

Распределённые системы.

Коньков К. А., Бабичев С. Л.

Московский физико-технический институт

Москва, 2009-2022

Требуется увеличение мощности вычислительной системы.
Этапы:

- Конвейеризация. Внутренний параллелизм устройств — памяти, блоков сложения, умножения.
- Многопроцессорность. Увеличение количества исполнительных узлов.
- Кластеризация. Агрегация вычислительных систем быстрыми локальными линиями связи.
- Облакизация. Агрегация вычислительных систем медленными общедоступными линиями связи.

Решение больших расчётных задач.

Хранение и обработка **больших данных**

Цель практики: понять *изнутри* функционирование распределённых систем.

Мы — не пользователи РС, мы — *разработчики РС*

Мы изучим:

- архитектурные особенности РС;
- технологии, применяемые в разработке РС;
- вопросы переносимости программ;
- асинхронное и синхронное взаимодействие гетерогенных вычислительных систем;
- архитектуру клиент/сервер;
- вопросы реализации удалённого вызова процедуры;
- распределённые алгоритмы;
- вопросы безопасности;
- распределённое хранение и репликацию;

Мы напишем программы:

- реализующие ряд распределённых алгоритмов;
- реализующие небольшую систему метакомпьютинга;

Что нужно для прохождения курса?

- Представление об основных конструкциях языка C++ — классах, методах, конструкторах, деструкторах, STL.
- Понимать функционирование операционных систем — процессы, потоки, примитивы синхронизации, ввод/вывод, сетевые функции.

Разбор сложных систем = анализ + изучение компонент + синтез.

Мы разложим РС на подсистемы

- топологии;
- транспортную;
- исполнения;
- синхронизации;
- безопасности;
- репликации;
- хранения;
- вычислений.

Одно из определений: РС есть совокупность взаимодействующих процессов, выполняющих общую задачу.

Сеть: совокупность двух множеств: узлов (процессов)

$$P = \{P_i, i \in 1 \dots n\},$$

и связей, множества троек

$$L = \{(P_i, P_j, C_{ij}), i, j \in 1 \dots n\}$$

C_{ij} есть вес связи от узла i до узла j

Статические сети: множества P и L не изменяются во времени.

Сети с полной информацией: каждый из P_i имеет информацию о множествах P и L .

Процессы в РС взаимодействуют только путём посылки и приёма сообщений и каждый из них не имеет информации о сети в целом.

- Алгоритмы, основанные на статическом представлении
- Алгоритмы, основанные на понятии процесса.
 - STP, Flood, Chandimistra, Toueg, ...

- Одноадресные (unicast). $P_i \rightarrow P_j$
- Многоадресные (multicast). $P_i \rightarrow \{P_k, \dots, P_n\}$
- Широковещательные (broadcast). $P_i \rightarrow \text{Reachable}(P_i)$

В курсе мы будем моделировать распределённые системы. Наша модель — набор классов, абстрагирующих такие понятия, как

- процесс
- сообщение
- сетевая инфраструктура
- событие
- очередь сообщений
- алгоритм
- исполнитель алгоритма

Моделирование позволит изучить взаимодействие процессов, обменивающихся сообщениями.

Можно моделировать:

- прохождение сообщения от одного процесса к другому с задержкой или без;
- потерю и дублирование сообщений;
- обработку сообщений рабочими функциями процессов.

- контексты процесса и алгоритма;
- архитектура клиент-сервер;
- синхронные и асинхронные запросы;
- контекст клиента на сервере;
- способы обслуживания клиентов PerSession и PerRequest.

Подсистема синхронизации отвечает за определение отношений причина-следствие а также за обеспечение непротиворечивости.

- Физическое время
 - Алгоритм Кристиана
 - Алгоритм Беркли
- Логическое время
 - Метки времени Лампорта
 - Векторные метки времени

- Алгоритмы выборов
 - Алгоритм забияки (bully)
 - Алгоритм Чанга-Робертса
- Алгоритмы взаимного исключения
 - Централизованный алгоритм
 - Распределённый алгоритм
- Алгоритмы консенсуса (PAXOS, Raft)

- Две ипостаси защиты:
 - ① Защита связи — защищённый канал, идентификация целостности сообщений
 - ② Авторизация и контроль доступа
- Механизмы защиты СААА.
- Объекты и субъекты защиты.

- Симметричная схема
- Несимметричная схема
- Хеширование

- избыточное хранение данных
- (n, k) –схемы
- дедупликация
- повышение производительности
- синхронизация больших объектов

- сильносвязанная модель
- слабосвязанная модель
- GRID

- метасистема
- задача
- вычислительный узел
- управляющий узел
- метасервер
- контракт
- метаклиент
- решатель
- обёртка (wrapper)