Санкт-Петербургский государственный университет
Факультет прикладной математики – процессов управления

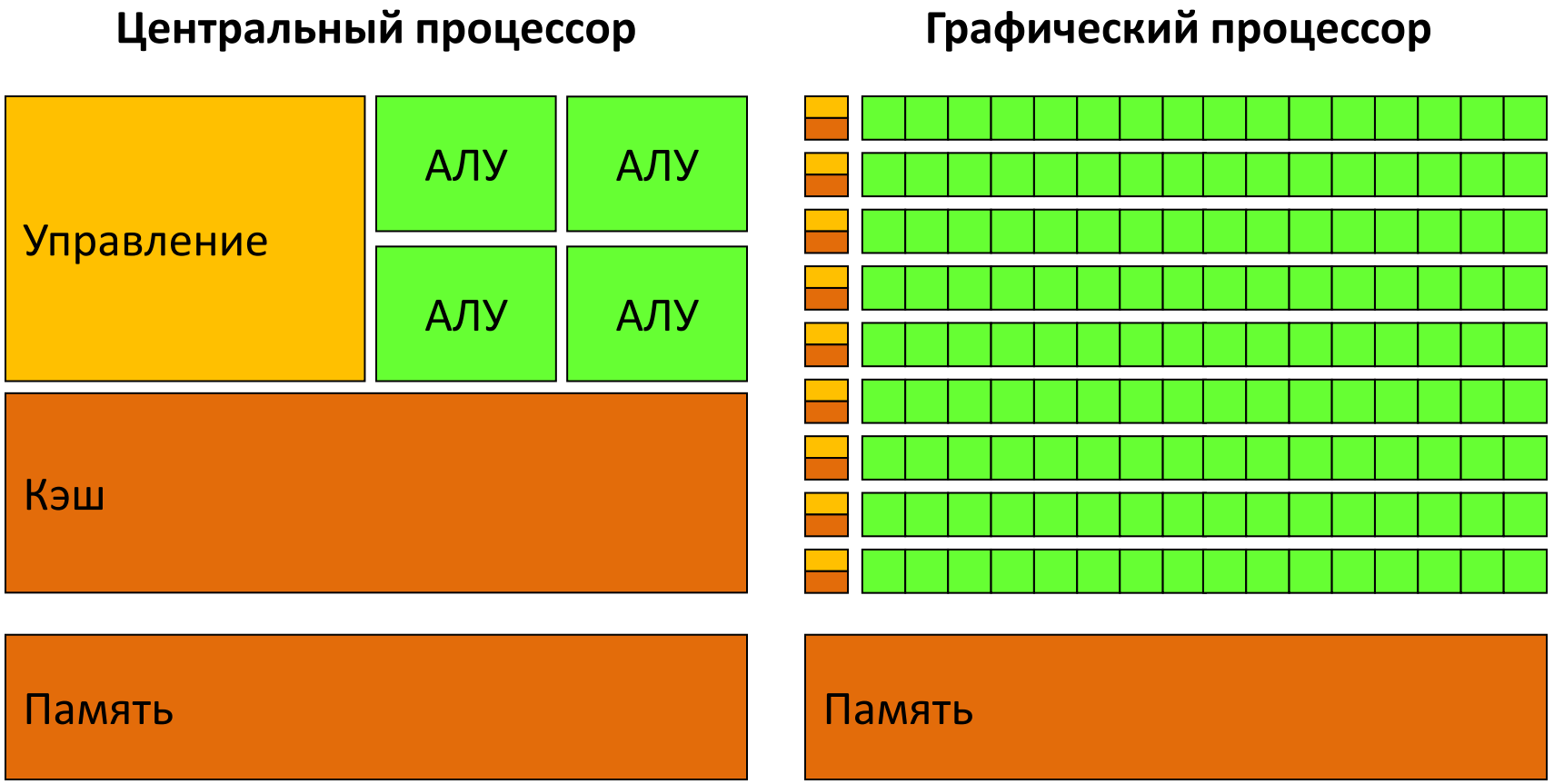
ЭФФЕКТИВНАЯ ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКОГО ПРОЦЕССОРА

Щекалёв Александр Сергеевич
Научный руководитель, к.т.н., доцент Гришкин В.М.

Постановка задачи

- Параллельная реализация метода опорных векторов и метода бустинга деревьев решений
  - Выделение вычислительно-сложных этапов каждого метода
  - Распараллеливание и перенос на графический процессор
  - Оптимизация с учётом архитектуры видеокарты

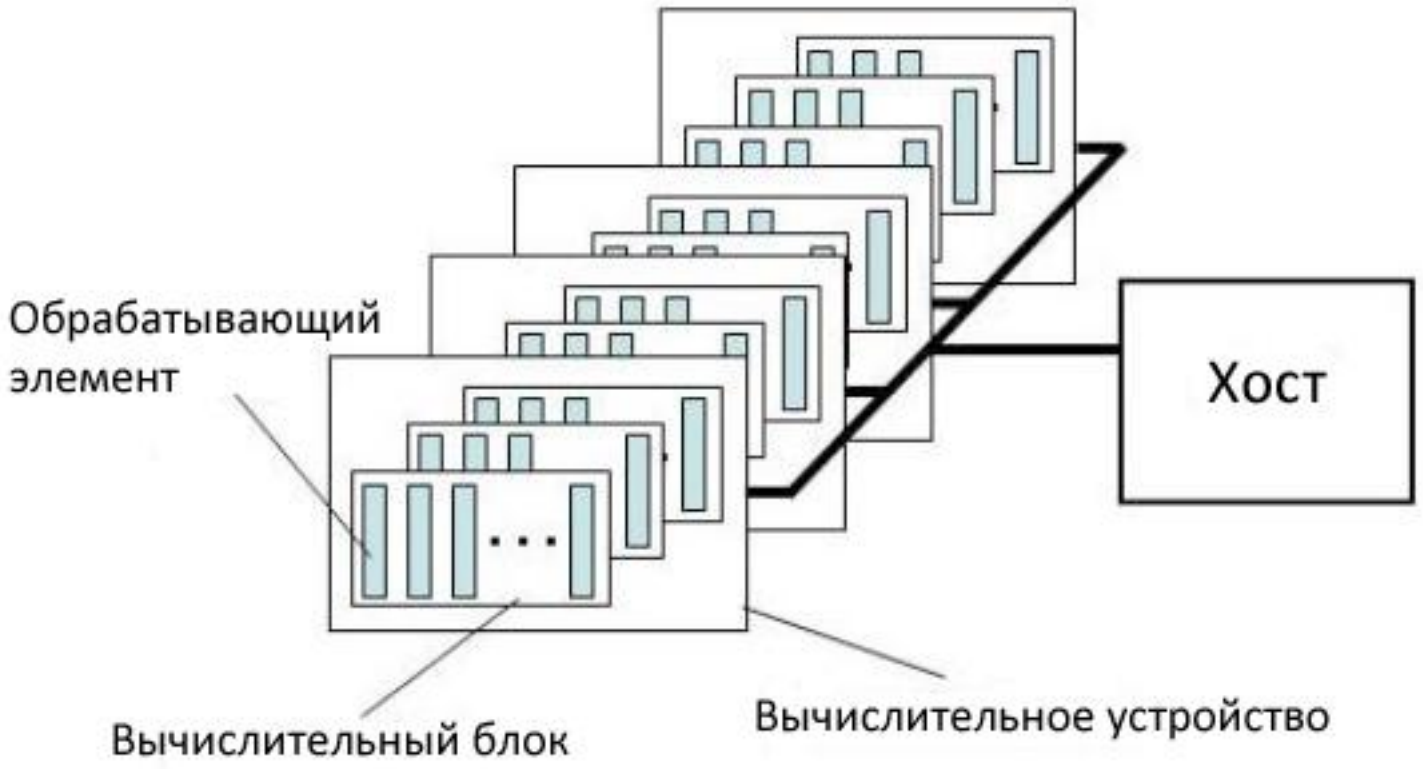
Графические процессоры



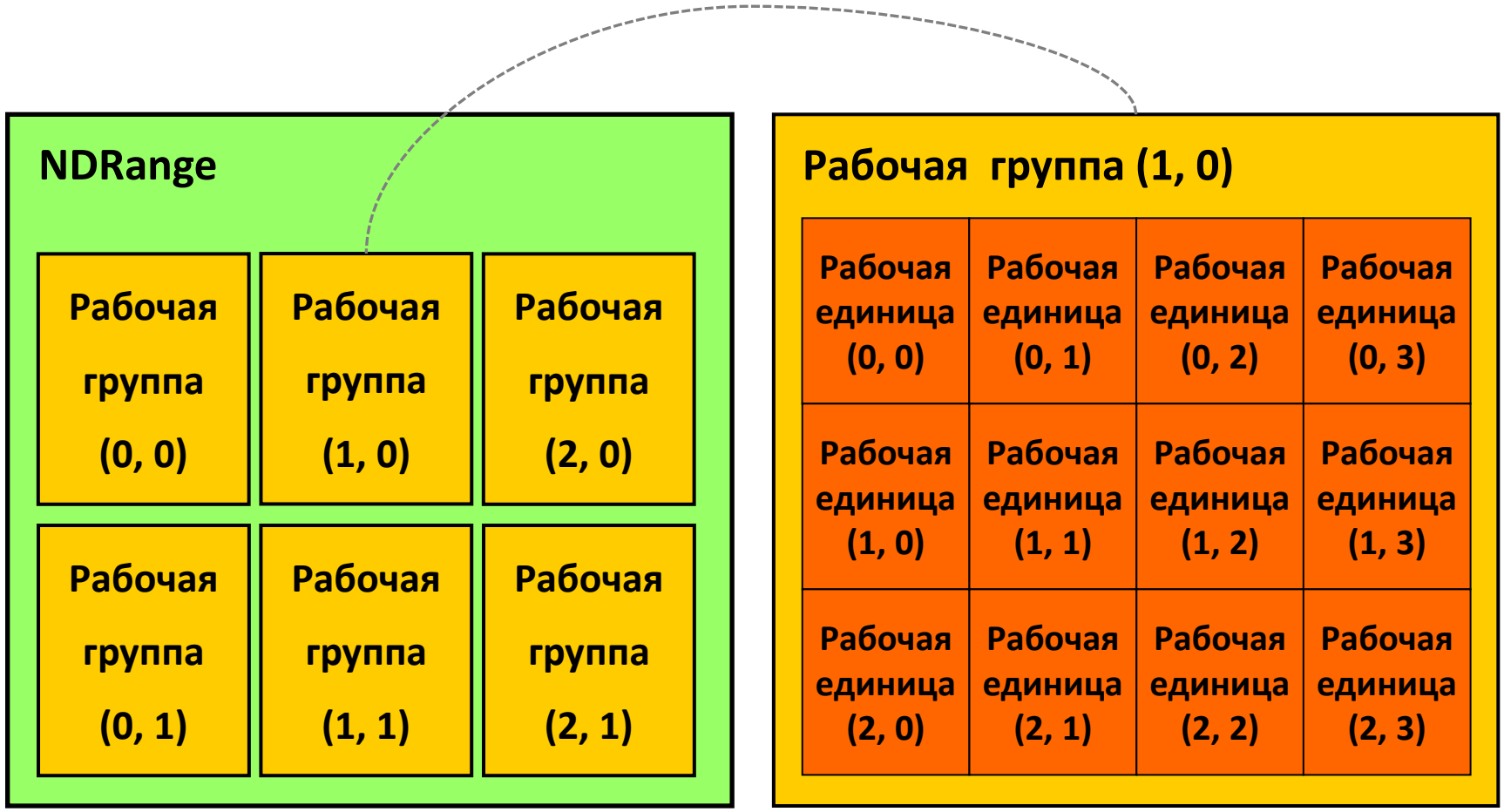
Доступные технологии

- Шейдеры
  - Решение задач, связанных с визуализацией графики
- Технологии от производителей видеокарт
  - NVIDIA CUDA
  - AMD Stream
- OpenCL
  - Универсальный стандарт

OpenCL. Модель платформы



OpenCL. Модель выполнения



OpenCL. Модель памяти

- Доступно четыре вида памяти
  - Глобальная память
  - Константная память
  - Локальная память
  - Собственная память

Метод опорных векторов

- Задача:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha}$$

$$0 \le \alpha_i \le C \quad i = 1 \dots L$$

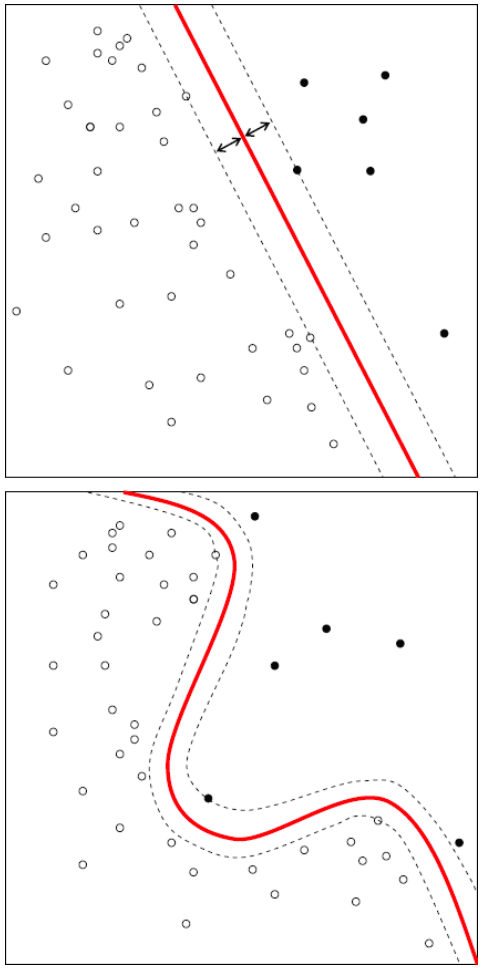
$$y^T \alpha = 0$$

$$Q_{ij} = y_i y_j \phi(x_i, x_j)$$

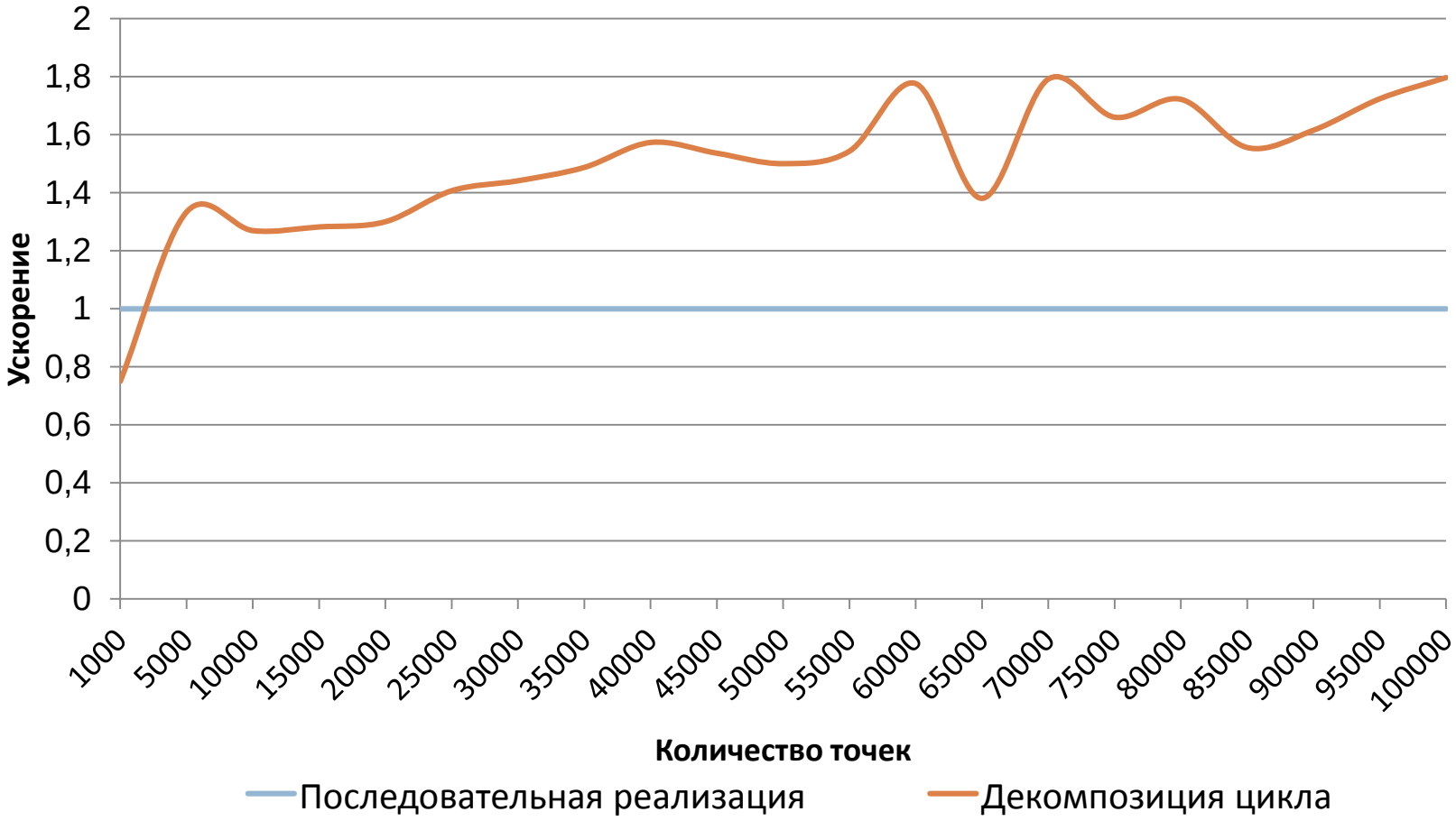
- Примеры ядер:

$$\phi(x_i, x_j) = (x_i \cdot x_j + a)^b$$

$$\phi(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$



Наивная реализация



Обращения к памяти

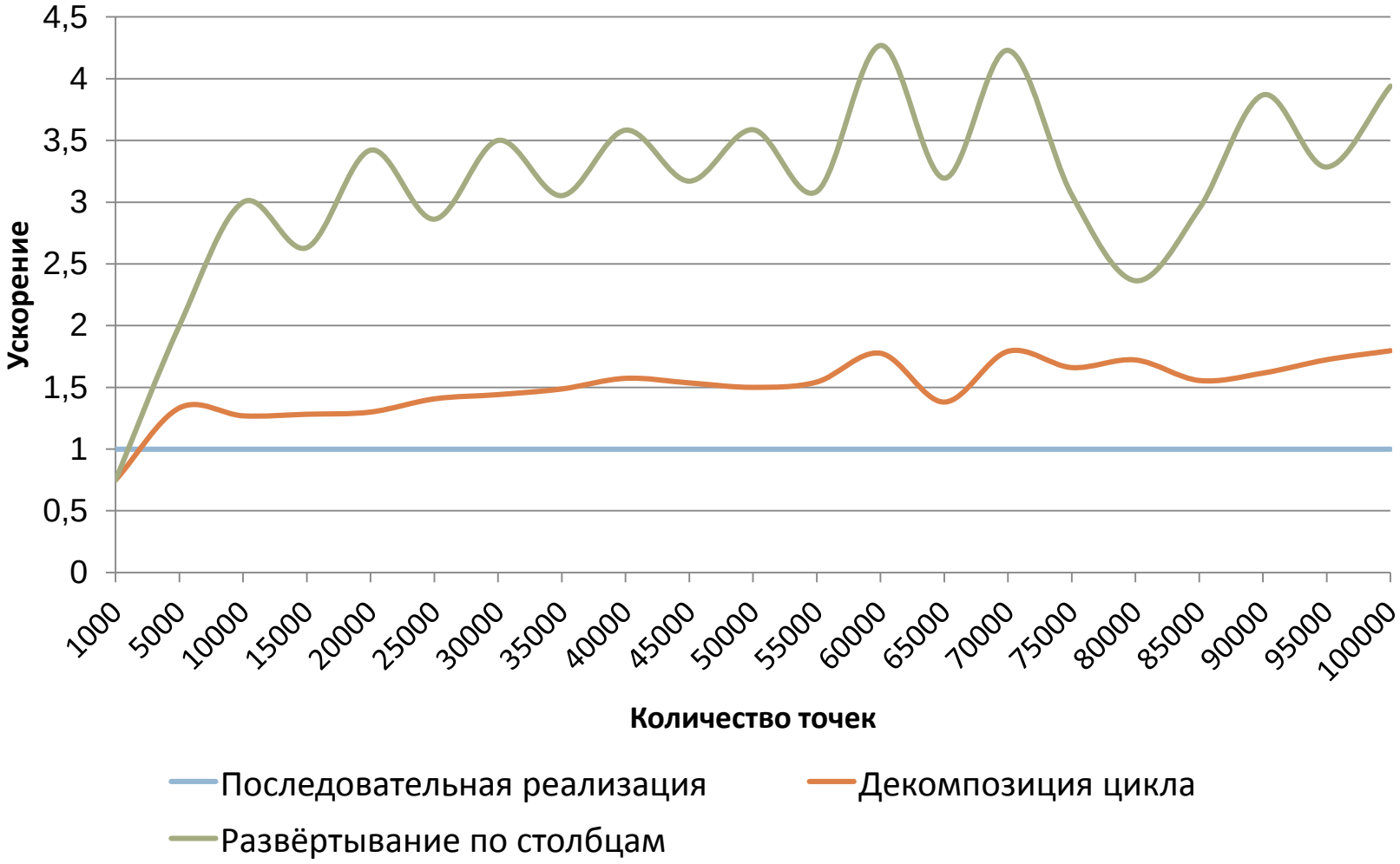
- Развёртывание по строкам

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ ]$$

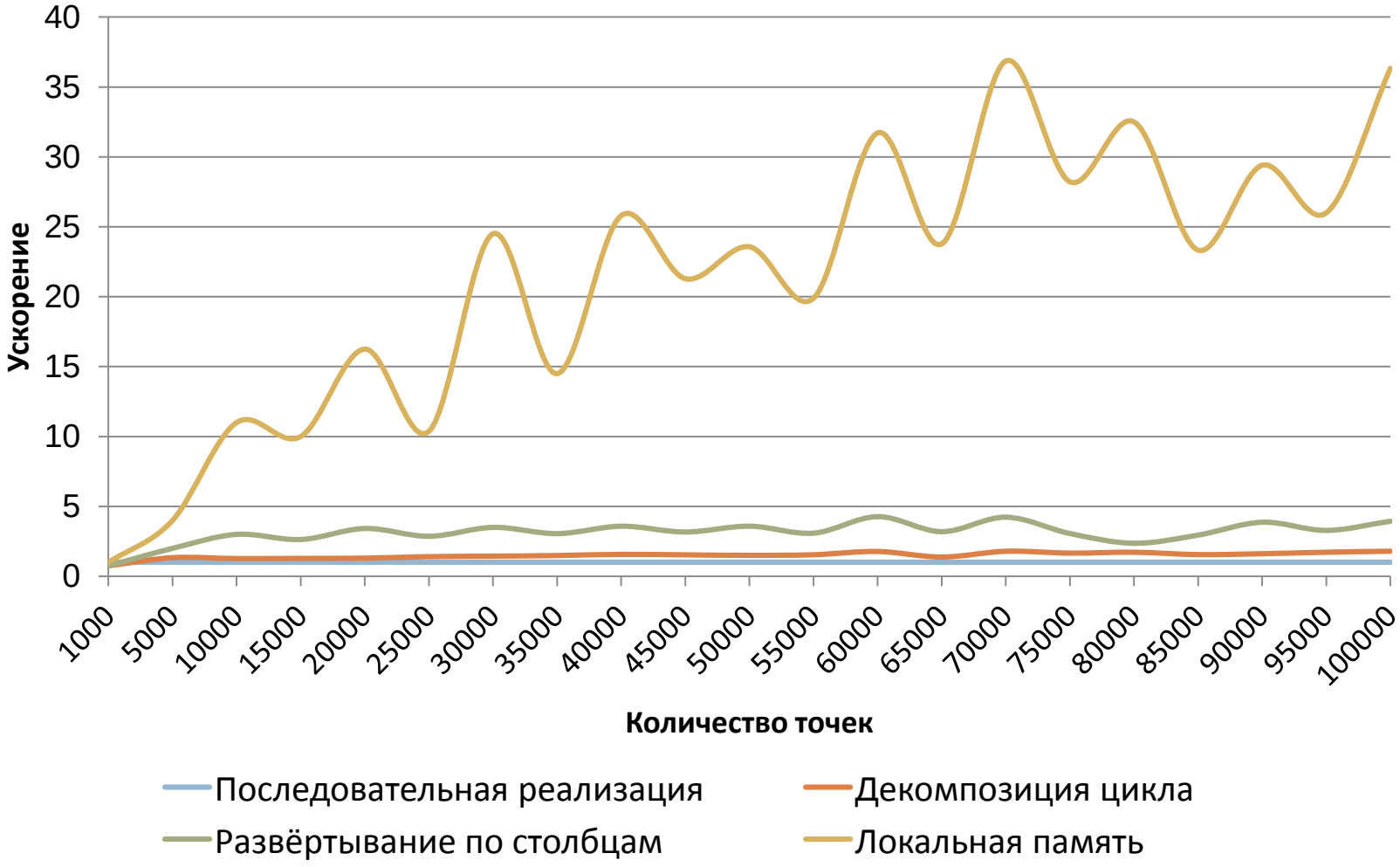
- Развёртывание по столбцам

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = [\ 1\ \ 4\ \ 2\ \ 5\ \ 3\ \ 6\ ]$$

Обращения к памяти

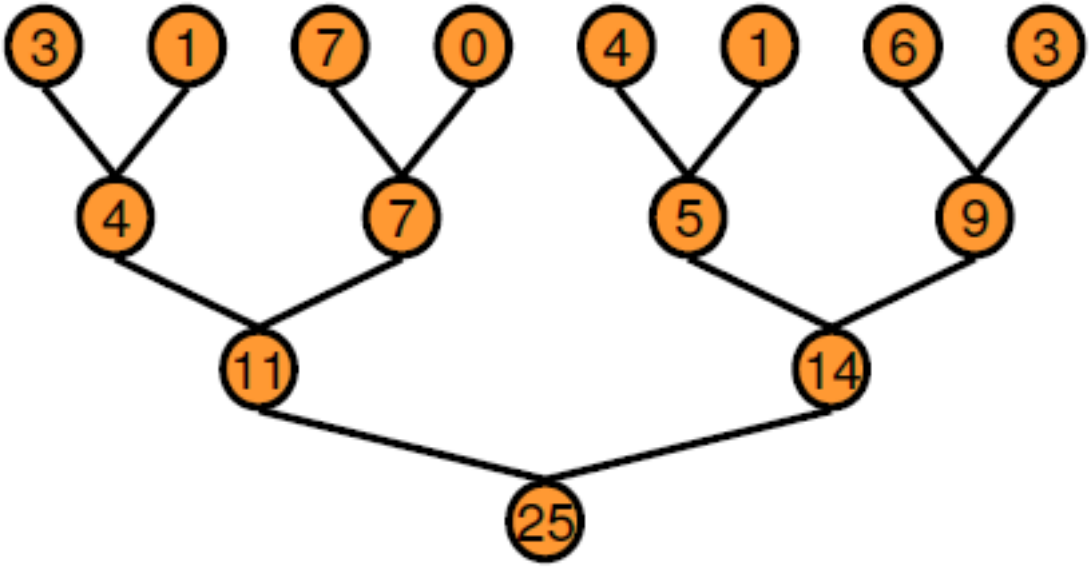


Локальная память



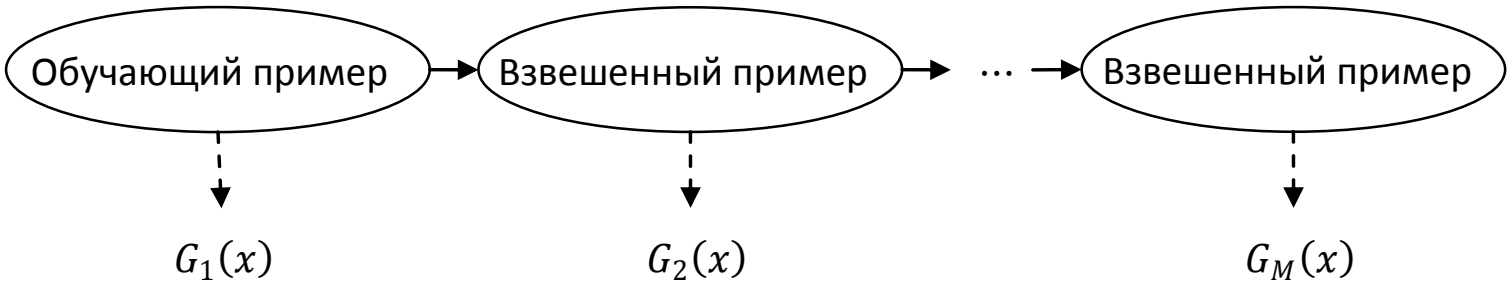
Сокращение

- Операции $\arg\min(\cdot)$ и $\arg\max(\cdot)$ реализуются с помощью алгоритма параллельного сокращения



Бустинг

- Построение сильного классификатора как набор слабых классификаторов



$$G(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$$

Аддитивная модель

- Аддитивное разложение на базисные функции

$$f(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m)$$

- Подбор модели:

$$\min_{\{\beta_m, \gamma_m\}_1^M} \sum_{i=1}^{N} L\left(y_i, \sum_{i=1}^{M} \beta_m b(x_i; \gamma_m)\right)$$

- Прямой пошаговый подбор

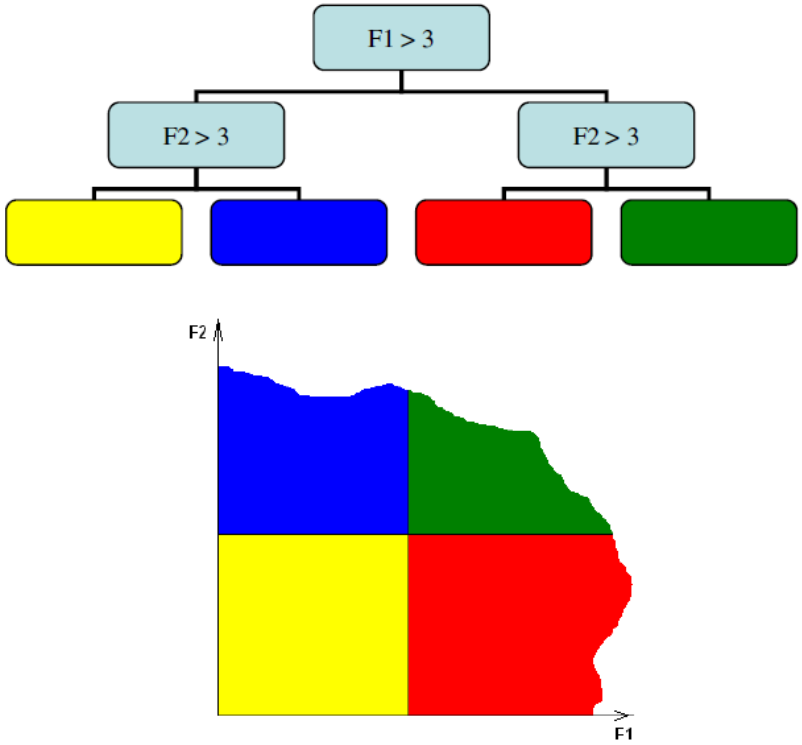
Градиентный бустинг

- Вывод слабого классификатора с учётом отрицательного градиента

$$g_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)}$$

- Подбор методом наименьших квадратов

Деревья решений



$$x \in R_j \Rightarrow f(x) = \gamma_j$$

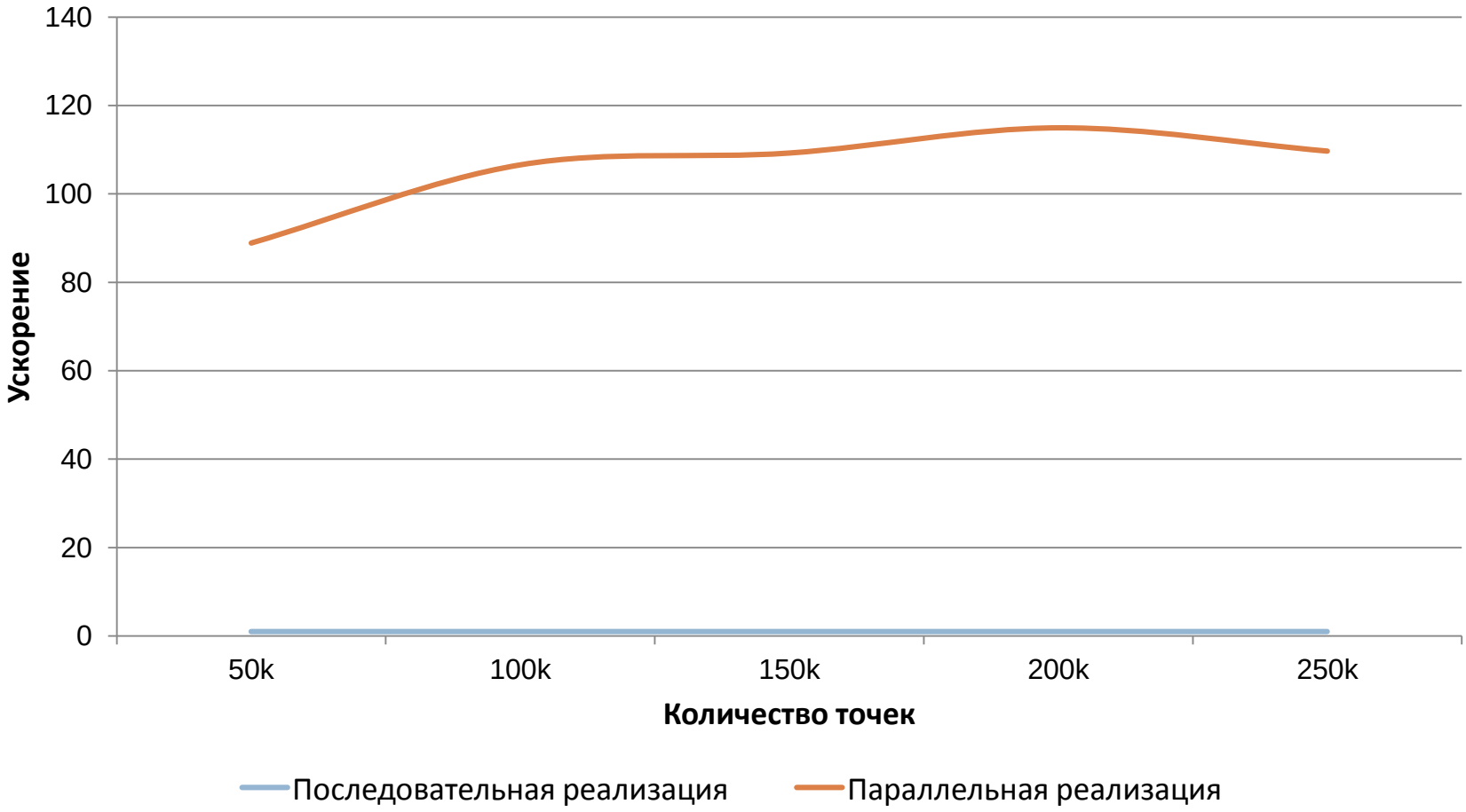
$$T(x;\Theta) = \sum_{j=1}^{J} \gamma_j I\left(x \in R_j\right)$$

$$\Theta = \left\{R_j, \gamma_j\right\}_1^J$$

Бустинг на графическом процессоре

- Будем строить решающие деревья с 64 листьями
- Будем рассматривать 32 разбиения для каждого фактора
- Независимость разбиений – источник параллелизма

Бустинг на графическом процессоре



Результаты

- Метод бустинга деревьев решений полностью реализован на графическом процессоре
  - Ускорение в 70 раз
- Для метода опорных векторов реализованы вычислительно-сложные этапы
  - Ускорение в 36 раз на старом оборудовании
- На основе анализа производительности вычислений на графическом процессоре выявлены основные цели оптимизации

Результаты

- Полученная реализация бустинга применена на практике для решения задачи перекрёстной проверки
  - Кластер из нескольких десятков машин можно заменить одной машиной, использующей видеокарты

Спасибо за внимание!

Перекрёстная проверка

- Оценка обобщающей способности модели
- К-блочная перекрёстная проверка