

Институт вычислительной математики и
математической геофизики

**ИНСТРУМЕНТАЛЬНАЯ ПОДДЕРЖКА
ОЦЕНКИ МАСШТАБИРУЕМОСТИ
ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ ПО
ВЗАИМОСВЯЗАННОСТИ ДАННЫХ**

докладчик: С. В. Помелов

научный руководитель: И. Н. Скопин

Статистика производительности суперкомпьютеров списка top500



Задача

Создать математическую модель и соответствующую реализацию в виде инструментальной программы, позволяющей получить оценки масштабируемости и эффективные способы распараллеливания алгоритма до разработки и тестирования вычислительного приложения.

Метод выполнения работы

Свойства архитектуры вычислительной системы, такие как скорость ядра вычислений, скорость обмена данными между ядрами и элементами памяти, формализуемы. Алгоритмы вычислений и способы их распараллеливания также формализуемы.

Математически точное, гарантирующее погрешность, с широкой применимостью к реальному многообразию систем и задач предсказание предполагается неразрешимой задачей.

Метод выполнения работы

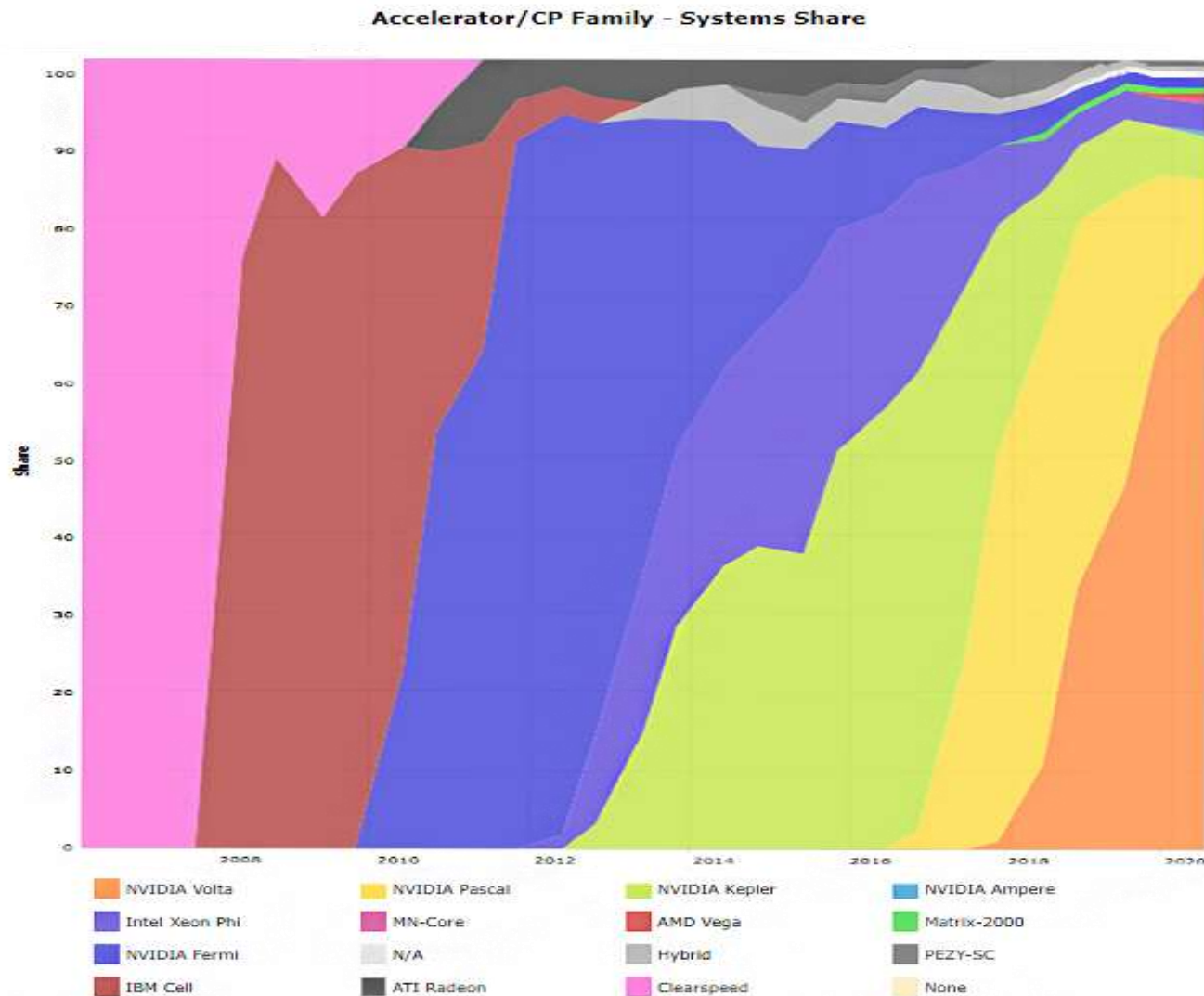
Построение математической модели и реализации приложения, с помощью которого просчитывается взаимодействие параллельным архитектур с вычислительным алгоритмом. С помощью такой симуляции исполнения задачи делается вывод о масштабируемости и свойствах способов разбиения исходных данных для параллельного исполнения на заданной архитектуре.

Метод выполнения работы

Сложность общей задачи нивелируется тем, что большую часть нюансов производительности в модели на практике можно заменить на реальные измерения подпрограмм в масштабе, где время их исполнения имеет небольшие отклонения.

При наличии достаточного числа точек и, в предположении что все возможные существенные факторы действовали в момент измерений, мы можем дать оценку погрешности для подпрограмм-блоков алгоритма и предсказания производительности в целом.

Распределение ускорителей / процессоров



Ряд проблем в суперкомпьютерных вычислениях

- Отставание разработки методов и стратегий параллельного программирования от современных суперкомпьютерных архитектур.
- Возрастающее с ростом числа компонент число аппаратных ошибок суперкомпьютеров. Необходимы как устойчивые к ошибкам аппаратные решения, так и программные решения, способные совершать корректные вычисления в условиях частичных аппаратных сбоев.
- Усложнение параллелизма с условных одного-двух слоёв устройства и программы в начале развития компьютерных технологий, до четырех и более для облачных и кластерных решений.

*«Software challenges for extreme scale computing: Going from petascale to exascale systems», Michael A. Heroux, Sandia National Laboratories.

Место симуляции исполнения алгоритма среди других подходов с точки зрения масштаба процессов исполнения

Уровень ВС из нескольких узлов
PVM, Supercomputer task managers.

Уровень приложений

Уровень простых абстракций
OpenCL, RapidMind, OpenHMPP, Intel TBB.

Уровень прямой аппаратной поддержки и ОС
OpenMP, CUDA, POSIX,
Windows Threads, Boost Threads.

Система отвечает требованиям.



Описание математической модели

Свойства элементов.

- E – Конечное множество элементов архитектуры. Включает в себя два непересекающихся множества элементов памяти M и арифметических элементов A . $M, A \subset E$; $M \cap A = \emptyset$
- P – Конечное множество кортежей свойств элемента архитектуры.
- p – Свойство допустимое для одного или всех типов элементов (M, A) . Свойства, разбитые по допустимости, обозначены на следующих слайдах. Элемент и множество его свойств образуют упорядоченное множество, где на первом месте элемент, а далее идут его упорядоченные свойства $(E \times P)$.

Элементы модели

Элементы памяти, их свойства ($M \subset E$).

- T – Конечное множество типов данных. Список возможных в вычислительном процессе типов данных.
- Свойство: Максимальный объем хранения различных данных (T) или, другими словами, максимальное число значений данного типа в элементе памяти.

$$F: T \rightarrow N.$$

Элементы модели

Элементы памяти, реализация.

```
class MemNode // Memory unit from M
{
private:
    std::string NodeName;
    const unsigned int dataType[cap]; // T
    const unsigned int dataCapacity[cap]; // F
    const unsigned long byteCapacity;
```

Элементы модели

Арифметические элементы ($\mathbb{A} \subseteq \mathbb{E}$).

- \mathbf{O} – Конечное множество типов операций. Список возможных в вычислительном процессе операций.
- $\mathbf{O} \times (\mathbf{T} \times \mathbf{N})$ – Множество n-арных операций на арифметическом элементе.
- Свойство 1: $A_{p1} \subseteq \mathbf{O} \times (\mathbf{T} \times \mathbf{N})$ – множество операций доступных данному арифметическому элементу.
- Свойство 2: A_{p2} – скорость операций в условных или реальных единицах.

Элементы модели

Связи (L).

- $L\{(e' \times P_{e'}, e'' \times P_{e''}) \mid e', e'' \in E; P_{e'}, P_{e''} \in P\}$ – Конечное множество связей в архитектуре. Связи могут быть между любыми элементами архитектуры из E.
- $T \times N$ – Подходящие для передачи по данной связи блоки размера из значений соответствующих им типов данных).
- Свойство: $F : T \times N \rightarrow N$ – скорость передачи для различных абстрактных типов (некоторого числа значений их представителей) между парой элементов архитектуры (e', e'') .

Представление архитектуры

В совокупности \mathbb{L} может быть представлено как множество дуг направленного связного мультиграфа с вершинами из \mathbb{E} . Множество связей и является полным описанием архитектуры в модели.

Представление алгоритма

1. Свойства операций алгоритма.

- E – Конечное множество операций. Включает в себя два непересекающихся множества операций передачи данных M и локальных операций над данными A .

2. Операции передачи данных (M).

- T – Конечное множество типов данных. Эквивалентно множеству из модели архитектур.
- Свойство: d – Объем передачи данных t .

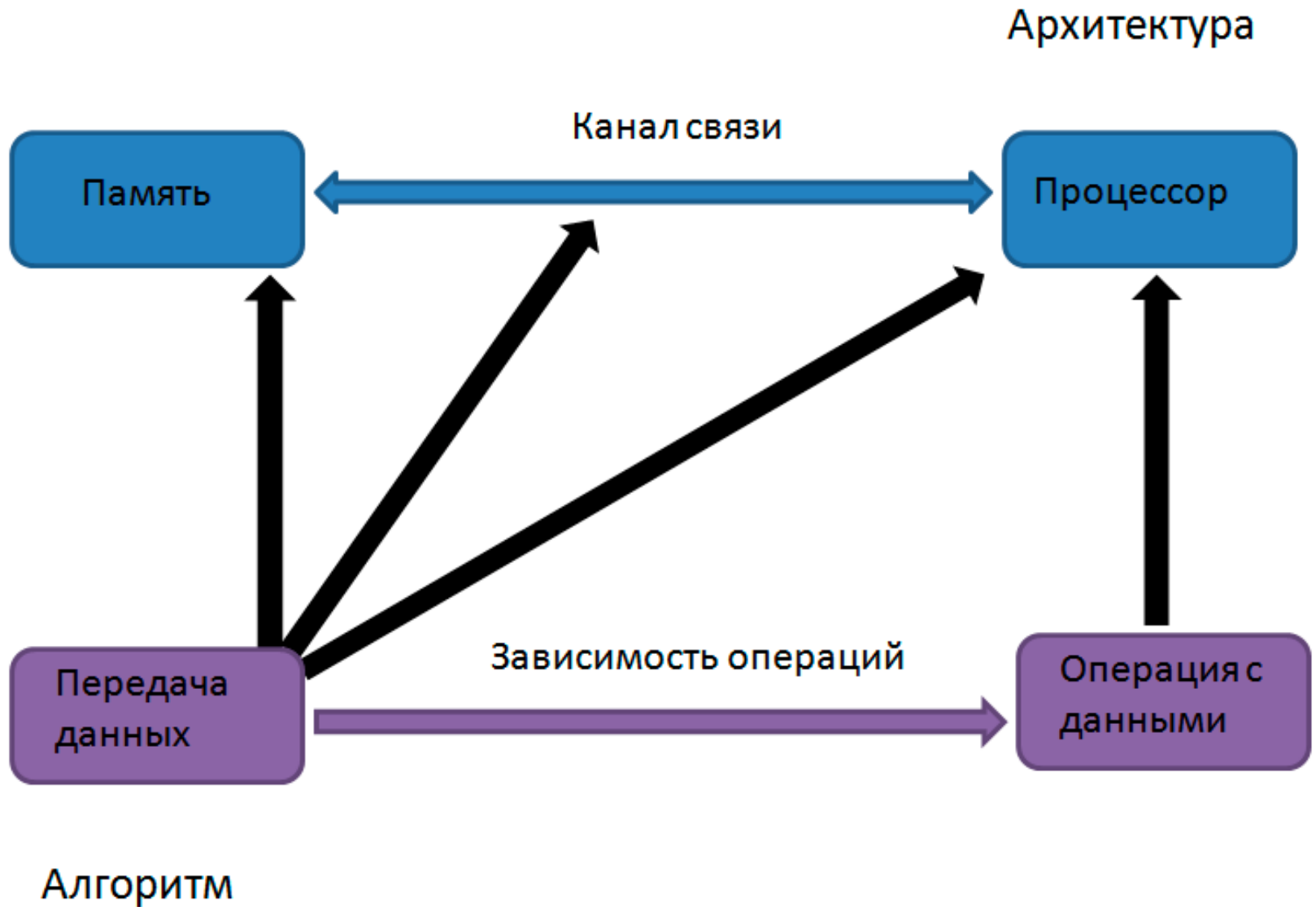
3. Операции над данными (A).

- O – Конечное множество типов операций. Эквивалентно множеству из модели архитектур.
- Свойство: a из $O \times (T \times N)$ – n -арная операция алгоритма с указанием типов и объемов данных.

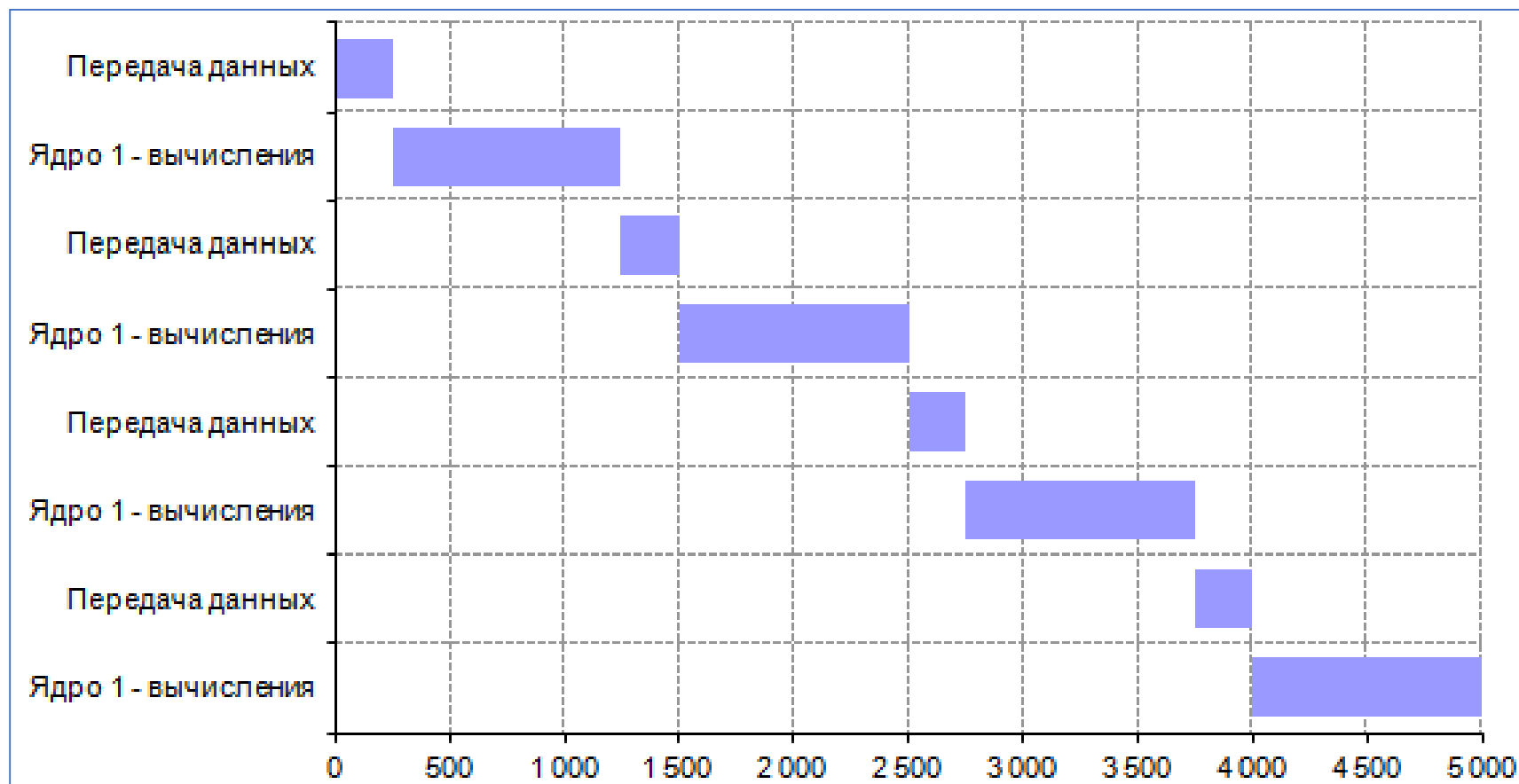
4. Связи между операциями (L).

- L – Конечное множество связей. Связи могут быть между любыми операциями. Множество связей является описанием архитектуры. Нет свойств.

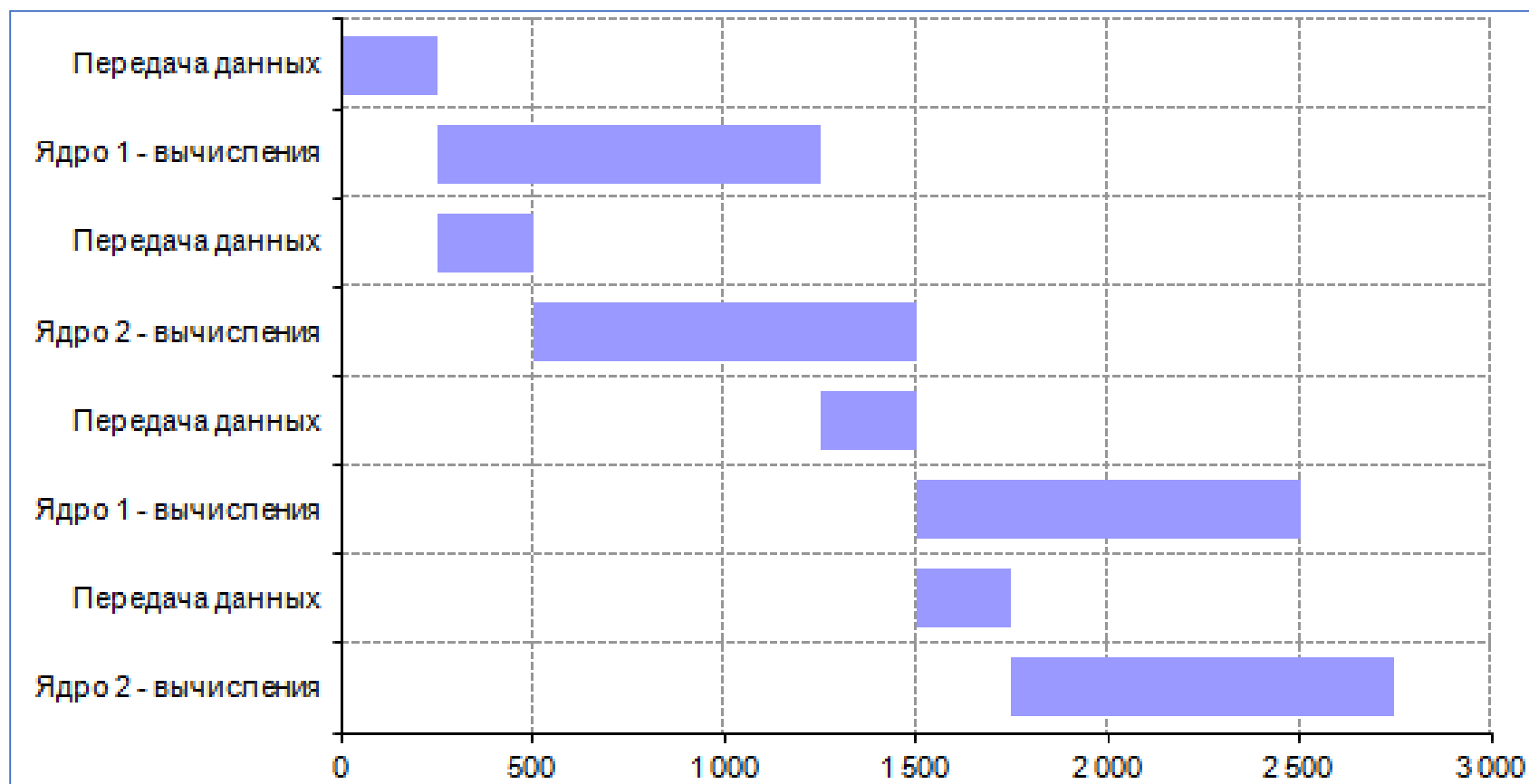
Схема симуляции



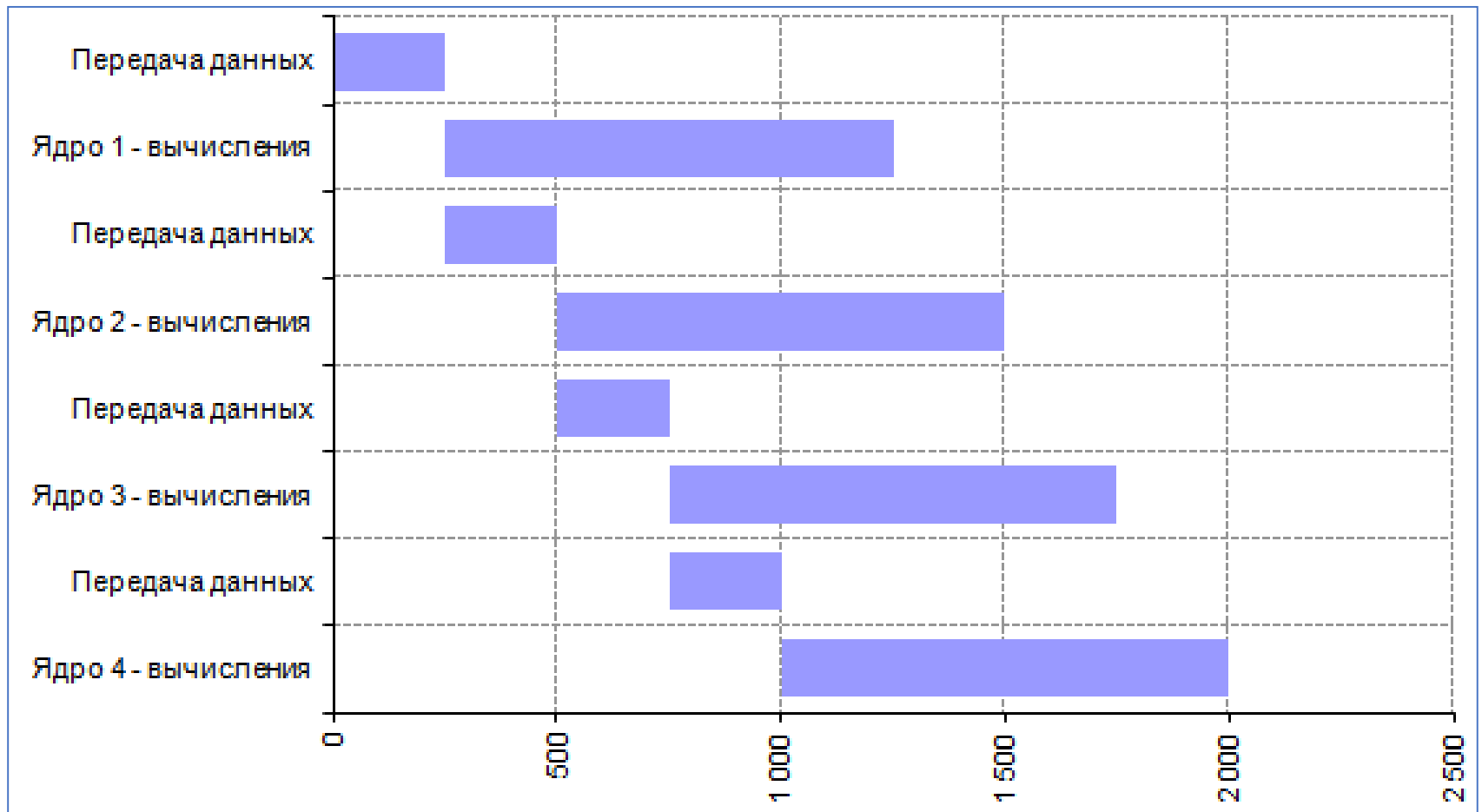
Журнал симуляции в виде диаграмм Ганта



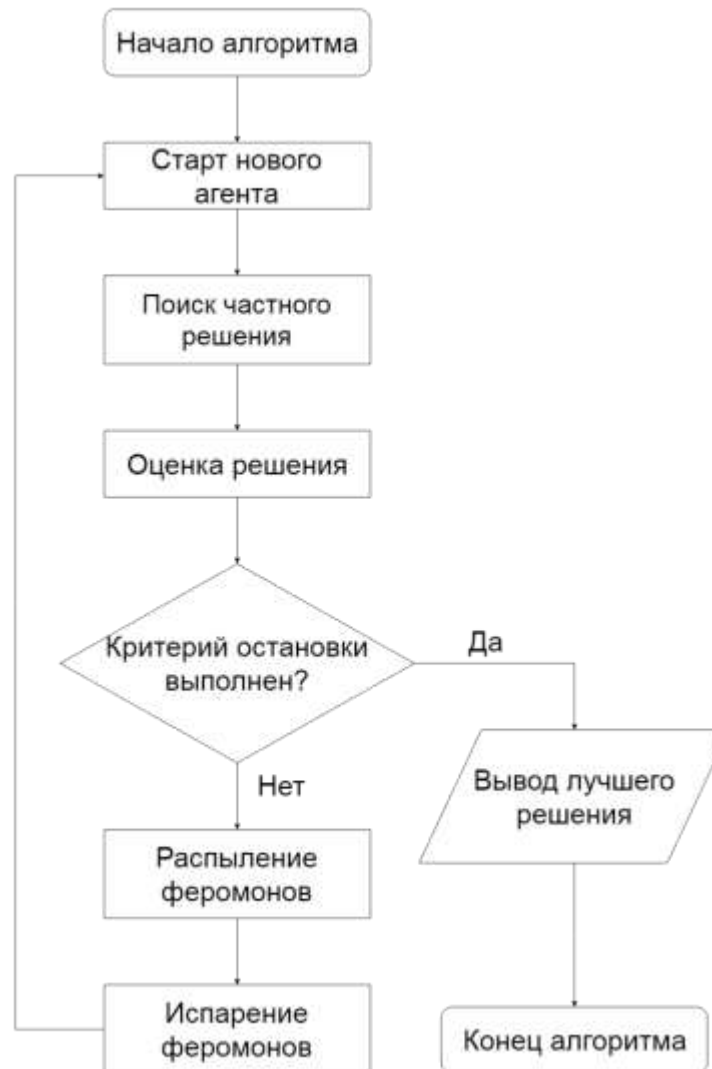
Журнал симуляции в виде диаграмм Ганта



Журнал симуляции в виде диаграмм Ганта



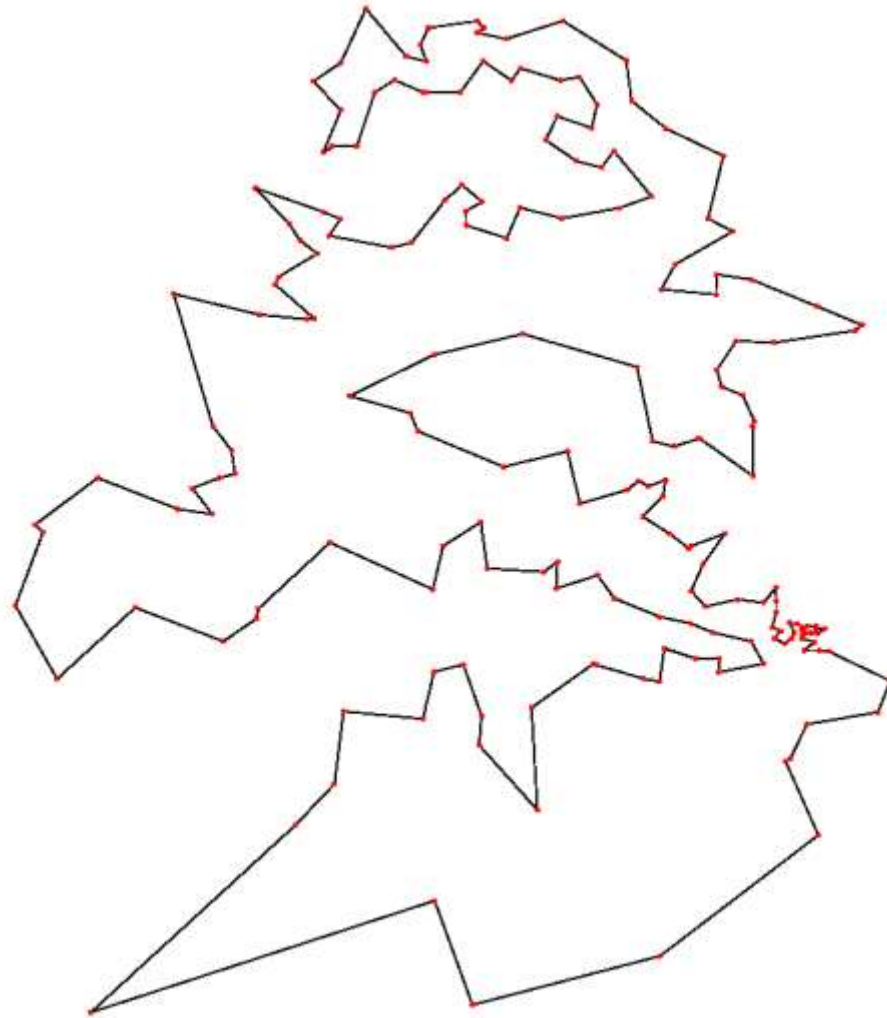
АСО, логика изолированного агента



Задача коммивояжера для АСО



Задача коммивояжера для АСО



Настройки параллельного АСО

Число потоков колоний.	от 1 до 4 с шагом 1
Число параллельных потоков агентов в каждой колонии.	от 1 до 4 с шагом 1
Частота попыток отсылки решений.	раз в секунду
Топология связей колоний.	все со всеми
Время запуска.	10 минут
Число запусков для получения среднего значения измерения и расчёта погрешности.	10
Начальное значение феромонов.	1.0
Коэффициент испарения.	0.1

Тестовые стенды

	T9600	i5-2300
Операционная система.	Ubuntu 14.04, 64 битная.	Microsoft Windows 7 Professional 6.1. 64, битная.
Процессор.	Intel Core 2 Duo.	Intel Core i5-2300.
Оперативная память.	SK Hynix HYMP125S64CP8, в режиме 333 МГц, двойной канал.	AMD AE32G1609U1, в режиме 622 МГц, двойной канал.
Исполняемый код.	Oracle JVM, Java -server compiler 1.8.0_60, JDK 8u77.	

Вывод программы для Ants Colony Optimization (ACO) алгоритма и журнал исполнения

ALGORITHM:

```
|—— | readAndPrepareData-READ_DATA TSPData {FOUR_B_FL*37636}| ...
|   |—— | colonyRuleOverhead-OVERHEAD TSPData {UNKNOWN*1}
|     |—— | systemOverhead-SYSTEM_OVERHEAD TSPData {UNKNOWN*1} | antSolutionGeneration-
ANT_SOLUTION_GENERATION TSPData {FOUR_B_FL*194} | ...
|       |—— | colonyRuleOverhead-OVERHEAD TSPData {UNKNOWN*1}
|         |—— | systemOverhead-SYSTEM_OVERHEAD TSPData {UNKNOWN*1} | antSolutionGeneration-
ANT_SOLUTION_GENERATION TSPData {FOUR_B_FL*194} | antColonyInteraction-INTERACTION TSPData
{FOUR_B_FL*37636}
```

ARCHITECTURE:

Link – bus 4000b: t=1000

In: DDR2: 1073741824b

Out: Core1: readData (TSPGraph {FOUR_B_FL*37636}) t=1; ...

Timings:

calculation: Core1 : readAndPrepareData-READ_DATA TSPData {FOUR_B_FL*37636} 0ns->1ns(0)%

calculation: Core1 : preCalculations-PRE_CALCULATIONS TSPData {FOUR_B_FL*37636} 1ns->2ns(0)%

wait: Core2 0ns->2ns(0)%

calculation: Core1 : colonyRuleOverhead-OVERHEAD TSPData {UNKNOWN*1} 2ns->~27.73ms(0.001)%

calculation: Core2 : colonyRuleOverhead-OVERHEAD TSPData {UNKNOWN*1} 2ns->~27.73ms(0.001)%

calculation: Core1 : systemOverhead-SYSTEM_OVERHEAD TSPData {UNKNOWN*1} ~27.73ms->~116.45s(4.3461)%

calculation: Core1 : antSolutionGeneration-ANT_SOLUTION_GENERATION TSPData {FOUR_B_FL*194} ~116.45s->~2501.26s(89.0244)%

calculation: Core1 : antColonyInteraction-INTERACTION TSPData {FOUR_B_FL*37636} ~2501.26s->~2678.82s(6.6286)%

calculation: Core2 : systemOverhead-SYSTEM_OVERHEAD TSPData {UNKNOWN*1} ~27.73ms->~116.45s(4.3461)%

calculation: Core2 : antSolutionGeneration-ANT_SOLUTION_GENERATION TSPData {FOUR_B_FL*194} ~116.45s->~2501.26s(89.0244)%

calculation: Core2 : antColonyInteraction-INTERACTION TSPData {FOUR_B_FL*37636} ~2501.26s->~2678.82s(6.6286)%

Timings:

Core2 – ~2678.82s

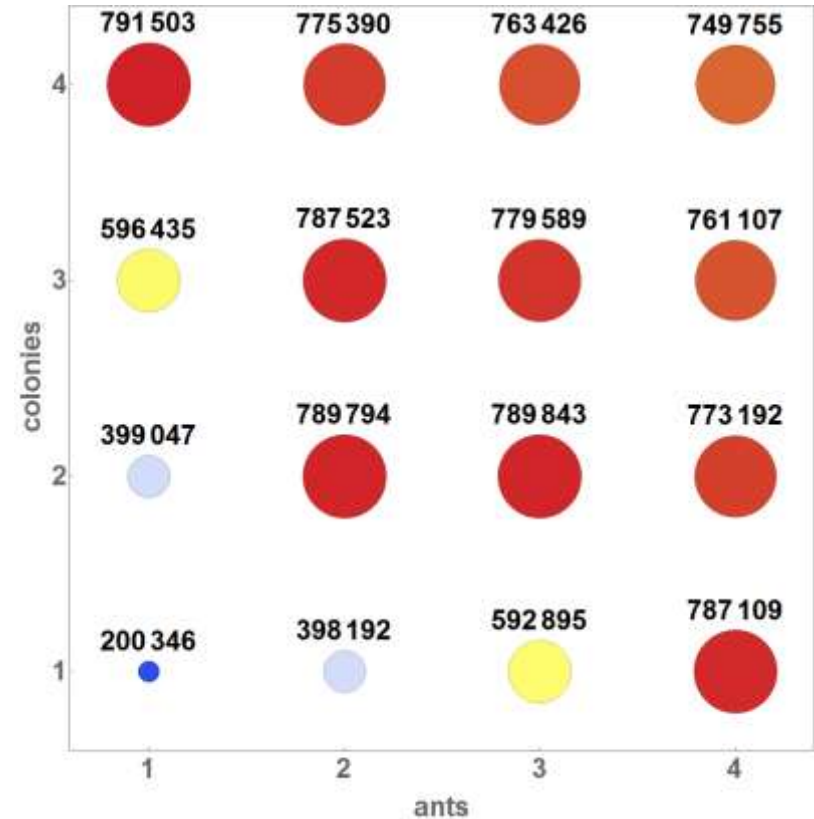
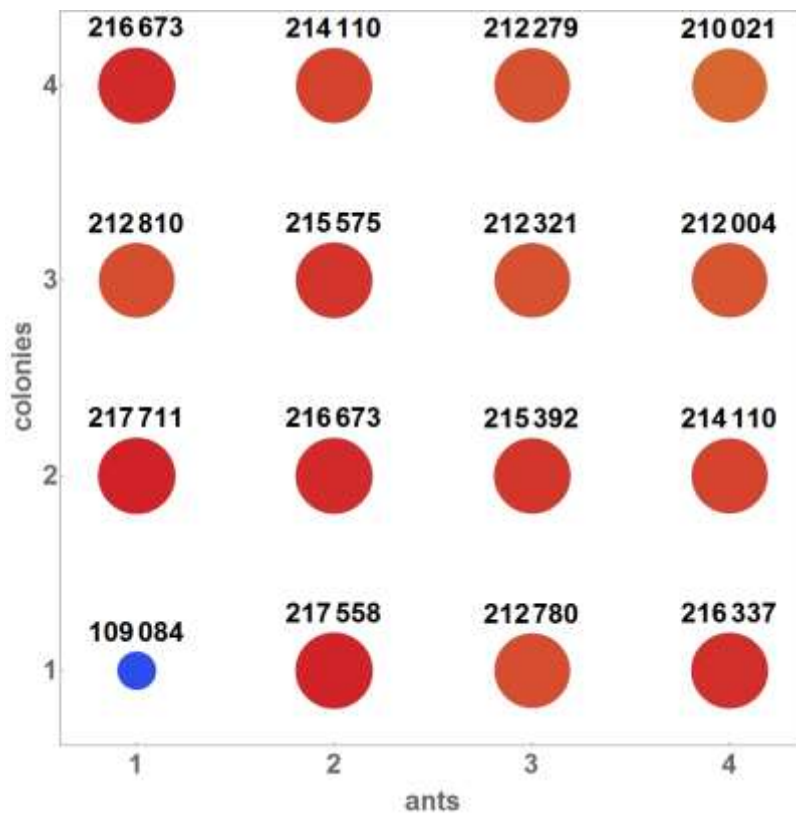
Core1 – ~2678.82s

Total time: ~2678.82s

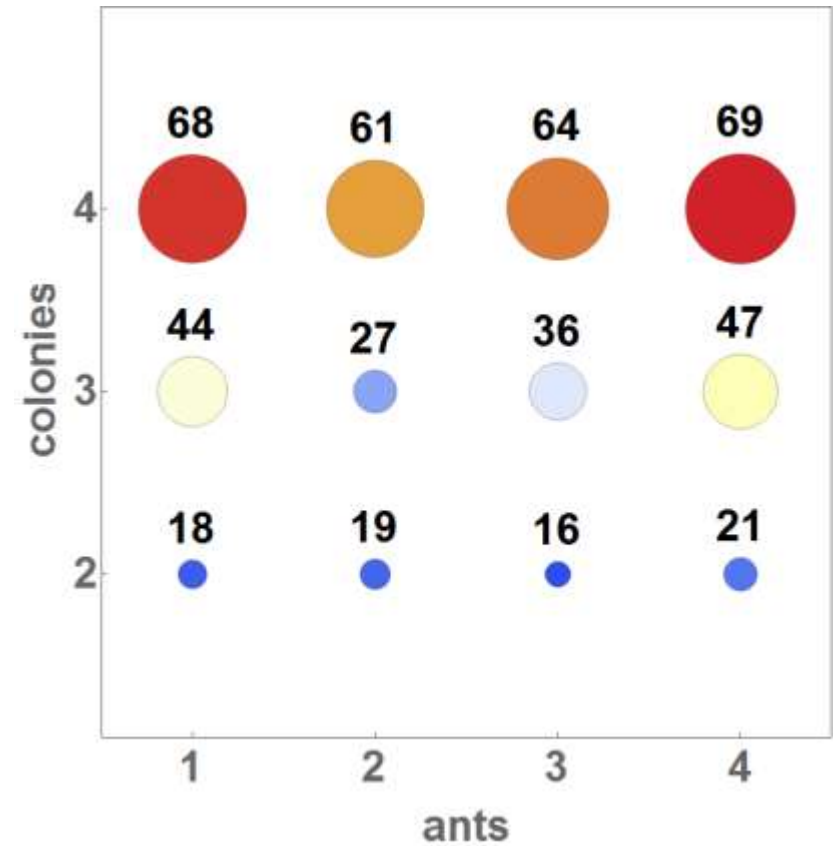
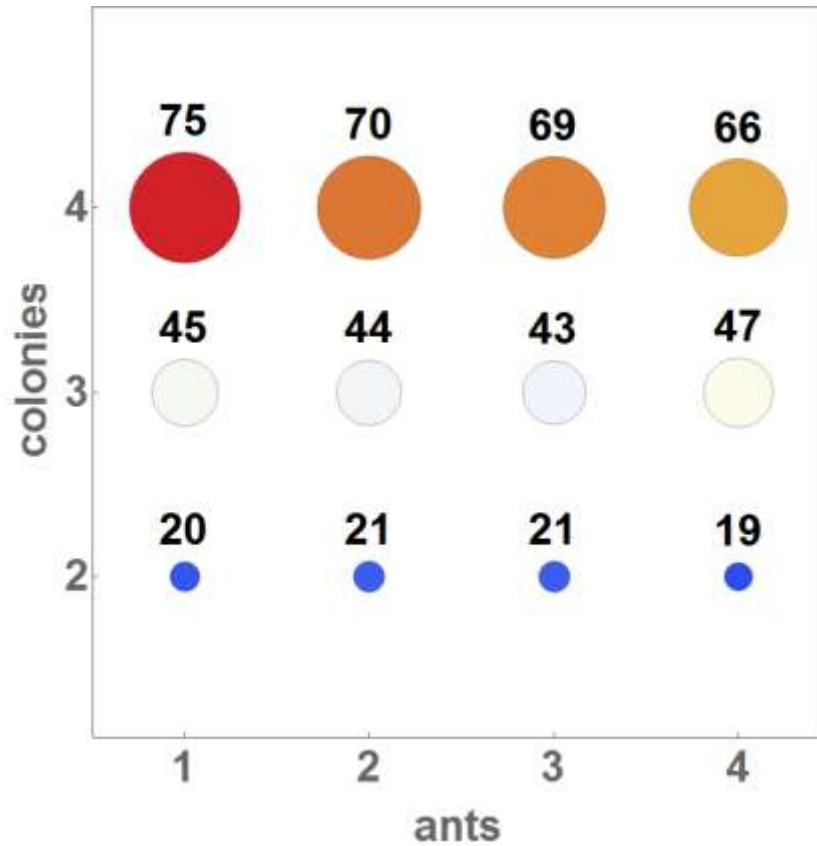
Журнал АСО в виде таблицы

Исполнитель	Операция	Начало операции	Конец операции (с)	Затрачено ресурсов (%)
Core1	SYSTEM_OVERHEAD	27.73 мс	116.45	4.34
Core1	ANTS_SOLUTIONS_GENERATION	116.45 с	2501.26	89.02
Core1	INTERACTION	2501.26 с	2678.82	6.62
Core2	SYSTEM_OVERHEAD	27.73 мс	116.45	4.34
Core2	ANTS_SOLUTIONS_GENERATION	116.45 с	2501.26	89.02
Core2	INTERACTION	2501.26 с	2678.82	6.62

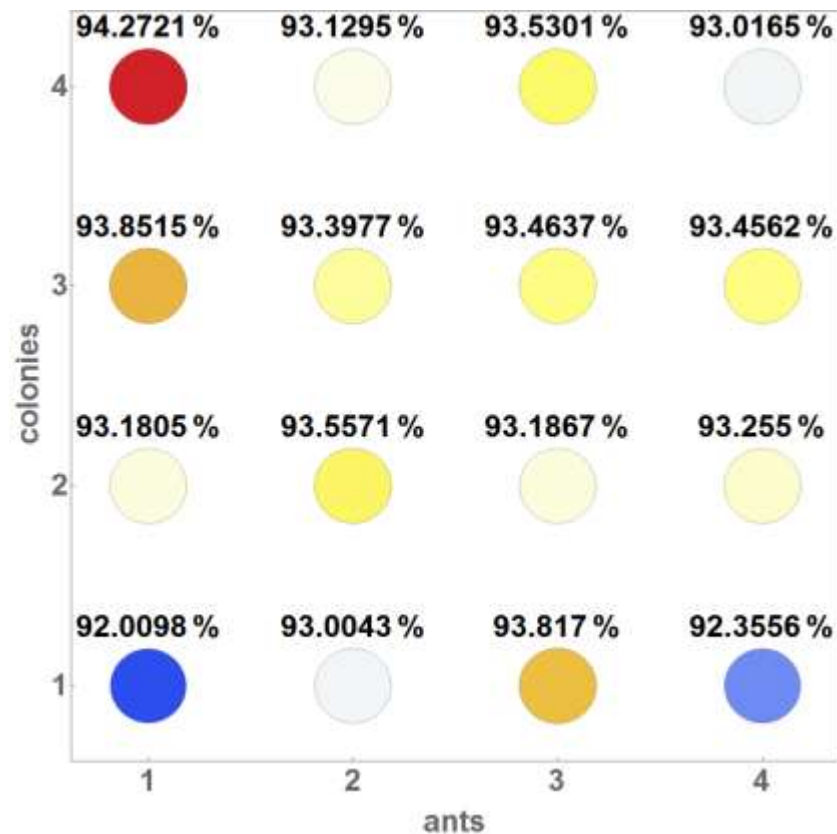
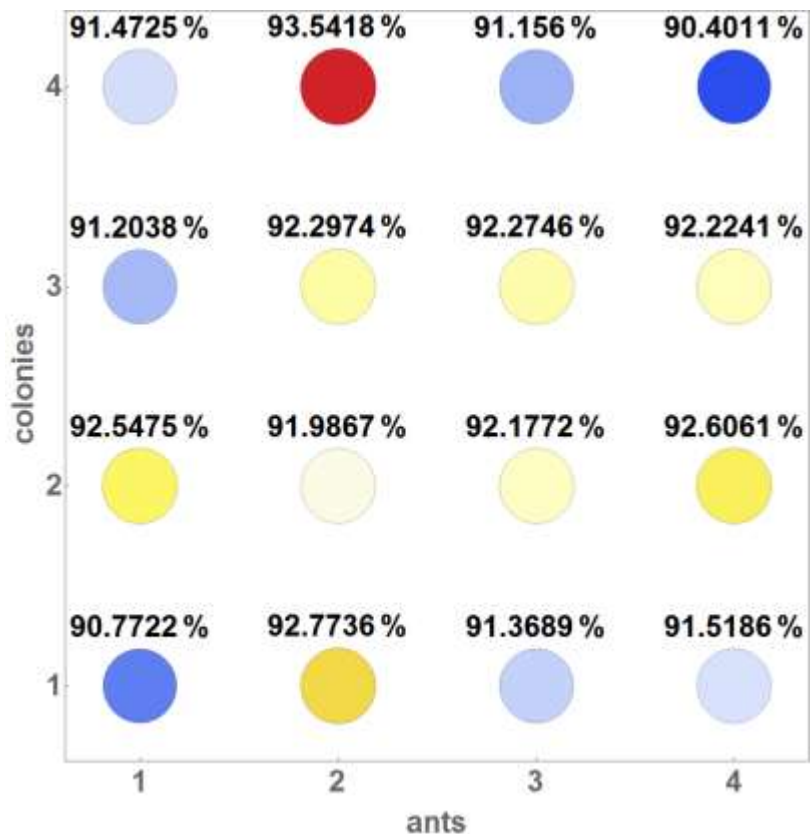
Симуляция числа запусков агентов для 2х и 4х ядерной конфигурации



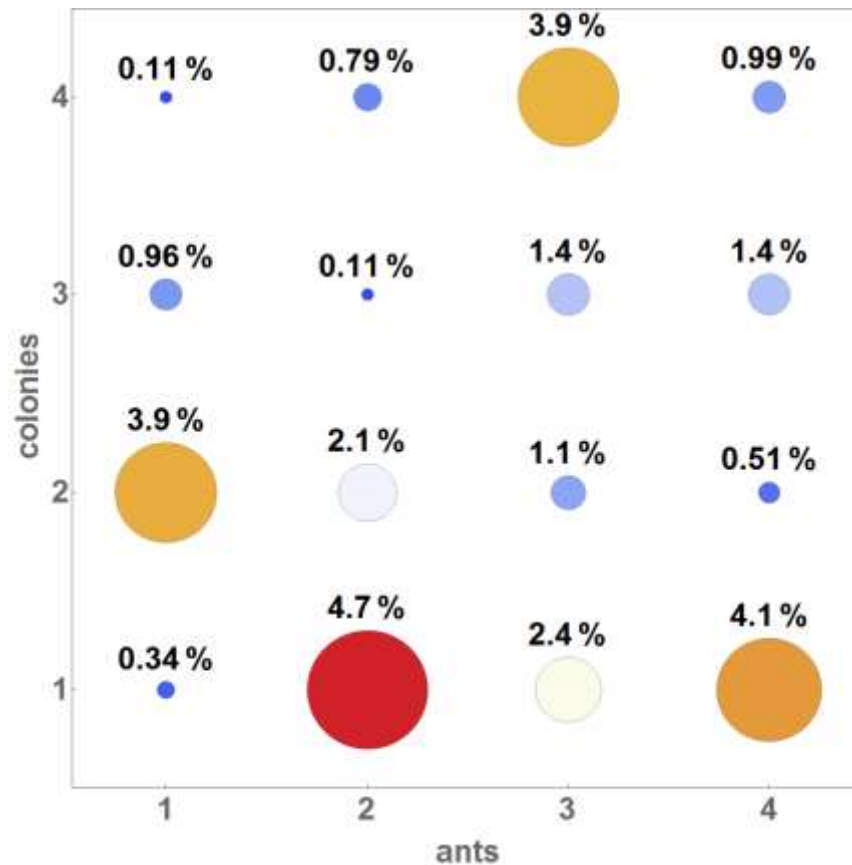
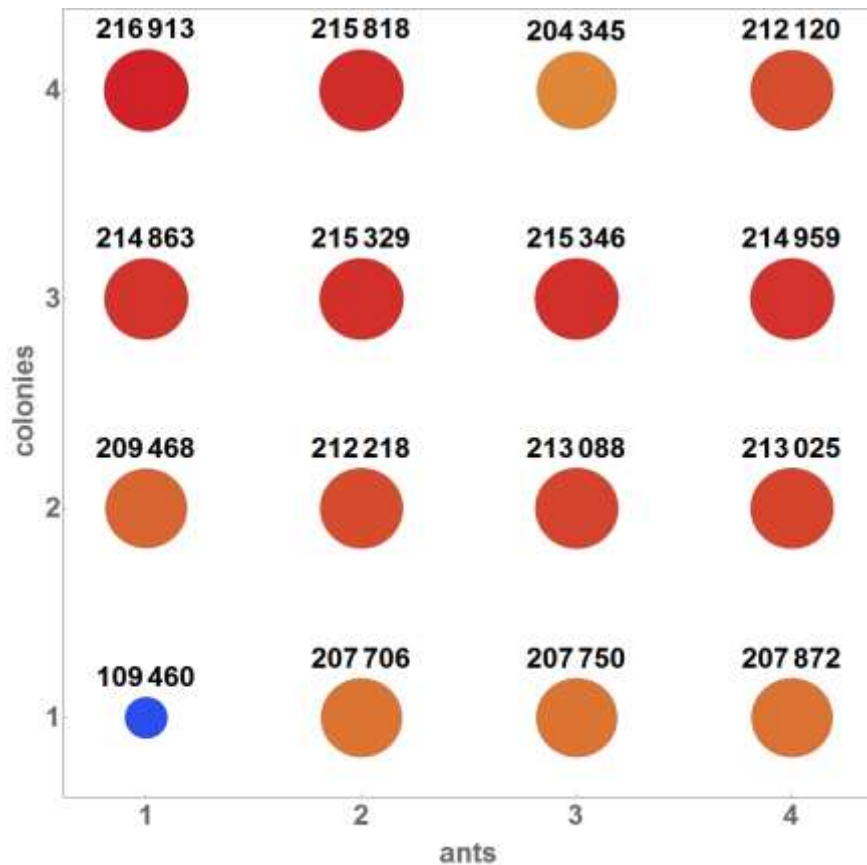
Измеренное число обменов данными для 2х и 4х ядерной конфигурации



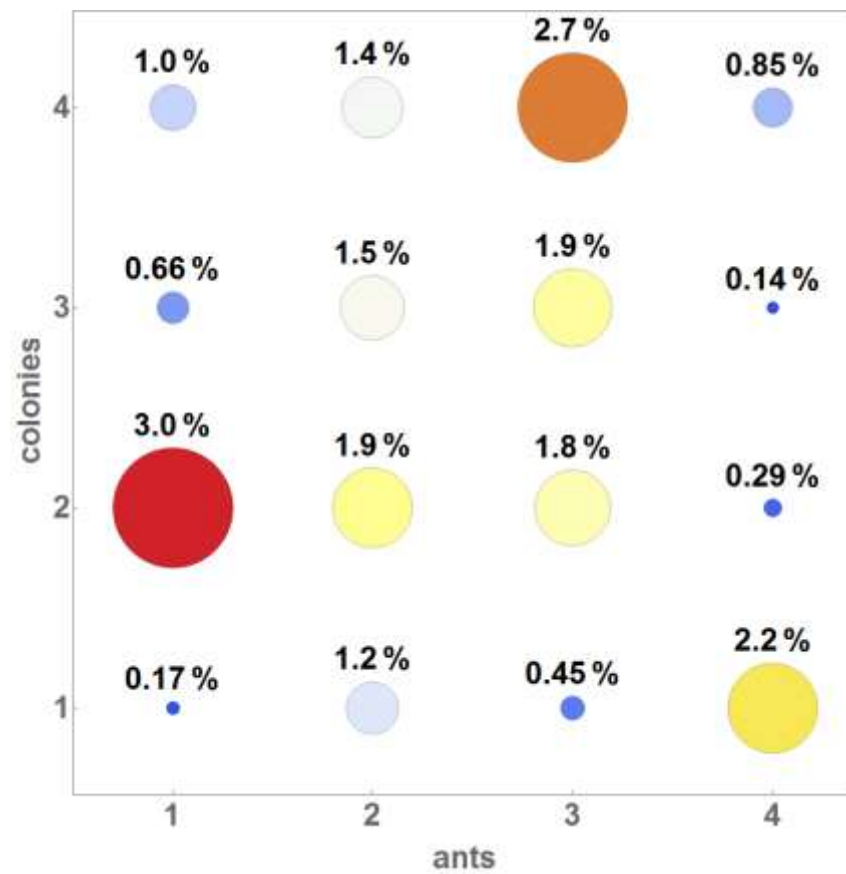
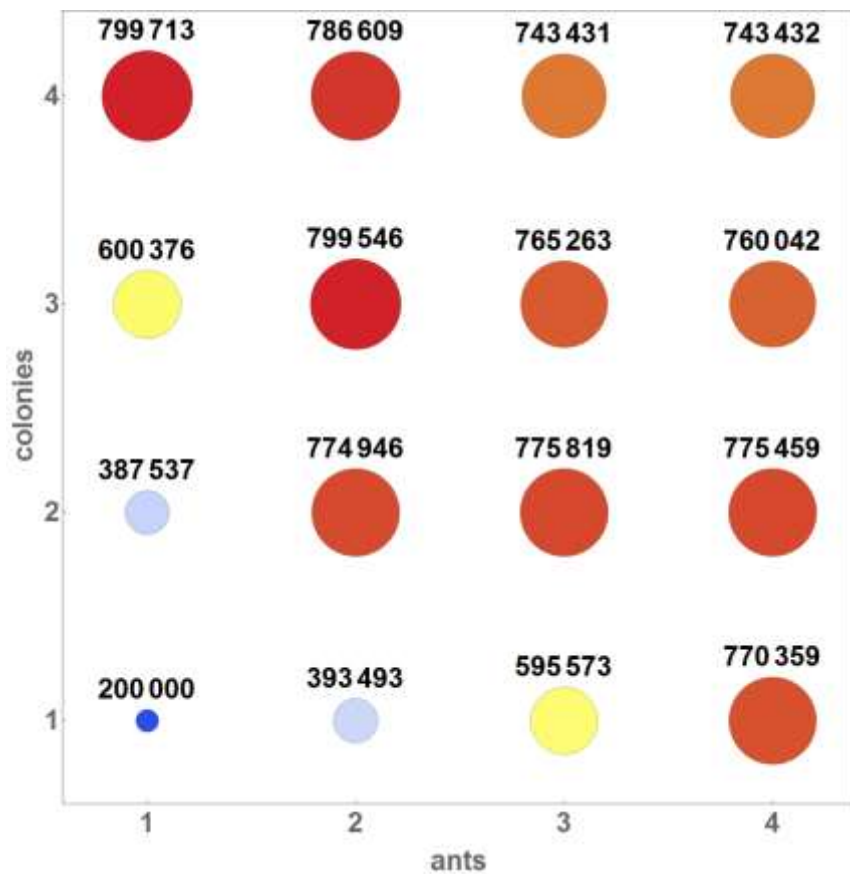
Измеренная точность для 2х и 4х ядерной конфигурации



Измеренное число запусков агентов и отклонения от предсказанного симуляцией значений, округлённые до второго знака. Для двудерной конфигурации.



Для четырёх ядерной конфигурации



Анализ предсказания производительности АСО

50% процентиль, макс. ошибка.	1.2%.
75% процентиль, макс. ошибка.	2.1%.
Макс. ошибка.	4.7%.
Средняя погрешность измерения операций.	1.42%.
Разброс измерений числа запусков агентов.	1.57%.

Результаты работы

1.1 Теоретическая работа.

- Проанализированы источники, включая научные статьи, учебники, материалы конференций, авторефераты и техническую документацию. Результаты включены в литературный обзор диссертации.
- Собрана информация об исследованиях в смежных областях, результаты её анализа описаны в первой главе диссертации под названием «текущее состояние параллельных вычислений на суперкомпьютерах».

Результаты работы

1.2 Теоретическая работа.

- Проанализированы и описаны актуальные исследования в области: лаборатории информатики и искусственного интеллекта MIT по анализу графа потока исполнения, и работа по оптимизации на основе Q-детерминанта кафедры системного программирования ЮУРГУ и ряд других работ.
- Формализована в главах диссертации № 4 и 5 математическая модель архитектуры и алгоритма, легшие в основу практической части работы.

Результаты работы

2.1 Экспериментальная работа.

- Реализована программа для симуляции исполнения алгоритма на архитектуре вычислительной машины.
- Проведено тестирование приложения на возможность отображения ими необходимых свойств архитектуры и алгоритмов.
- С помощью профилировщика произведены замеры реального и сравнение с теоретическим временем работы модификаций алгоритма на персональном компьютере при различных входных параметрах программы.

Результаты работы

2.2 Экспериментальная работа.

- Реализован и описан процесс выбора наиболее эффективной архитектуры для одной из параллельных модификаций алгоритма «муравьиной колонии».
- В форме эксперимента оптимизирован для конкретной вычислительной машины алгоритм выделения подобластей нерегулярной сетки.
- Получены выводы касательно корректности, области применимости математической модели, а также касательно дальнейшего развития приложения.

Основные результаты

- Разработана математическая модель архитектуры и алгоритма.
- Создано вычислительное приложение для оптимизации параллельных алгоритмов.
- Получены результаты оптимизаций для частных случаев алгоритмов муравьиной колонии и декомпозиции вычислительной сетки.

Публикации

Тезисы и доклады:

- Помелов С. Параллельные вычисления: симуляции исполнения задачи на заданной архитектуре // ББК 22.1 Современные проблемы математики и её прикладные аспекты–2013: сборник тезисов научно-практической конференции (Пермь, 29–31 октября 2013 г.) / гл. ред. ВИ Яковлев; Пермский государственный национальный исследовательский университет. – С. 39. ISBN: 978-5-7944-2215-3.
- Pomelov S. An algorithm for constructing a partition of the computational area with irregular grid // The strategies of modern science development: VIII international scientific-practical conference. – North Charleston. – США. – 7-8 Апрель . – 2015. С. 10-12. ISBN: 978-1–512-16853-2.

Гос. регистрация программы для ЭВМ:

- Помелов С. Свидетельство о регистрации ПрЭВМ «Симулятор взаимодействия алгоритма и ЭВМ (СВАЭ)» № 2015618314 от 5 августа 2015 года. Роспатент. Автор и правообладатель: Помелов С. В.

Публикации

Статьи:

- Помелов С. В. Масштабирование алгоритма оптимизации муравьиной колонии для заданной ЭВМ с помощью симуляции исполнения. // Информационно-вычислительные технологии и их приложения. XIX Международная научно-практическая конференция. Под общей редакцией А.Н.Кошева. 2015. С. 52-55.
- Pomelov S. Parallel programming: simulation of the algorithm execution on a given architecture // Prikladnaya Informatika — Journal of Applied Informatics. — 2016. — Т. 11. — №. 6 (66). — С. 70–83.
- Помелов С.В. СИМУЛЯЦИИ ИСПОЛНЕНИЯ И МАСШТАБИРОВАНИЕ АЛГОРИТМА ОПТИМИЗАЦИИ МУРАВЬИНОЙ КОЛОНИИ ДЛЯ ЗАДАННОЙ ЭВМ Том: 75 Номер: 1 Год: 2019 Страницы: 42-49 УДК: 004.942 ЖУРНАЛ: СИСТЕМЫ УПРАВЛЕНИЯ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ Издательство: ООО "Издательство "Научная книга" (Воронеж) ISSN: 1729-5068.
- Pomelov S., Skopin I. A Performance Prediction Method Showcase: Ant Colony Optimization Scaling for a Particular Computer System. // Test Engineering & Managements. — Май–Июнь 2020.— С. 27330–27350. ISSN: 0193-4120.

Дальнейшее направление работы

- Апробация полученных в результате экспериментов результатов касательно оптимизации алгоритмов и подбора облачных конфигураций в AWS.
- Публикация на тему «Convolutional Neural Networks (CNN) Algorithm Performance Prediction and Scaling in Amazon Web Services (AWS) EC2 M5 Instances Cloud Environment» («Предсказание производительности CNN в облачном окружении AWS EC2 из виртуальных машин типа M5») в Scopus.

Спасибо за внимание