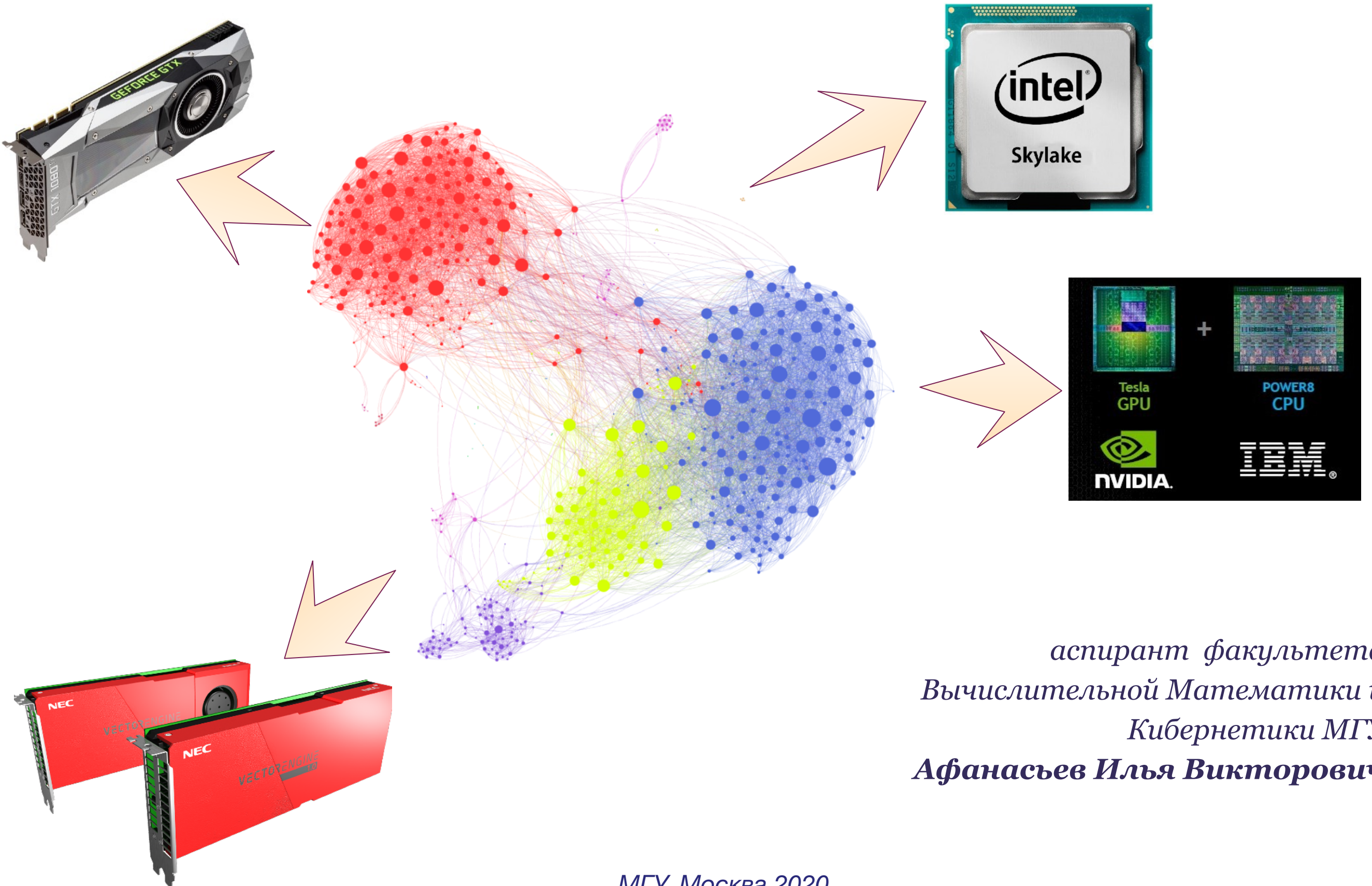


# Использование методов суперкомпьютерного кодизайна для решения графовых задач



аспирант факультета  
Вычислительной Математики и  
Кибернетики МГУ  
Афанасьев Илья Викторович

# Решение графовых задач на суперкомпьютерах

- \* *Графы удачно моделируют многие объекты реального мира (социальные сети, веб-графы, инфраструктурные объекты)*
- \* *В том числе графы применяются и в здравоохранении для моделирования процессов распространения заболеваний в обществе (к примеру агентные модели)*
- \* *Многие из моделируемых графами объектов имеют крайне большой размер (миллионы и миллиарды вершин и ребер), из-за чего возникает необходимость в использовании суперкомпьютеров для графовых вычислений*
- \* *На сегодняшний день в мире существует значительное многообразие суперкомпьютерных архитектур, позволяющих решать графовые задачи с различной степенью эффективности*



# Почему выгодно решать графовые задачи на различных суперкомпьютерных архитектурах?

## Многоядерные центральные процессоры



Архитектуры  
KNL  
Broadwell  
Skylake  
Cascade Lake  
....

Архитектуры  
Power 8  
Power 9  
....

## Графические ускорители



Архитектуры  
Kepler  
Pascal  
Volta  
Turing  
....

## Векторные процессоры

Orchestrating a brighter world



Архитектуры  
SX-ACE  
SX-Aurora TSUBASA  
....

- \* Принципиально отличные подходы к вычислениям для систем различных классов (например векторный/ скалярный поток команд)
- \* Различные поколения архитектур могут кардинально отличаться как с аппаратной, так и с программно-аппаратной точки зрения (например поддержка Unified memory для GPU)
- \* **Потенциально различная эффективность систем различных классов для решения графовых задач**

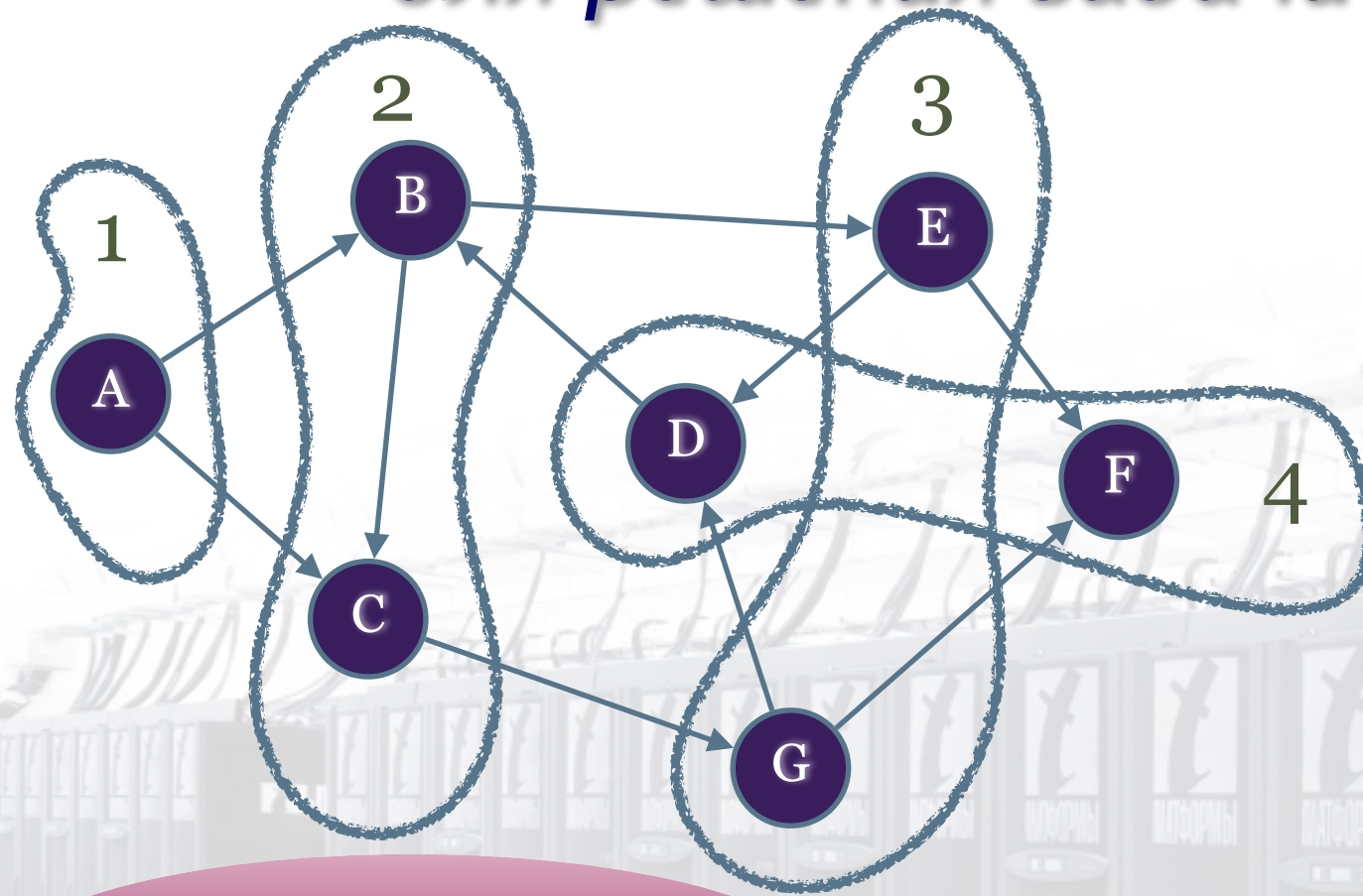
# Метод суперкомпьютерного кодизайна для эффективного решения графовых задач



- \* *Создавать эффективные программы для современных суперкомпьютерных архитектур крайне нетривиально*
- \* *В отличие от традиционного процесса оптимизации алгоритма под заданную целевую архитектуру, мы **выбираем суперкомпьютерные архитектуры**, максимально подходящие для решения конкретной графовой задачи*



# Применение метода суперкомпьютерного кодизайна для решения задачи поиска в ширину



- \* Поиск в ширину - обход вершин графа «по слоям» от заданной вершины-источника
- \* крайне низкая вычислительная интенсивность (ops per byte)
- \* сильная нерегулярность в зависимости от структуры входного графа

Социальные графы

Задача поиска в ширину

Суперкомпьютерный кодизайн

Использование массивно-параллельных архитектур с HBM2 памятью

- V100 GPU (~900 GB/s)
- NEC SX-Aurora TSUBASA (~1.2 TB/s)

Выбор алгоритма, модификаций алгоритмов, форматов хранения графа, оптимизаций

Эффективные реализации графовых алгоритмов

# Применение метода суперкомпьютерного кодизайна для решения задачи поиска в ширину

Социальные графы

## Выбор алгоритма поиска в ширину

- top-down
- bottom-up
- **direction optimizing**

## Выбор формата хранения графа

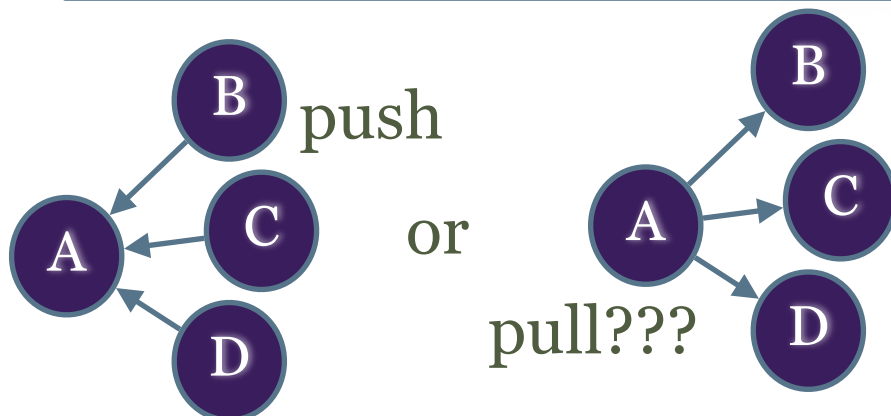
- использование векторного расширения CSR для эффективной векторной обработки данных
- препроцессинг графа для многократных запусков алгоритма
- повышение локальности загрузки данных из подсистемы памяти (кластеризация вершин)
- и др.

## Выбор модификаций алгоритма

- полные/частичные обходы графа на различных итерациях
- push/pull направления обходов на различных итерациях
- и др.

## Микроархитектурные оптимизации

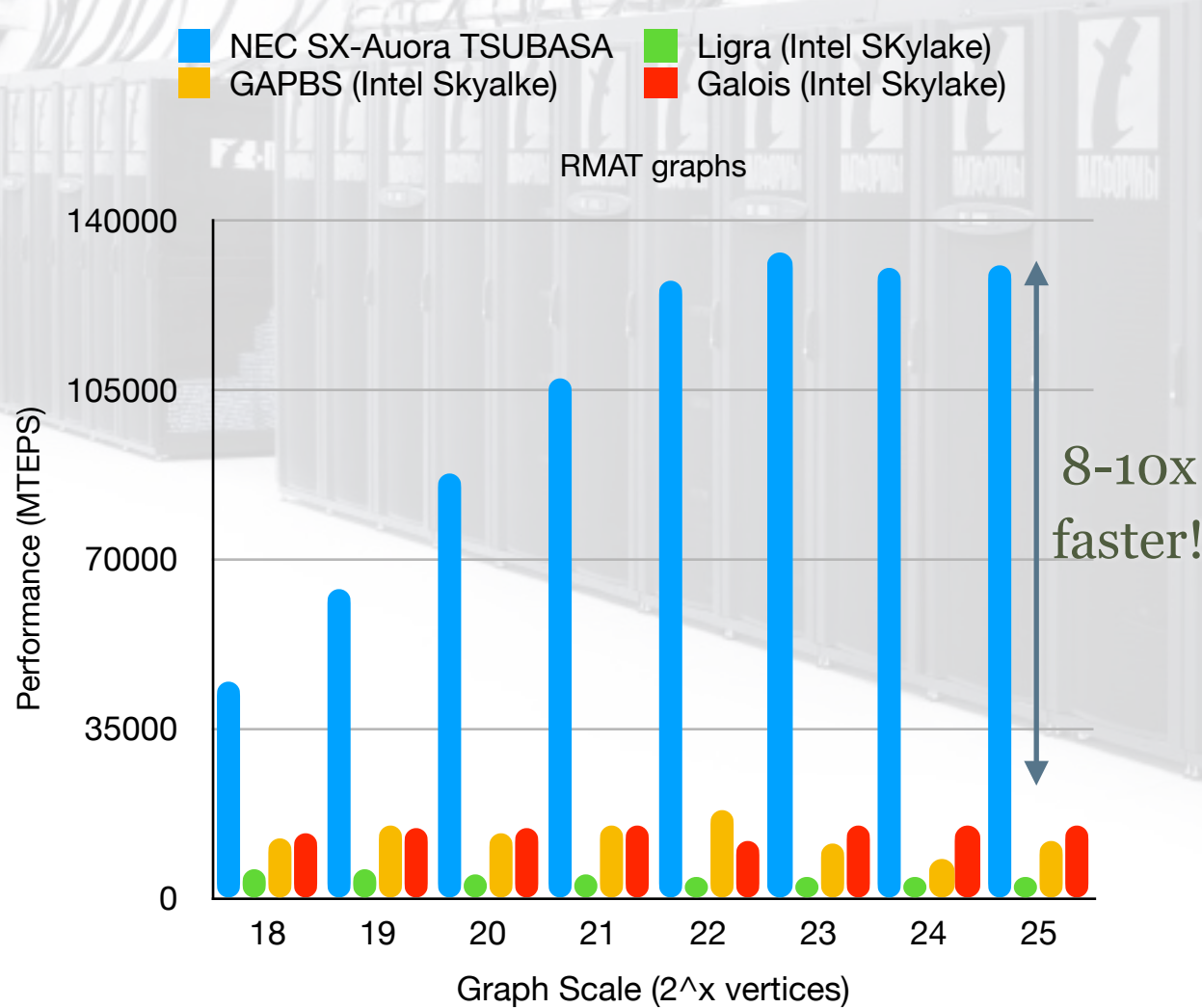
- кэширование hub-вершин (NEC, GPU)
- развертка циклов (NEC)
- использование unified памяти для out-of-core обработки (GPU)
- оптимизация коллективных доступов к подсистеме памяти (NEC, GPU)
- использование динамического параллелизма для балансировки параллельной нагрузки (GPU)
- virtual-warp (GPU)
- и др.



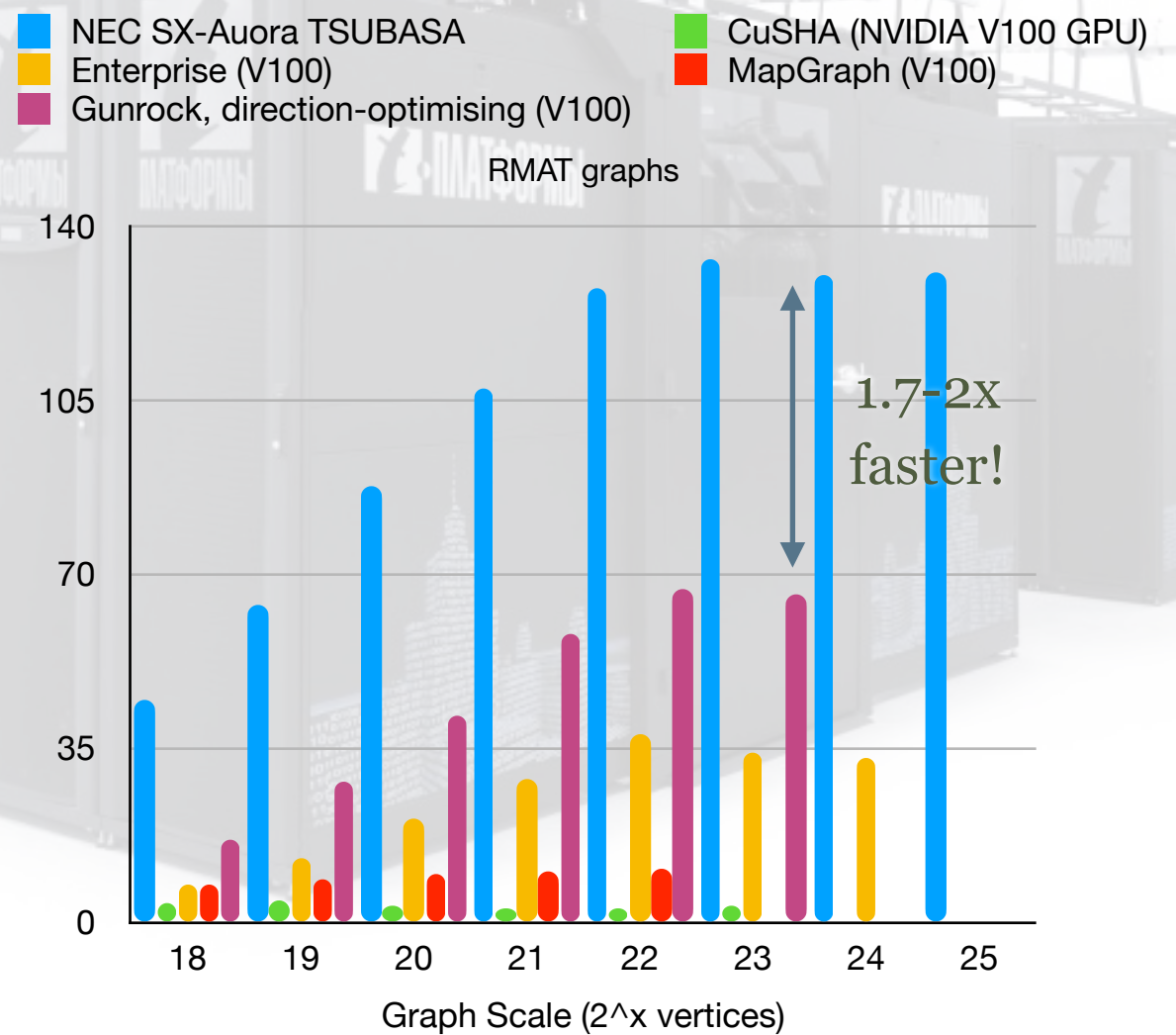


# Сравнение производительности с ведущими мировыми реализациями

- \* На сегодняшний день существует большое количество библиотек и графовых фреймворков, позволяющих решать графовые задачи на современных системах с общей памятью Gunrock, Ligra, Galois, Enterprise и др.
- \* Разработанные на основе принципов суперкомпьютерного кодизаина реализации существенно опережают ведущие мировые библиотечные реализации графовых алгоритмов
- \* Так, использование векторной архитектуры **NEC SX-Aurora TSUBASA с быстрой памятью** позволяет достичь существенного ускорения **при решении задачи поиска в ширину**



Сравнение с реализациями для многоядерных центральных процессоров



Сравнение с реализациями для NVIDIA GPU