Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра системного программирования

Автоматизация разработки параллельных программ с помощью библиотеки легковесных нитей пользовательского уровня

Отчёт за осенний семестр 2024 года

Рябыкин Владислав Александрович, 427 группа

Научный руководитель:

Катаев Никита Андреевич

Москва, 2024

План работы

- Реализовать операцию редукции с использованием Argobots;
- Проверить корректность и эффективность полученной реализации. Сравнить с аналогами. Добиться увеличения эффективности;
- Изучить статьи по планировщикам задач (scheduler) и стратегиям планирования;
- Реализовать планировщик задач с использованием Argobots; (2025 год)
- Добиться увеличения эффективности программ с использованием полученной реализации. (2025 год)

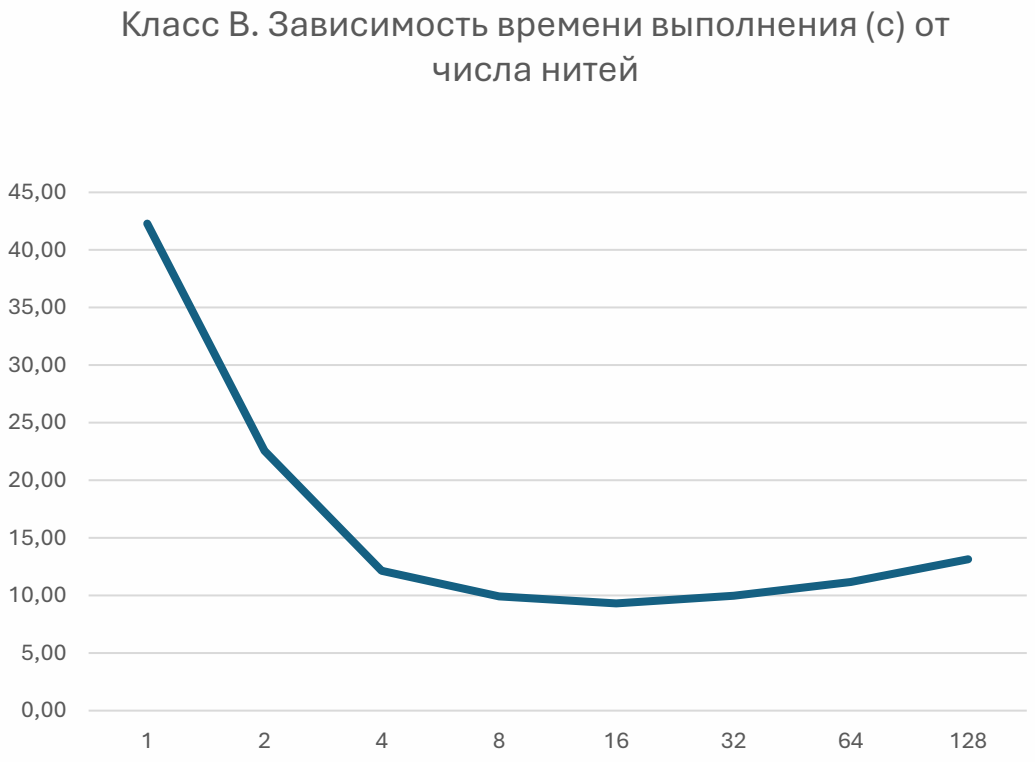
Было сделано ранее

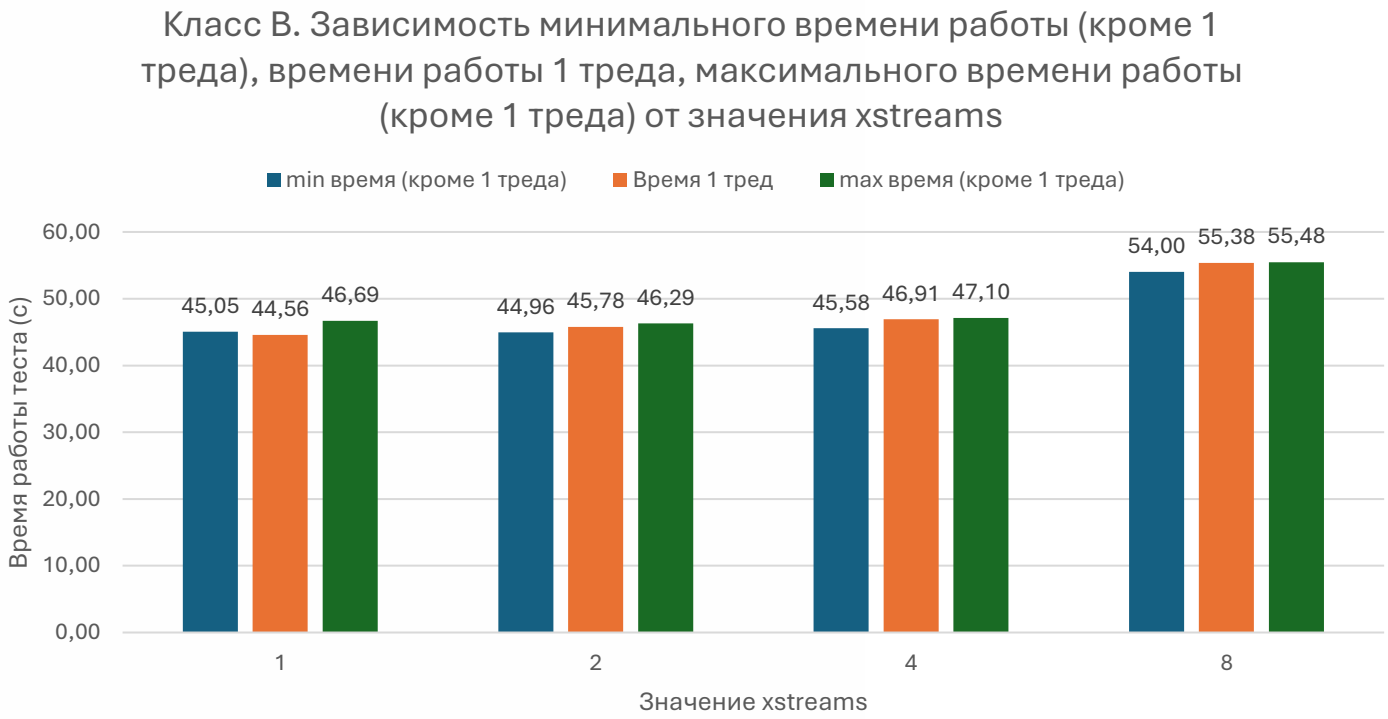
1. Реализована операция редукции;
2. Реализован тест NAS CG;
3. Тест завершается корректно;
4. Замерено время работы теста с различными параметрами;
5. Собрана статистика и построены графики.

Ранее полученные результаты запуска NAS CG

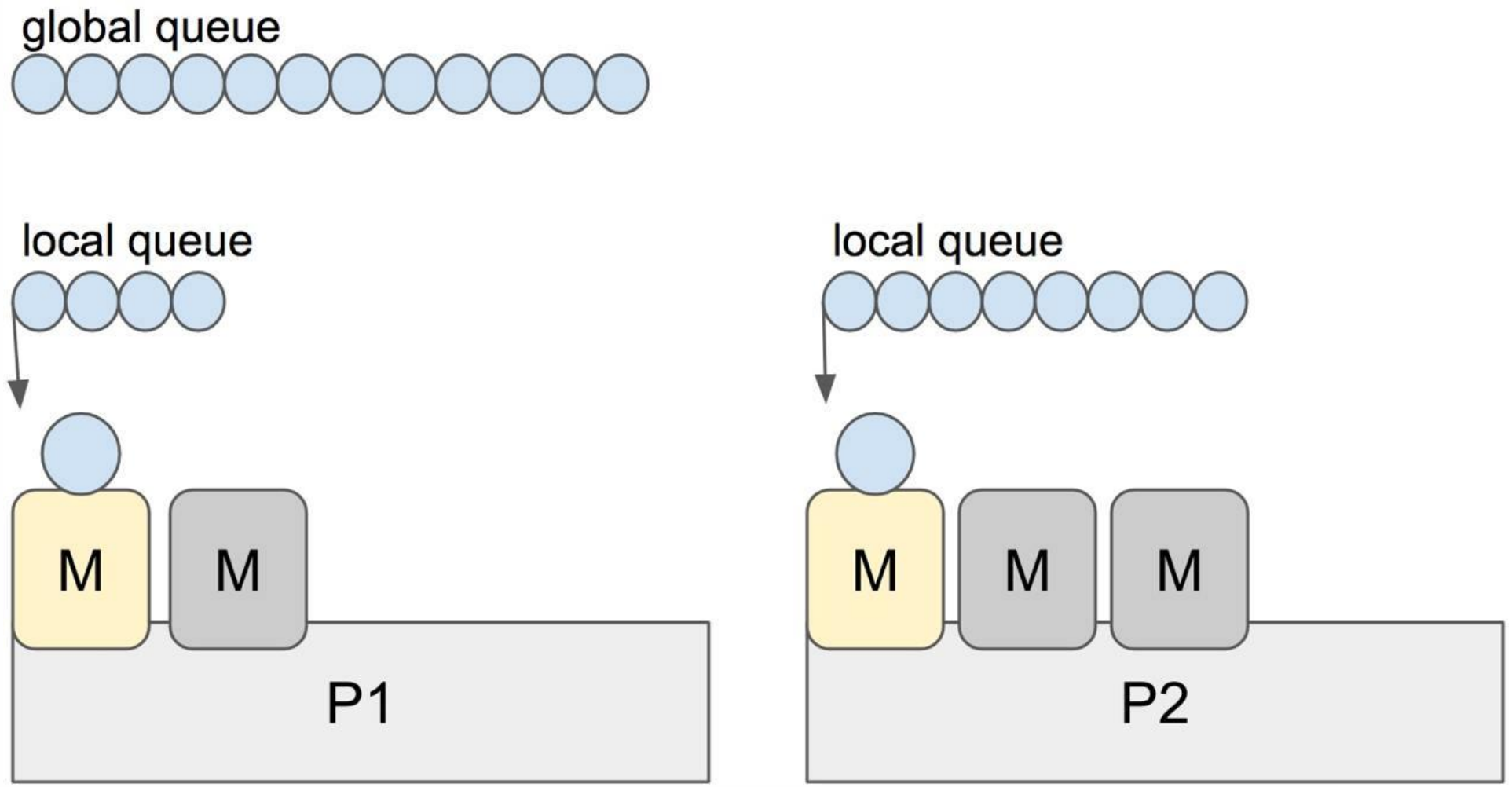
Класс B. 75000 элементов.

Параллельная версия на Argobots работает дольше последовательной.





Планировщик задач (scheduler, скедулер)



Упрощённая схема планировщика языка Golang

Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing

- Хорошо формализованное описание различных статических алгоритмов;
- Описаны собственные алгоритмы: HEFT (Heterogeneous Earliest-Finish-Time) и CPOP (Critical-Path-on-a-Processor);
- Сравнение полученных алгоритмов с аналогами сразу по нескольким метрикам;
- Показано превосходство HEFT;

❖ Интересный подход – генерация тестовых данных по набору параметров;

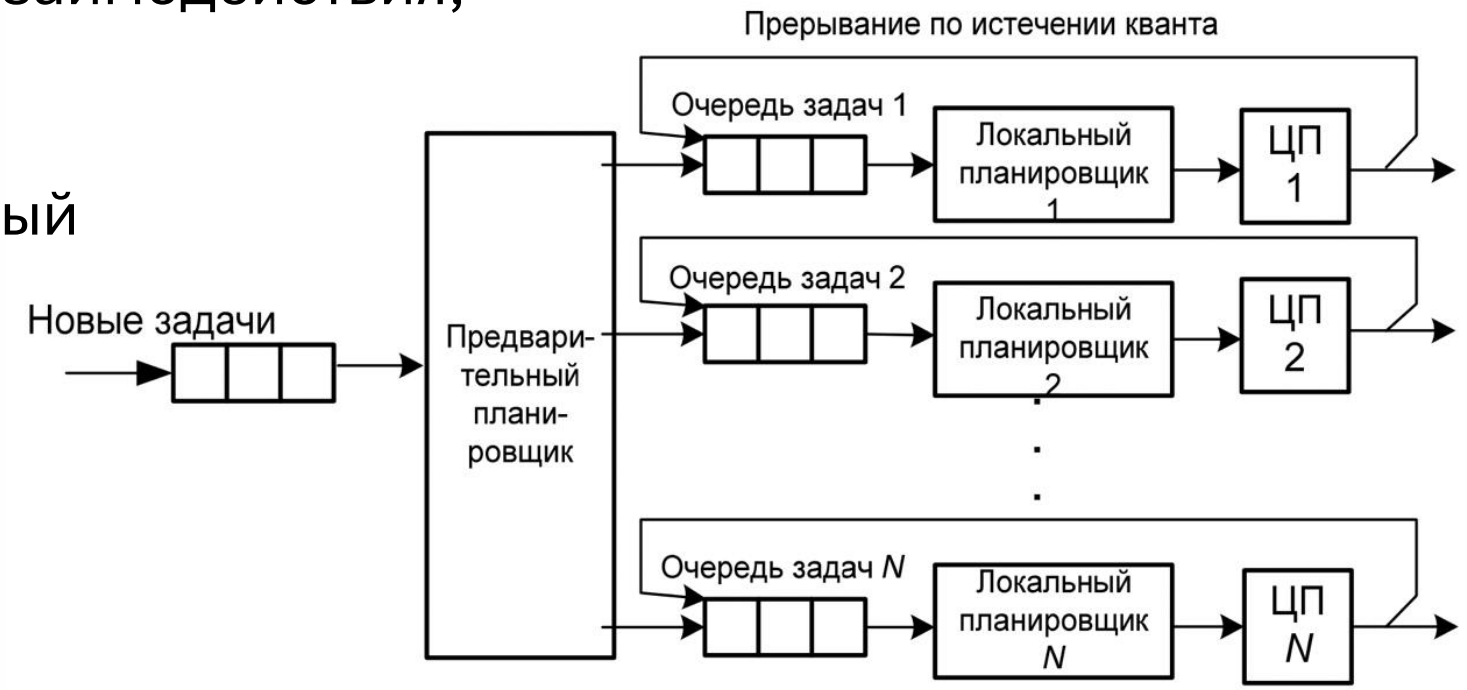
❖ Ещё один интересный подход. Реализовали алгоритм, дальше пробуем менять часть его параметров, может быть какие-то используемые функции.

Task Parallel Assembly Language for Uncompromising Parallelism

- Разработан TPAL – Task Parallel Assembly Language;
- Показана его эффективность (от -10% до +53% - прирост скорости относительно Cilk);
- Большое количество бенчмарков и графиков в статье;
- Для работы использует принцип heartbeat scheduling;
- Требуются доработки на уровне компилятора и операционной системы;
- Используются интересные техники на разных уровнях. На данный момент скорее интересно с теоретической точки зрения.

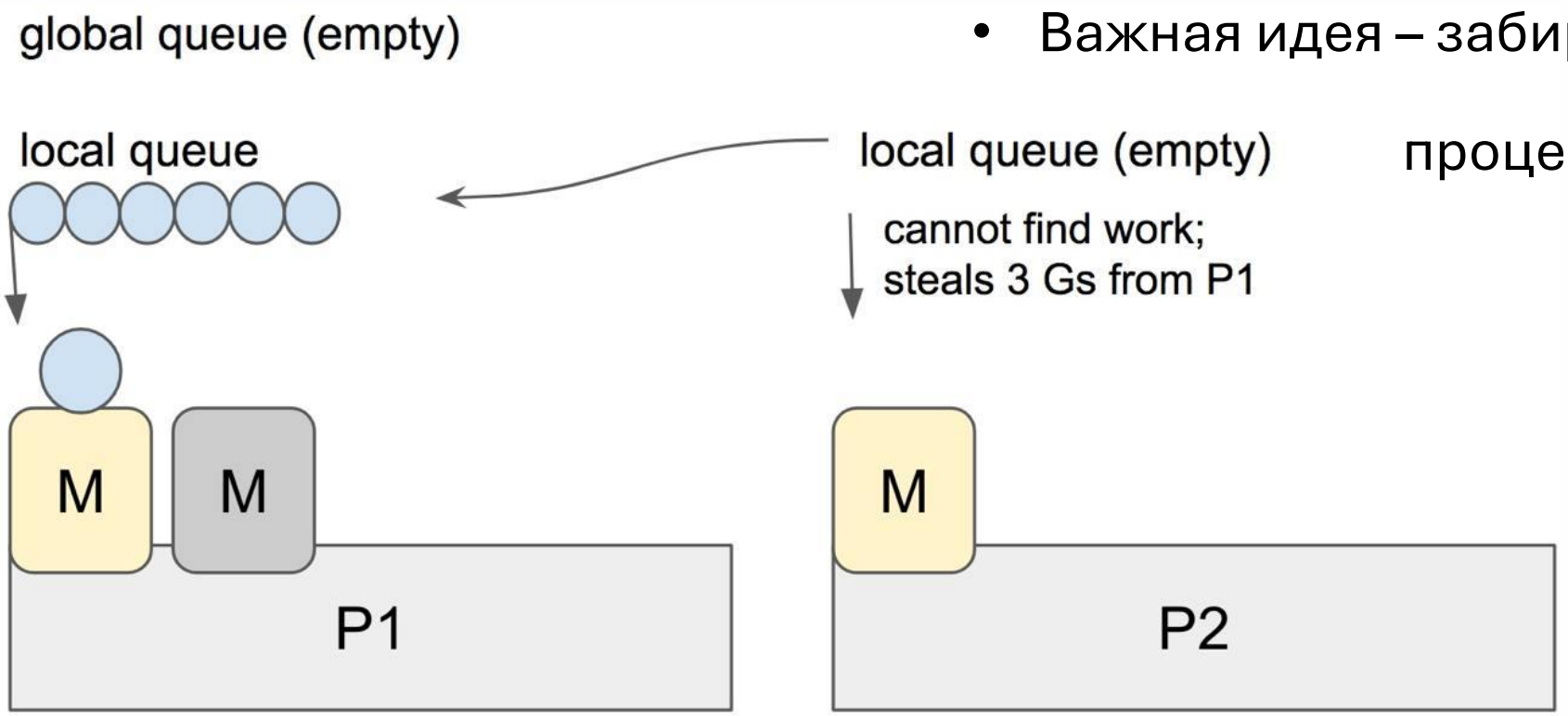
Планировщик задач с аппаратной поддержкой для многопроцессорных систем

- Уровень абстракции ниже, чем нам требуется;
- Много теории: недетерминированные автоматы, системы канонических уравнений. Также много схем взаимодействия;
- Полезные идеи:
  - ❖ глобальный предварительный планировщик;
  - ❖ локальные планировщики;
  - ❖ локальная очередь задач.



Go’s work-stealing scheduler

- Описаны два различных подхода: work-sharing и work-stealing;
- Планировщик управляется рантаймом языка
- Важная идея – забираем у другого случайного процессора половину его задач



Дальнейшие планы

1. Реализовать тест jac3d; (начало января 2025)
2. Довести до стабильно работающего состояния метод сдваивания; (середина января 2025)
3. Добиться ускорения параллельной версии теста NAS CG; (середина января 2025)
4. Реализовать первую версию планировщика (стратегию планировщика) с использованием work-stealing; (начало февраля 2025)
5. Использовать новые возможности планировщика в тестах NAS CG и jac3d. Добиться ускорения их работы. (середина февраля 2025)