February 6, 2025

Announcements

- HW1

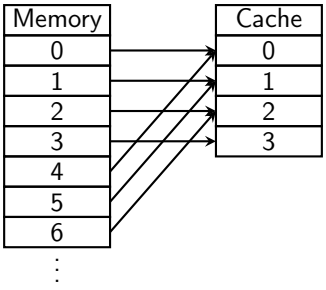
Goals

- Talk assignments
- Caches/mem. hierarchy
  - empirical
  - models

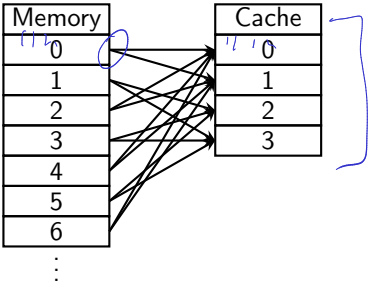
Review

Associativity

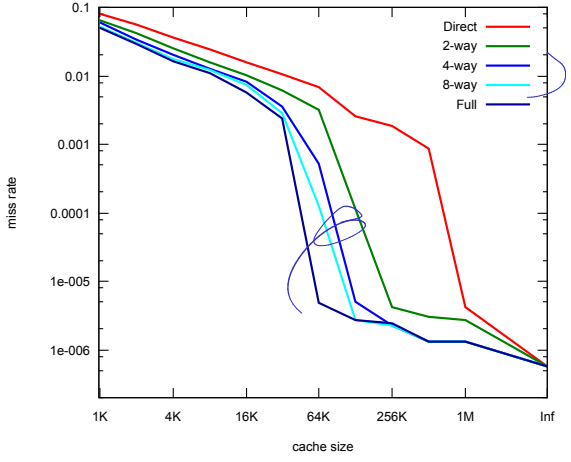
Direct Mapped:



2-way set associative:



Size/Associativity vs Hit Rate



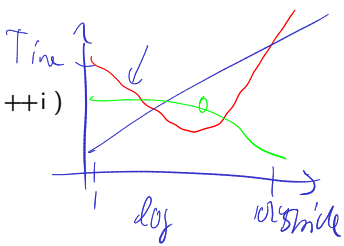
Miss rate versus cache size on the Integer portion of SPEC CPU2000 [Cantin, Hill 2003]

Demo: Learning about Caches

Demo: intro/Cache Organization on Your Machine

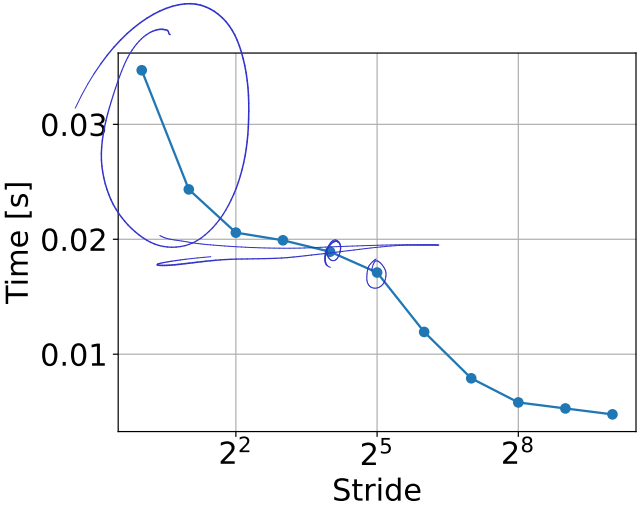
Experiments: 1. Strides: Setup

```
int go(unsigned count, unsigned stride)
{
  const unsigned array_size = 64 * 1024 * 1024;
  int *ary = (int *) malloc(sizeof(int) * array_size);

  for (unsigned it = 0; it < count; ++it)
  {
    for (unsigned i = 0; i < array_size; i += stride)
      ary[i] *= 17;
  }

  int result = 0;
  for (unsigned i = 0; i < array_size; ++i)
      result += ary[i];

  free(ary);
  return result;
}
```



What do you expect? [Ostrovsky '10]

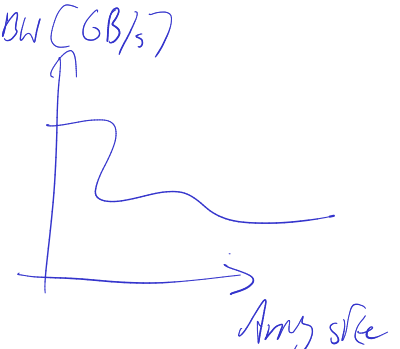
Experiments: 1. Strides: Results



```
float a[1024*1024];
for(int i=0; i<1024; ++i)
    for(int j=0; j<1024; ++j)
        a[i*1024+j] *= 2;   A
        a[j*1024+i] *= 2;   B
```

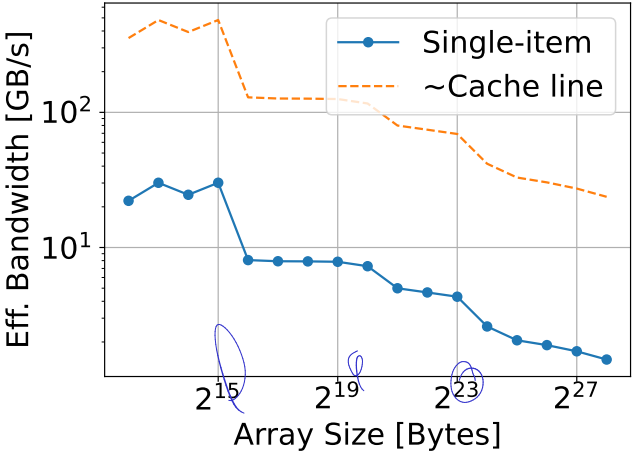
Experiments: 2. Bandwidth: Setup

```
int go(unsigned array_size, unsigned steps)
{
  int *ary = (int *) malloc(sizeof(int) * array_size);
  unsigned asm1 = array_size - 1;

  for (unsigned i = 0; i < 100*steps;)
  {
    #define ONE ary[(i++*16) & asm1] ++;
    #define FIVE ONE ONE ONE ONE ONE
    #define TEN FIVE FIVE
    #define FIFTY TEN TEN TEN TEN TEN
    #define HUNDRED FIFTY FIFTY
    HUNDRED
  }

  int result = 0;
  for (unsigned i = 0; i < array_size; ++i)
      result += ary[i];

  free(ary);
  return result;
}
```



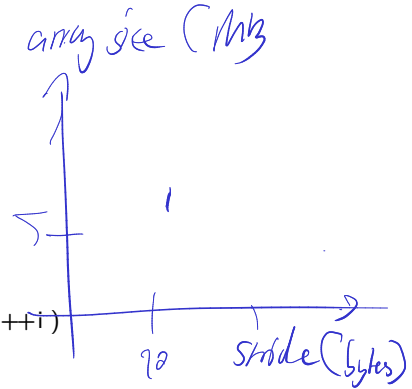
What do you expect? [Ostrovsky '10]

Experiments: 2. Bandwidth: Results

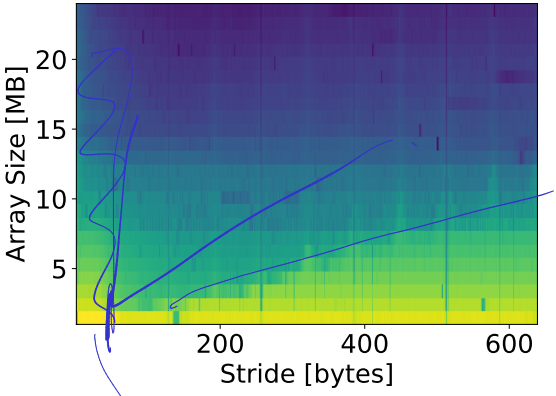


Experiments: 3. A Mystery: Setup

```
int go(unsigned array_size, unsigned stride, unsigned steps)
{
  char *ary = (char *) malloc(sizeof(int) * array_size);

  unsigned p = 0;
  for (unsigned i = 0; i < steps; ++i)
  {
    ary[p] ++;
    p += stride;
    if (p >= array_size)
      p = 0;
  }

  int result = 0;
  for (unsigned i = 0; i < array_size; ++i)
      result += ary[i];

  free(ary);
  return result;
}
```



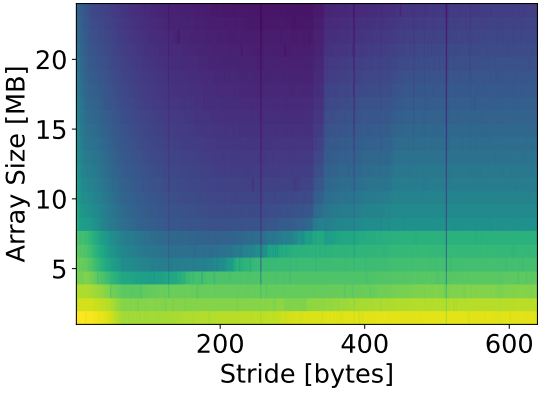
Experiments: 3. A Mystery: Results



Color represents achieved bandwidth:

- Yellow: high
- Blue: low

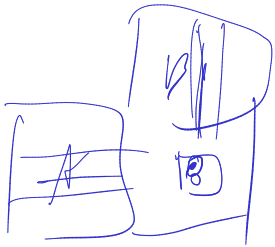
Experiments: 3. A Mystery: Results on Sandy Bridge



Color represents achieved bandwidth:

- Yellow: high
- Blue: low

Thinking about the Memory Hierarchy



- What is a working set?
- What is data locality of an algorithm?
- What does this have to with caches?

Case Study: Streaming Workloads

Q: Estimate expected throughput for saxpy on an architecture with caches. What are the right units?

$$z_i = \alpha x_i + y_i \quad (i = 1, \ldots, n)$$

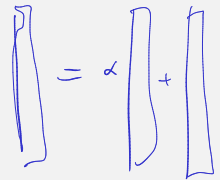
"Triad" / "StreamTriad"

Units: GBytes/s

Mem accessed: $4 \cdot n \cdot 3$

$4 \cdot n \cdot (3+1)$ ← comes from needly caheling for z

$z_0 = \alpha x_0 + y_0$



Demo: https://github.com/lcw/stream_ispc