



iSCALARE



Лаборатория суперкомпьютерных технологий для биомедицины, фармакологии и малоразмерных структур

Альтернативы имитационному моделированию

Григорий Речистов

grigory.rechistov@phystech.edu

- 
- Аналитические модели
 - Стохастические модели
 - Трассы (оффлайн модели)

На предыдущих лекциях

- Функциональные и потактовые модели
 - «Онлайн» модели
 - Основаны на детальной информации об исполнении
 - Позволяют предсказывать функциональные и скоростные характеристики систем

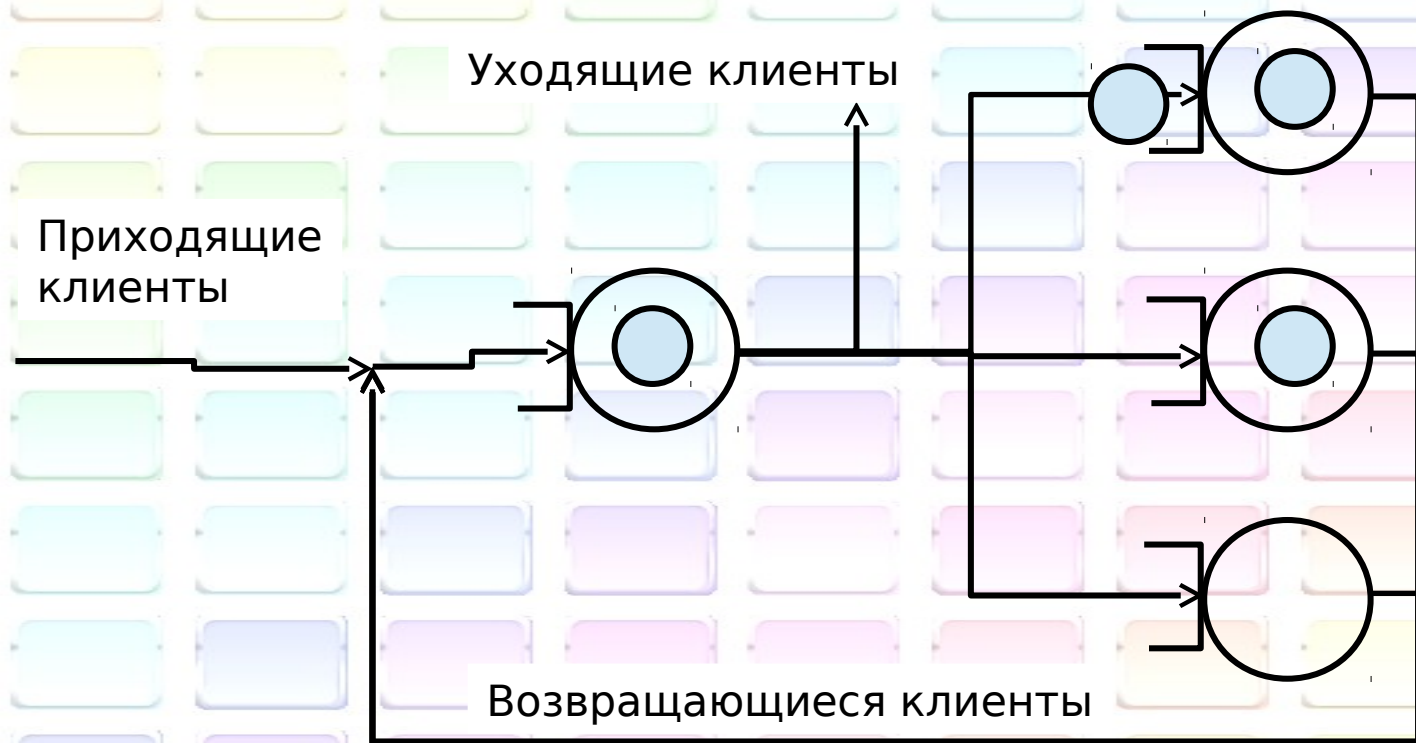
Альтернативы

- Мы не всегда имеем всю информацию о системе
 - И не хотим её иметь
- Цель исследования — производительность ЭВМ
 - Какая из систем работает быстрее?
 - Какова степень загруженности узлов?
 - Что необходимо улучшить?

Очереди обслуживающих центров (queueing systems networks)

- Очерчиваем диаграмму обслуживания
- Входные данные — характеристики производительности
- Используем законы, связывающие производительность подсистем
- Получаем недостающие характеристики производительности

Диаграмма обслуживания



Величины

- Скорость прибытия (arrival rate) λ
- Пропускная способность (throughput) X
- Степень утилизации (utilization) U
- Среднее время обслуживания запроса (service requirement per request) S
- Длина очереди некоторого центра q

Законы

- Закон использования (The utilization law): $U = XS$
- Закон Литтла (Little's Law): $N = \lambda T$
- Соотношение для отклика системы: $R = N / X - Z$
- Закон вынужденного потока (The forced flow law):
 $X_k = V_k X$
- Баланс потока: $\lambda = X$

Упрощающие предположения

- В простейшем случае мы имеем дело с сепарабельными (separable) системами
 - Производительность не зависит от политик обслуживания в отдельных узлах
 - Законы применимы как к частям системы, так и к целому
- Изучаем устойчивое состояние
 - Отсутствия насыщения узлов ($U < 1$)

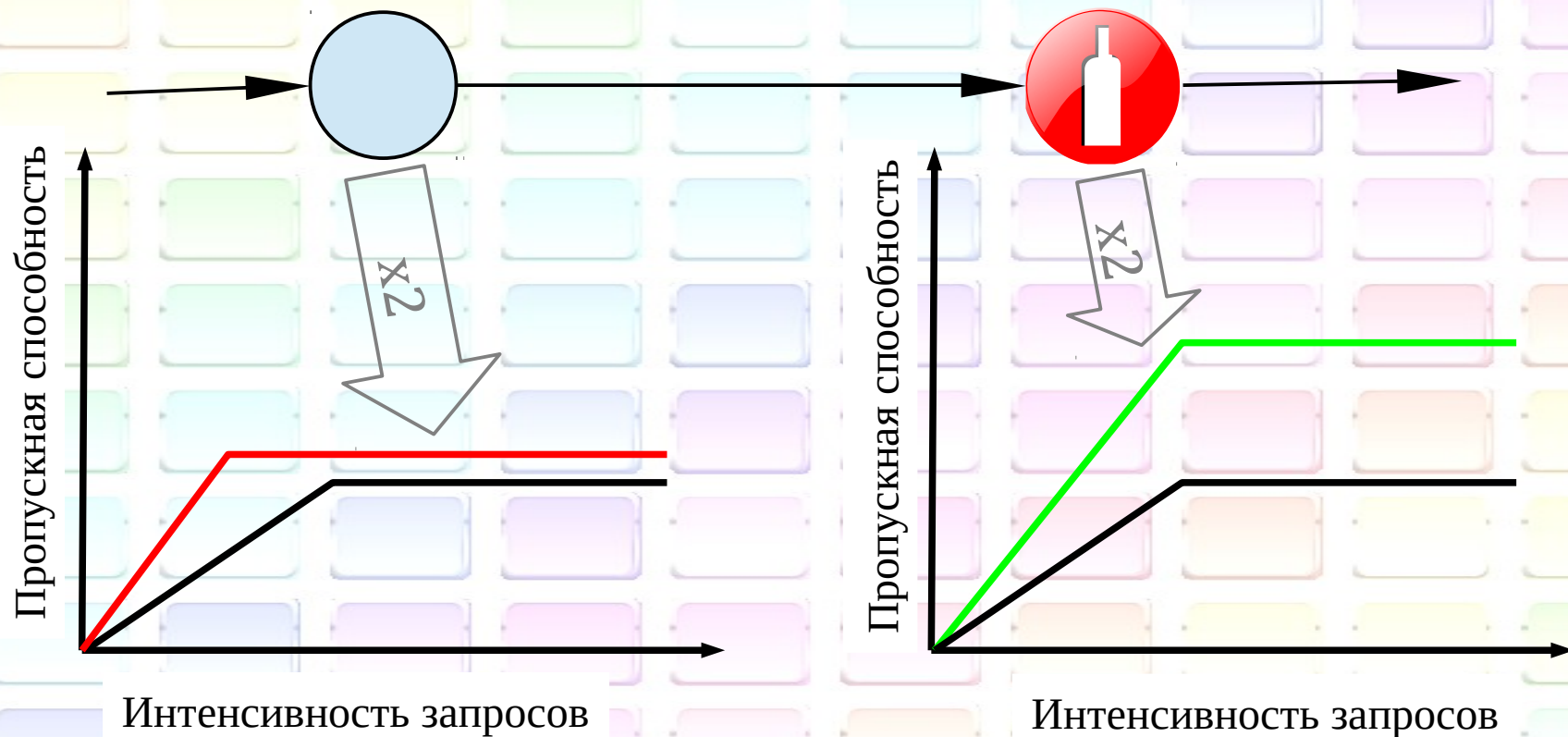
Откуда брать входные данные

- Из реальных экспериментов!
 - Измерения, датчики
производительности, спецификации
аппаратуры, бенчмарки...

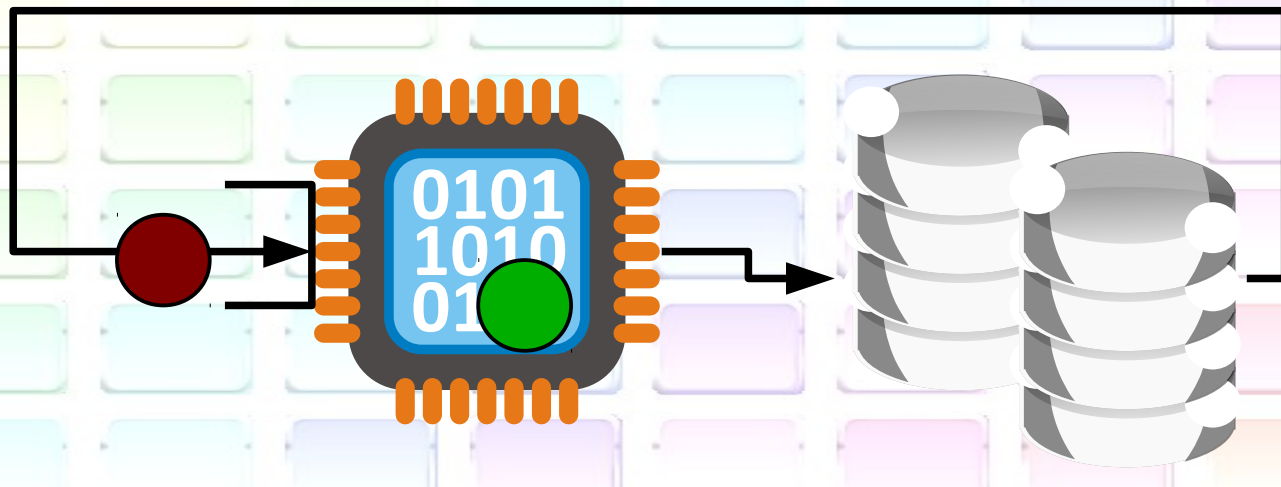
Что ищется

- Недостающие характеристики
 - Время отклика интерактивной системы
 - Загруженность отдельных узлов
 - Пропускная способность всей системы как функция характеристик её частей

Анализ узких мест (bottleneck analysis)



Compute-bound и throughput-bound задачи



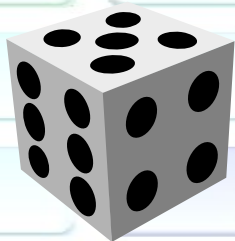
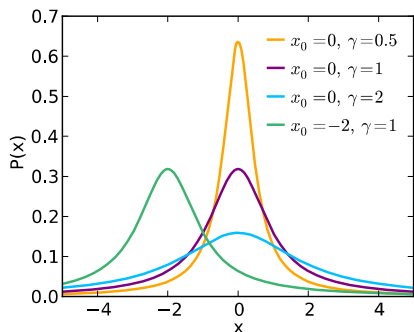
Может оказаться так, что улучшение характеристики компонента приведёт к ухудшению характеристик системы в целом!

Статистический анализ

- Моделируем «внешнее» воздействие на систему случайным процессом с заданными характеристиками
- Изучаем реакцию изучаемой системы
- Является «расширением» queueing networks (точнее, QN — упрощение статистического анализа)

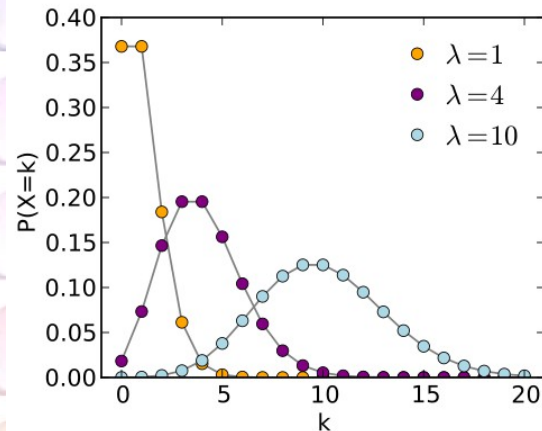
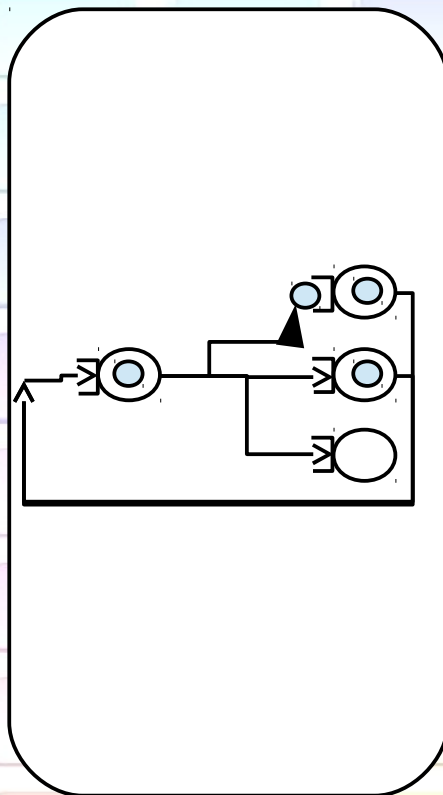
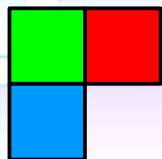
Статистический анализ

Статистические характеристики
входных данных



Г(П)СЧ

Входное
воздействие



Выходные
статистические
характеристики
функционирования

Нотация Кендалла (Kendall) (1/2)

- A/B/c/K/m/Z
 - A — распределение времени между прибытиями
 - B — распределение времени обслуживания
 - c — количество серверов
 - K — ёмкость системы обслуживания
 - m — численность источника клиентов
 - Z — принцип обслуживания

Нотация Кендалла (Kendall) (2/2)

- A/B/c — сокращённый вариант
 - Неограниченная длина очереди ожидания
 - Бесконечный источник клиентов
 - Политика обслуживания FCFS (first come — first serviced)

Распределения

- GI general independent interarrival time
- G general service time
- H_k k-stage hyperexponential interarrival or service time distribution
- E_k Erlang-k interarrival or service time distribution
- M exponential interarrival or service time distribution
- D deterministic (constant) interarrival or service time distribution
- U uniform interarrival or service time distribution

Pro et contra

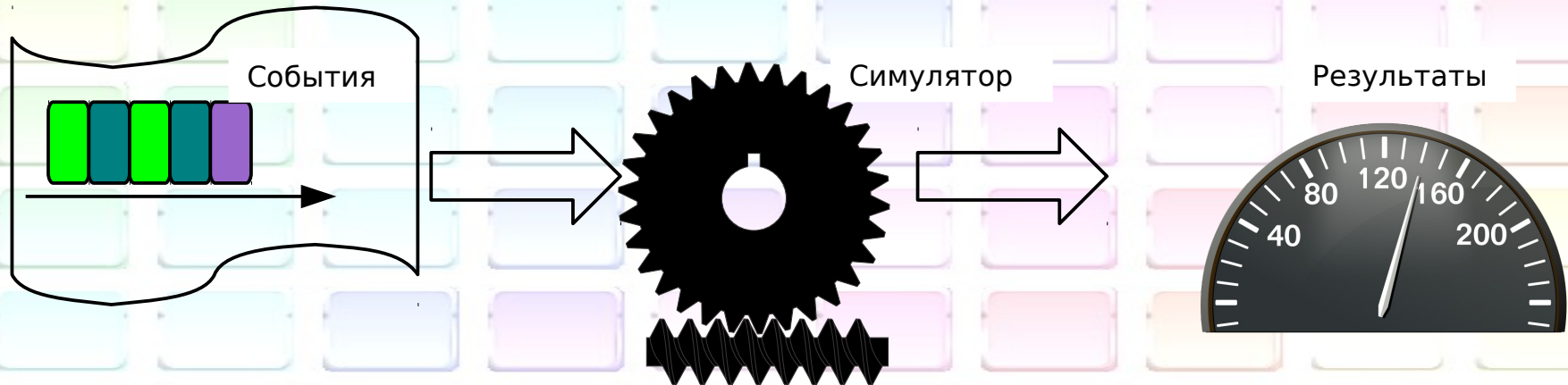
- Точнее QN, позволяют проверить устойчивость системы (выводы QN *исходят* из предположения об установившемся режиме)
- Допускают параллельную реализацию (отдельные эксперименты не связаны друг другом)
- Сложнее в аналитическом описании
- Требуют больше информации об изучаемой системе

Генераторы случайных чисел

- Должен генерировать последовательность, достаточно случайную для нужд конкретной задачи
- Скорость генерации должна быть высокой
- Для параллельных реализаций: СП должны быть независимы
- Не обязан быть надёжным с криптографической точки зрения (хотя такое качество не помешает)

Оффлайн-симуляция

События, связанные с работой симулируемой системы, могут браться из её *записанной* истории



Оффлайн == event-driven simulation

Онлайн == execution driven simulation

Трасса

time=10 read addr=0x45df4 result=0x0455

time=14 write addr=0x35df4 data=0xffff

time=20 interrupt 10

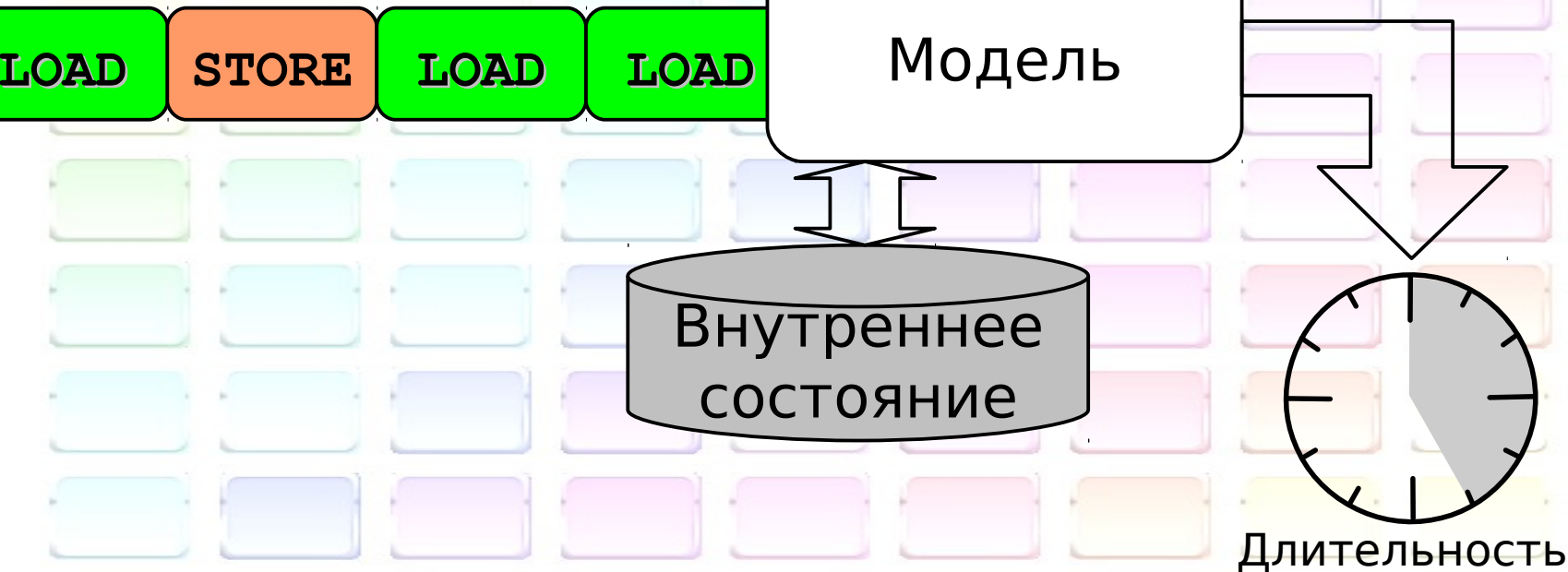
time=25 port write addr=0x10 data=0xabcd

...

Собирается на реальной аппаратуре или на модели

Пример: система кэшей/памяти

Трасса доступов в память



Ещё использование трасс

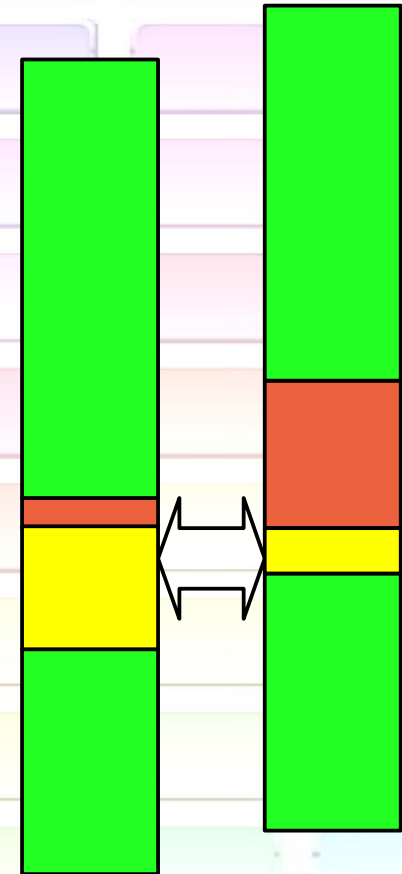
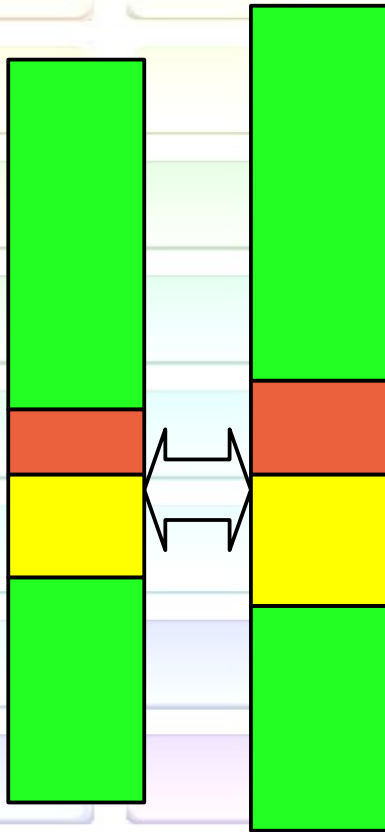
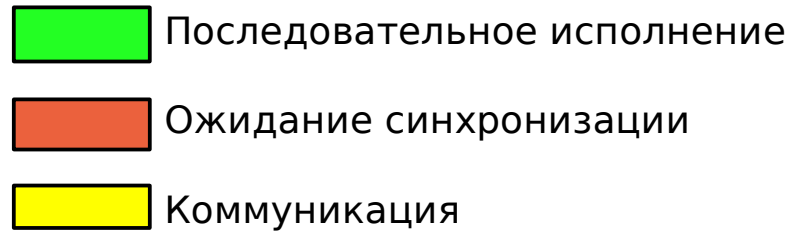
- Тестирование симуляторов
 - Проверка корректности состояния на каждом шаге
- Канал связи между различными классами моделей
 - Функциональная $\Leftrightarrow$ потактовая модель

Pro et contra

- Трассы могут быть переиспользованы в различных моделях
- Скорость моделирования может быть выше, чем у онлайн-модели
- В трассе не сохраняется информация о коде приложения (можно перепоручить анализ)
- Большой объём файлов
- Трудно адекватно отразить историю параллельной системы

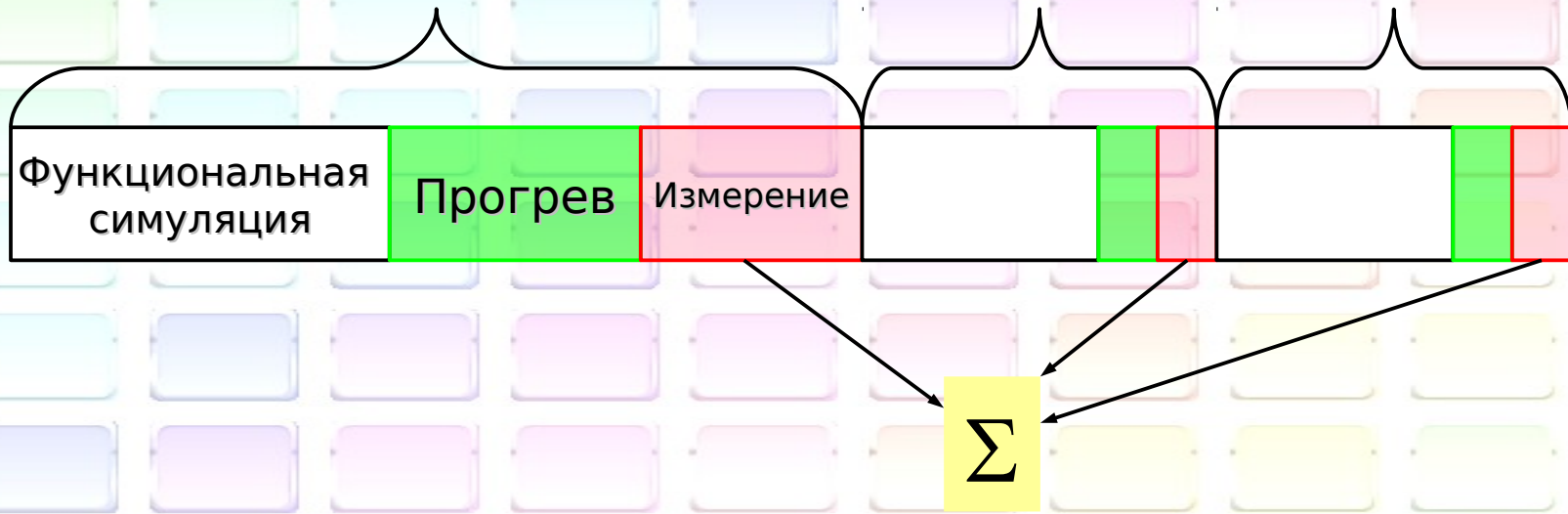
Параллельные системы

Длительность событий из различных потоков зависит от *порядка* операций



Трюки

- Фильтрация событий
- Мета-события, связанные с синхронизацией
- Статистическая выборка событий



Рекомендуемая литература (1/2)

- Lazowska Edward D. et al. **Quantitative system performance: computer system analysis using queueing network models.** —Upper Saddle River, NJ, USA: Prentice-Hall, Inc. 1984.
- Allen Arnold O. **Probability, Statistics, and Queueing Theory with Computer Science Applications** (Second Edition). — Academic Press Inc., 1990. — 768 C.

Рекомендуемая литература (2/2)

- Cain Harold W. **Precise and Accurate Processor Simulation.** — 2002.

http://pages.cs.wisc.edu/~cain/pubs/caecw2002_final.pdf

- Lee Hyunjin, J et al. **Two-phase trace-driven simulation (TPTS): a fast multicore processor architecture simulation approach** // Software – Practice and Experience. — 2010. — T. 40, No 3. — C. 239-258.

<http://www.cs.pitt.edu/cast/papers/lee-spe10.pdf>

- Rico Alejandro, et al. **Trace-driven simulation of multithreaded applications** //ISPASS. — IEEE Computer Society, 2011. — 87-96.

http://ispass.org/ispass2011/slides/3_1.pdf

Экскурсия на базу и собеседование

- 31 марта и 7 апреля — экскурсия на кафедру
- 20 апреля — собеседование на кафедру

Спасибо за внимание!

Все материалы курса выкладываются на сайте лаборатории:
http://iscalare.mipt.ru/material/course_materials/

Замечание: все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.