

Параллельный конь в вакууме
или Проблема выбора технологий
параллелизации (для CPU и общей памяти)

Антон Потапов, C++ Russia 2024

Очень Краткое Содержание

- Рассмотрим как выбрать технологию параллелизации, чтобы не было мучительно больно

Отказ от ответственности

- Для простоты изложения некоторые упрощения допущены сознательно
- Автору неизвестны хорошие/устоявшиеся переводы некоторых английских терминов ☹️
- Это доклад **не** про CV/DL
- «понять и простить»

Отказ от ответственности

- Ниже описанная история — вольный пересказ/компиляция реальных случаев
- Все совпадения — случайны :)

(Не) Выдуманная История

Ликбез :

- Нейронные сети – чистая магия 😊
- Инференс – получение результата
- Препроцессинг – приведение изображения к размеру и формату понятному сетию

(Не) Выдуманная История

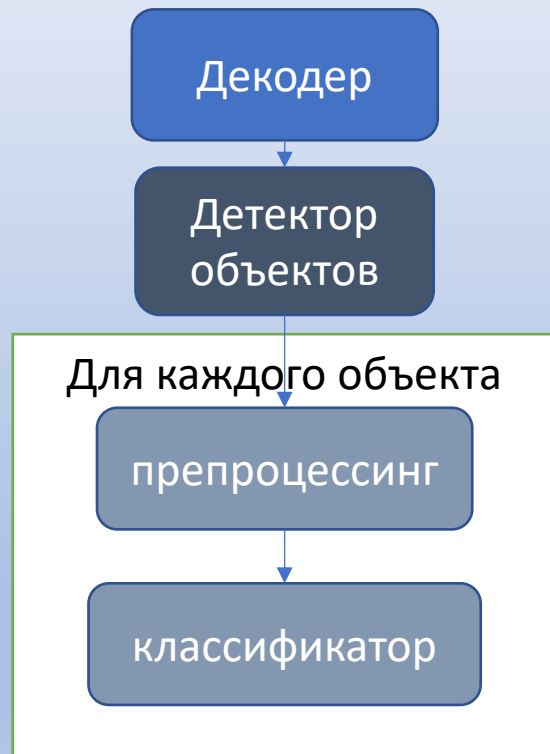
Дано:

- Вы - программист Вася П.
- работаете в одной большой IT компании
- которая, помимо прочего, разрабатывает набор библиотек для ускорения всего и вся CV/DL 😊
- **SuperCatFinder** - (идеальное) приложение – пример, оно же бенчмарк,
 - косвенно влияющий на вашу (и не только) ЗП 😊

(Не) Выдуманная История

SuperCatFinder — или как искать котиков на видео:

- Получаем от **декодера** очередной кадр
- При помощи **детектора** ищем **объекты** в кадре
- Для каждого найденного объекта:
 - **препроцессинг** (меняем размер и формат изображения на необходимый)
 - Запускаем **классификатор** — (Котик ли это?)
 - Найденных котиков «подсвечиваем»



(He) Выдуманная История

SuperCatFinder использует

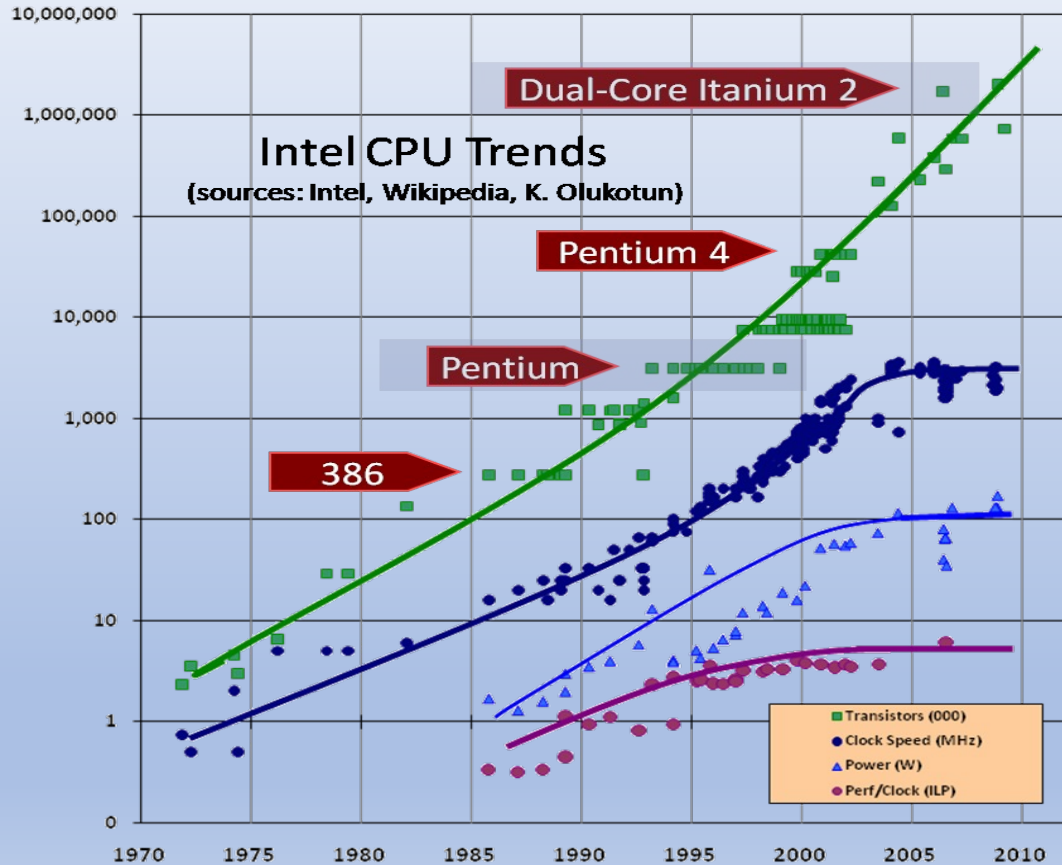
- FFmpeg — декодер видео
- 1_DNN - детектор и классификатор объектов
- **PreProc** - препроцессинг

(Не) Выдуманная История

PreProc — библиотека для препроцессинга изображений для инференса

- На замену «велосипедам»
- Инновационный алгоритм
- Крутая реализация :)
- Вася П. – (главный?) разработчик
- Версия 0.1

Free Lunch is Over



Известная картинка Герба Саттера 😊

(Не) Выдуманная История - PreProc

TBB, OpenMP , pthreads ? - Что взять ?

(Не) Выдуманная История - PreProc

TBB, OpenMP , pthreads ? - Что взять ?

- Надо мерить :
 - Пишем несколько (один :) ?) микробенчмарков
 - Запускаем
 - Замеряем
 - Сравниваем
 - Повторяем

(Не) Выдуманная История - PreProc

TBB, OpenMP , pthreads ? - Что взять ?

- OpenMP победил «за явным преимуществом»
 - Одна `#pragma`
 - Быстрее конкурентов в разы!
- Ура !

(Не) Выдуманная История - PreProc

TBB, OpenMP , pthreads ? - Что взять ?

В конце концов - мы же не threading гуру :)

OpenMP терминология

- *параллельный регион* – участок кода исполняемый несколькими потоками одновременно
- *команда* – набор потоков, участвующих в исполнении параллельного региона
- *главный (master) поток* – поток, встречающий *parallel* – конструкции OpenMP (например: `#pragma omp parallel for`)
 - обычно он будет один – тот где функция `main()` исполняется

Распределение Работы (work sharing)

#pragma omp parallel for :

- *главный поток :*
 - *создает команду*
 - *делит пространство итераций между членами команды*
 - *дожидается исполнения всех частей*
- **Гарантированное число рабочих потоков**
 - Mandatory parallelism
- **Минимальные накладные расходы**
 - **Сложно сделать быстрее**

Закон Амдала

- *«В случае, когда задача разделяется на несколько частей, суммарное время её выполнения на параллельной системе не может быть меньше времени выполнения самого длинного фрагмента»*
- **Серийные (не параллельные) части кода - «вне закона»**

(He) Выдуманная История

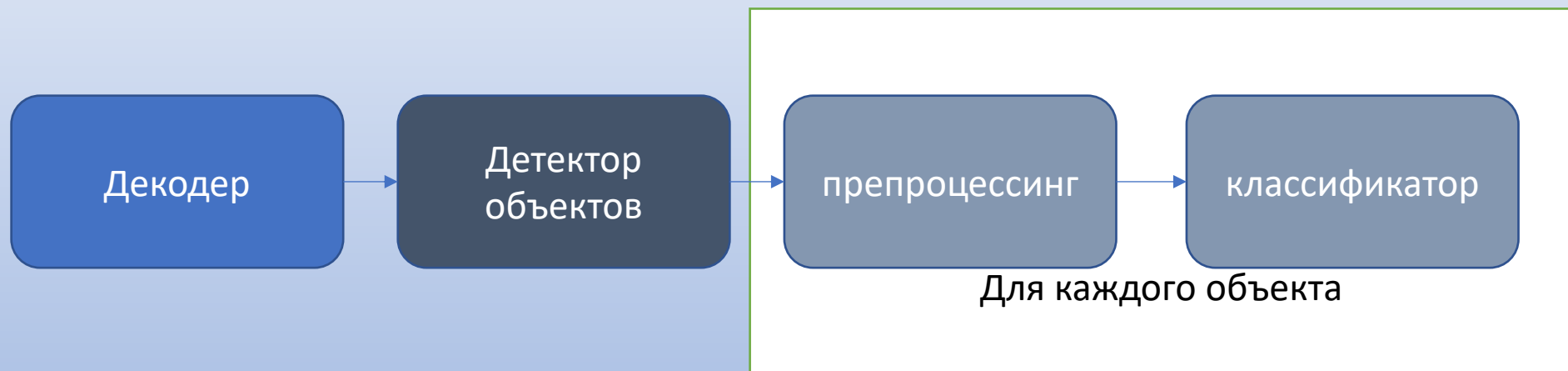
SuperCatFinder:

- Добавим конвейер (pipeline)
- Сделаем цикл обработки найденных объектов параллельным
- Profit !

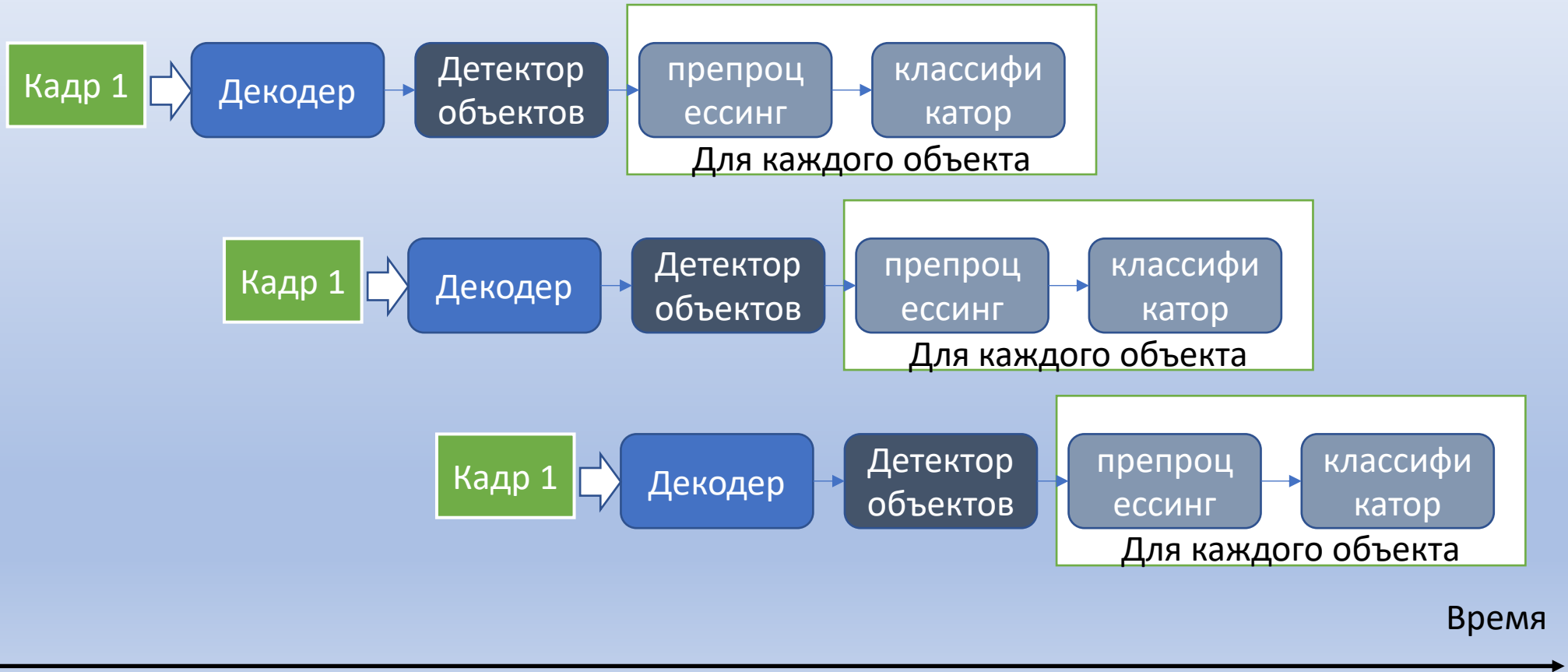
Что такое конвейер (pipeline)

- Способ «наложить» во времени друг на друга разные активности:
 - ввод/вывод и вычислительные задачи
 - разные серийные задачи между собой
 - Параллелизм!

Что такой конвейер (pipeline)



Что такой конвейер (pipeline)



(Не) Выдуманная История

SuperCatFinder

- **Добавили** конвейер (`tbb::pipeline`)
- **Сделали** цикл обработки найденных объектов **параллельным**
- Profit !
- «Может премию дадут?» :)

(Не) Выдуманная История

SuperCatFinder:

- **Добавили** конвейер (pipeline)
- **Сделали** цикл обработки найденных объектов **параллельным**
- **«Ой всё»**
 - . Стало только хуже — в разы!

(Не) Выдуманная История

- Здравствуй, **Oversubscription** !
- Причем, дважды :)
 - Вложенный параллелизм
 - Множественные главные (main) потоки

Переподписка/Oversubscription

Готовых к исполнению потоков
значительно больше чем ядер CPU

Oversubscription, Проблемы

- Переключение контекстов
 - Больше потоков — больше переключений
- «Вымывание» кэша процессора
- Активное ожидание
 - Ждущие потоки
 - не выполняют полезной нагрузки
 - активно потребляют CPU
 - не дают другим потокам выполнять полезную работу

Oversubscription, Проблемы

10 процессов * 64 потока дает
жуткий oversubscription:

```
sh$ python demo_mp.py
Seq: 2.27189302444458
295.8639030456543
295.4570367336273
297.9307312965393
```

```
# demo_mp.py
import time, numpy as np
from multiprocessing import Pool

data = np.random.random((2000, 2000))

t0 = time.time()
[np.linalg.qr(data) for i in range(10)]
print("Seq:", time.time() - t0)

p = Pool()
for j in range(3):
    t0 = time.time()
    p.map(np.linalg.qr, [data for i in range(10)])
    print(time.time() - t0)
```

Kunpeng-powered machine (Huawei TaiShan 200 Model 2280), only single socket selected with 64 cores, running pthread-based OpenBLAS

Oversubscription и OpenMP

- Параллельный регион OpenMP :
 - Неявный барьер в конце (обычно) с активным ожиданием

Oversubscription и OpenMP

- Вложенный параллелизм
 - Параллельный регион внутри другого параллельного региона
 - При входе во вложенный регион будет создана еще одна «команда» потоков
 - Размер команды зависит от настроек

Oversubscription и OpenMP

- Множественные главные (master) потоки
 - главный поток :
 - Это любой поток, в котором встречаются параллельные регионы
 - Создает команду потоков

Oversubscription и OpenMP

- Множественные главные потоки
 - `tbb::pipeline + openMP`
 - Рабочие потоки `tbb` – для OpenMP *главные потоки*
 - Oversubscription $X * X$

Oversubscription и OpenMP

- Как бороться

- KMP_BLOCK_TIME

- OMP_THREAD_LIMIT

- OMP_NESTED / OMP_MAX_ACTIVE_LEVELS

- OMP_WAIT_POLICY PASSIVE

Oversubscription и OpenMP

- Как бороться
 - KMP_BLOCK_TIME
 - OMP_THREAD_LIMIT
 - OMP_NESTED / OMP_MAX_ACTIVE_LEVELS
 - OMP_WAIT_POLICY PASSIVE
- Все они ухудшают производительность «нормального» случая
- «Лучше, чем ничего» :)

Oversubscription

- Иногда может быть полезен для
 - Блокирующий ввод/вывод
 - Параллельный запуск «плохо параллельных» приложений
 -
- Лучше, чем ничего

Oversubscription и OpenMP

- Радикальный метод борьбы
 - Не использовать ~~OpenMP~~ Work-Sharing

Перехват работы (work stealing)

- **Один** пул рабочих потоков на процесс
 - Число потоков == (число ядер CPU — 1)
 - Размер пула не меняется
- Работа **не привязана** к потокам
 - очередь (deque) задач на поток
 - новые задачи поток кладет в свою очередь
 - выполнив текущую задачу – берет новую из своей очереди
 - Перехватывает (steal) задачи у других потоков, если своя очередь пуста
- Нет гарантий на **точное** число рабочих потоков в каждом «параллельном регионе»
 - Адаптивность «из коробки»

Work Stealing & Oversubscription

- Сам подход не создает Oversubscription
- Подход более устойчив
 - При вытеснении потока — его работа может быть перехвачена другими (кроме той, которая уже начата)

Work Stealing

- Заметные накладные расходы
 - На поиск работы требуется время

Work Stealing

- Cilk
- TBB
- OpenMP tasks

(He) Выдуманная История

SuperCatFinder

– Work-Stealing – наше всё!

(He) Выдуманная История

SuperCatFinder

– **Work-Stealing** – наше всё!

- `tbb::pipeline`
- `tbb::parallel_for` - цикл обработки найденных объектов
- `1_DNN` - взяли `tbb` версию (ура!)
- `PreProc` – перевели на `tbb`

(He) Выдуманная История

SuperCatFinder

– **Work-Stealing** – наше всё!

- `tbb::pipeline`
- `tbb::parallel_for` - цикл обработки найденных объектов
- `1_DNN` - взяли `tbb` версию, (ура!)
- `PreProc` – перевели на `tbb`
- «Ну теперь точно премию дадут !» :)

(Не) Выдуманная История

SuperCatFinder

- **Work-Stealing** – наше всё!
- «премию не дадут» ☹️
 - FFMpeg подкачал ☹️
 - Но это уже другая история 😊

(Не) Выдуманная История - PreProc

- TBB, OpenMP , pthreads? – так все-таки, что взять?

(Не) Выдуманная История - PreProc

- TBB, OpenMP , pthreads? – так все-таки что взять?
 - OpenMP быстрее на микробенчмарках
 - tbb – лучше «ищет котиков» (в **SuperCatFinder**)
 - ??????

(Не) Выдуманная История - PreProc

- TBB, OpenMP , pthreads?
 - Ничего не брать!
 - Дать возможность выбора пользователю!
 - **«Правильный выбор» зависит от конечного приложения!**

Заключение

- Не микробенчмарками едиными...
- Сочетаемость (composability) разных частей/библиотек **влияет** на общую производительность
- Дать пользователю возможность выбора «подкапотной» реализации многопоточности – хорошая идея!
- Знать «детали реализации» иногда необходимо

Заключение

- Надеюсь, данный доклад побудит вас глубже изучить затронутые темы 😊

Спасибо за внимание!

Ссылки

- “The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software” <http://www.gotw.ca/publications/concurrency-ddj.htm>
- **Закон Амдала**
<https://ssd.sccc.ru/ru/content/%D0%B7%D0%B0%D0%BA%D0%BE%D0%BD-%D0%B0%D0%BC%D0%B4%D0%B0%D0%BB%D0%B0>
- OpenMP <https://www.openmp.org/>
- oneTBB <https://oneapi-src.github.io/oneTBB/index.html>