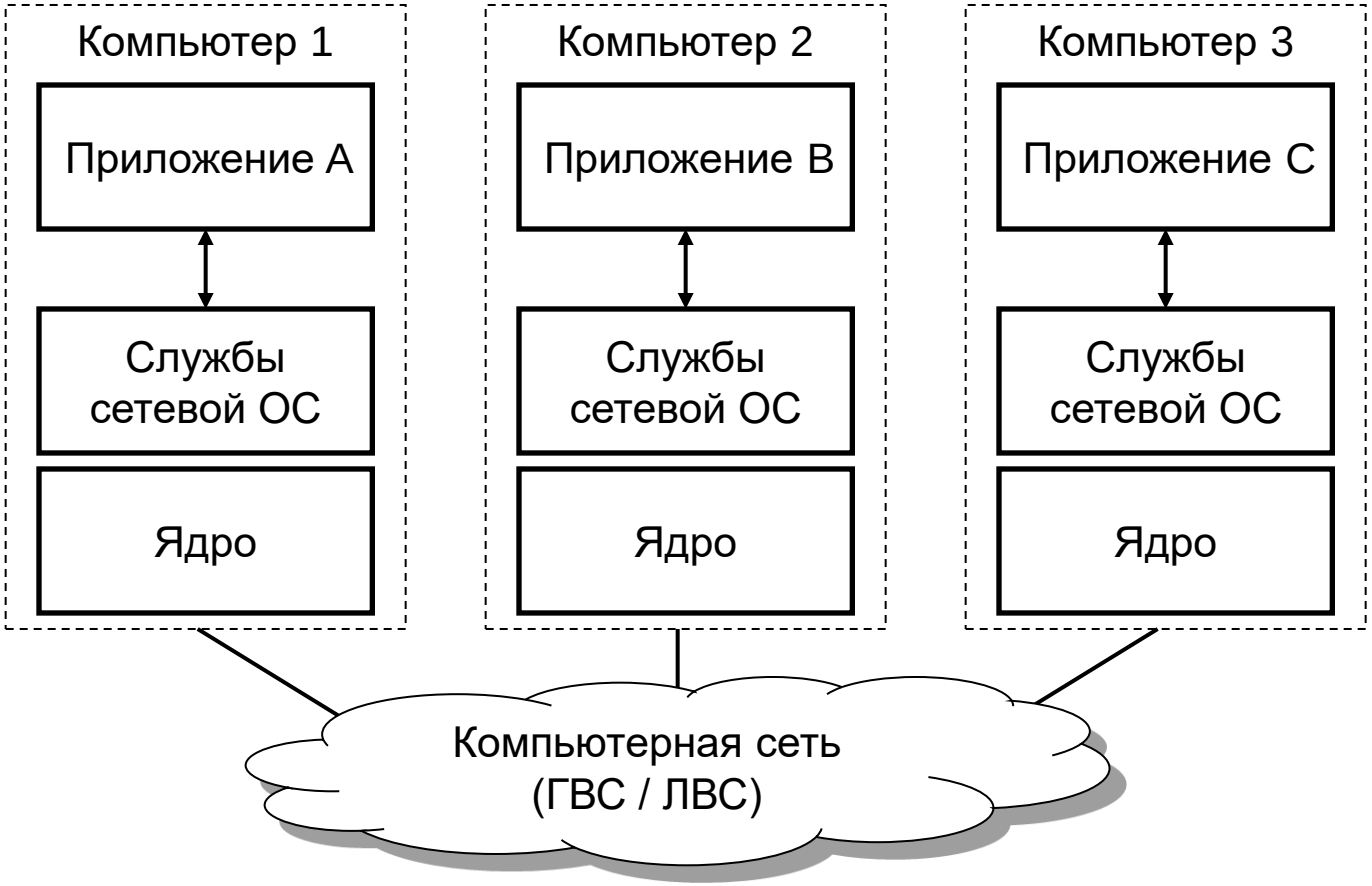
LET IT TRADE

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


Компьютер 1
Приложение A
Службы сетевой ОС
Ядро
Компьютер 2
Приложение B
Службы сетевой ОС
Ядро
Компьютер 3
Приложение C
Службы сетевой ОС
Ядро
Компьютерная сеть (ГВС / ЛВС)

ПО промежуточного уровня (middleware)

ITIVITI

LET IT TRADE

- 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()