Теория и практика многопоточного программирования

ЛЕКЦИЯ 2. АРХИТЕКТУРА. ШИНА И КЭШ.

В прошлый раз...

- предпосылки возникновения многопоточного программного обеспечения (закон Мура)
- типология параллельных систем (многопоточность, кластеры, GRID, Load Balancing)
- о чём надо помнить при написании параллельной программы

Сегодня…

- Основы архитектур фон Неймана и PC
- Компоненты архитектуры, явно влияющие на производительность
  - CPU
  - Bus
  - Cache
  - Interconnect
- Программные способы влияния на производительность

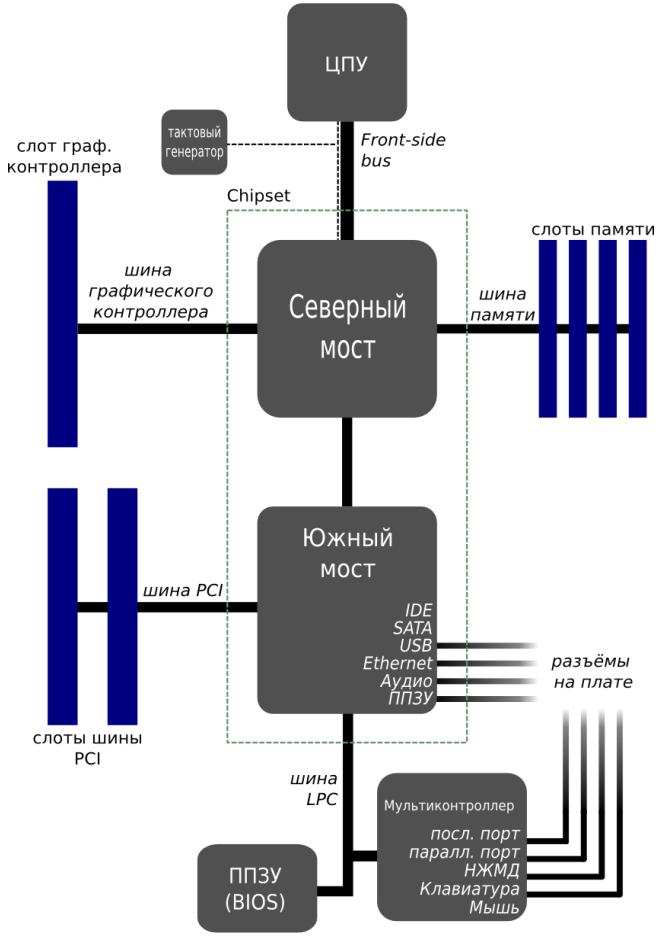
Архитектура Фон Неймана

- Принцип **двоичного** кодирования
- Принцип однородности памяти - совместного хранения **данных** и исполняемого **кода** в **памяти**
- Принцип **адресности**
- Принцин **программного управления**

Архитектура PC

Важные особенности

- Несколько ЦПУ
- Наличие памяти нескольких уровней (регистры, кэш, RAM)
- Наличие общих ресурсов – **шин**



Шины

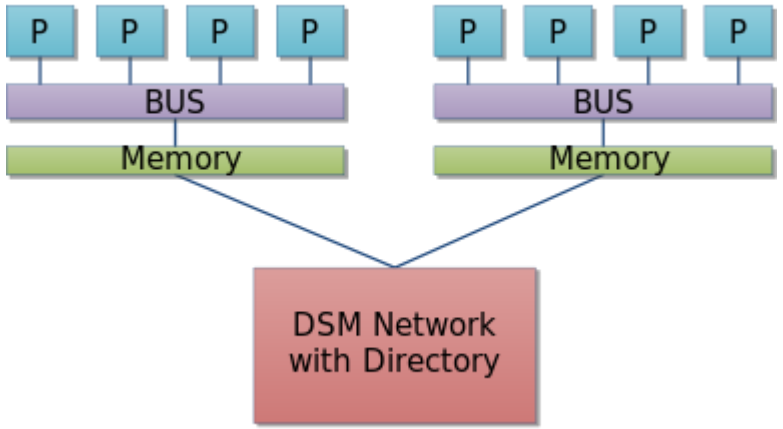
- Шина гоняет данные между устройствами, к ней подключенными
- Устройств много
- DMA – механизм «развязывания рук» при общении с медленными устройствами
- Чем меньше мы общаемся с использованием общей шины, тем мы шустрее

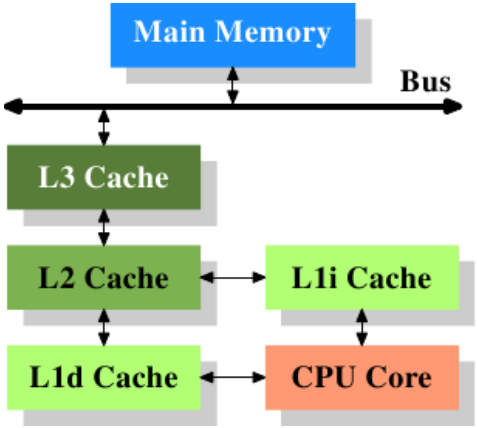
CPU и доступ к данным

Общение с памятью – большая проблема для современных систем

Выходы из ситуации:

- NUMA vs UMA/SMP
- Cache

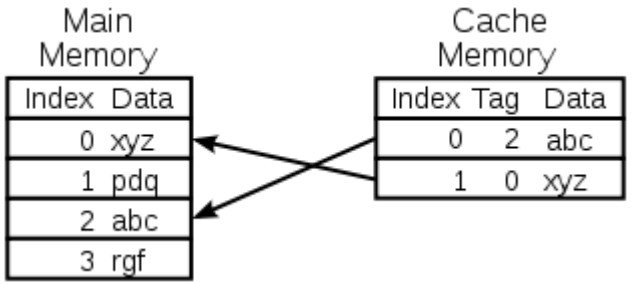




Cache

Cache хранится «линиями»

Cache **hit** vs cache **misses**




```
for (k = 0; k < times; k++) {
    for (i = 0; i < N; i++) {
        _copy[i] = _data[i];
    }
}
for (j = 0; j < N / line; j++) {
    for (k = 0; k < times; k++) {
        for (i = j * line; i < (j + 1) * line; i++) {
            _copy[i] = _data[i];
        }
    }
}
```

Многоядерность и многопоточность

Многоядерность:

- Процессоров много – шина одна
- Процессоры с локальным кэшем (NUMA) – проблема cache coherence

Многопоточность:

- Проблема переключения контекстов
- Проблема инвалидации кэша
- Технология Hyper Threading – процессор хранит контекст 2 потоков.

Итого: думай о данных

Количество инструкция – не синоним времени исполнения. Доступ к памяти определяет многое.

Что приводит в промахам кэша:

- Неупорядоченное чтение массивов
- Большое количество разыменований объектов (*a.some.property.value*)
- Большое количество уровней абстракции в протоколах (когда при конвертации данные прокачиваются через кэш).

Кэш, Interconnect– неуправляемые устройства, однако понимание принципов их работы поможет повысить скорость программы.