

Cachemulator

A Snooping-based Coherent Parallel Cache Emulator

A 15-418 Spring 2014 Final Project

Yuyang Guo (yuyangg@andrew.cmu.edu)

Isaac Lim (idl@andrew.cmu.edu)

The problem

```
static int grades[NUM_STUDENTS];

// Called by each pthread i
void updateStudent(int i) {
    grades[i] = i*2;
}
```

Bad

```
struct student {
    int grade;
    char pad[64 - sizeof(int)];
};

static struct student grades[NUM_STUDENTS];

// Called by each pthread i
void updateStudent(int i) {
    grades[i].grade = i*2;
}
```

Good

Making an invisible problem visible



Cachemulator: Inputs

- The user's executable
- Parallelism can be written in any language
 - e.g. *pthread*s, OpenMP, Open MPI, etc.
- Cache configuration
 - e.g. Cache hit: 4 cycles, Memory access: 100 cycles

Cachemulator: Outputs

- Hits / Misses / Evictions
- Bus traffic
- Invalidations
- Number of processor **ticks** (cycles)
- Highly-contended lines

Cachemulator: A 3-step process

1. Using **pin**, generate a memory trace
 - Intel's dynamic binary instrumentation tool
2. Simulate execution of the trace
 - Carry out **coherence** through a simulated bus
3. Generate charts and statistics

Cachemulator is cache coherent

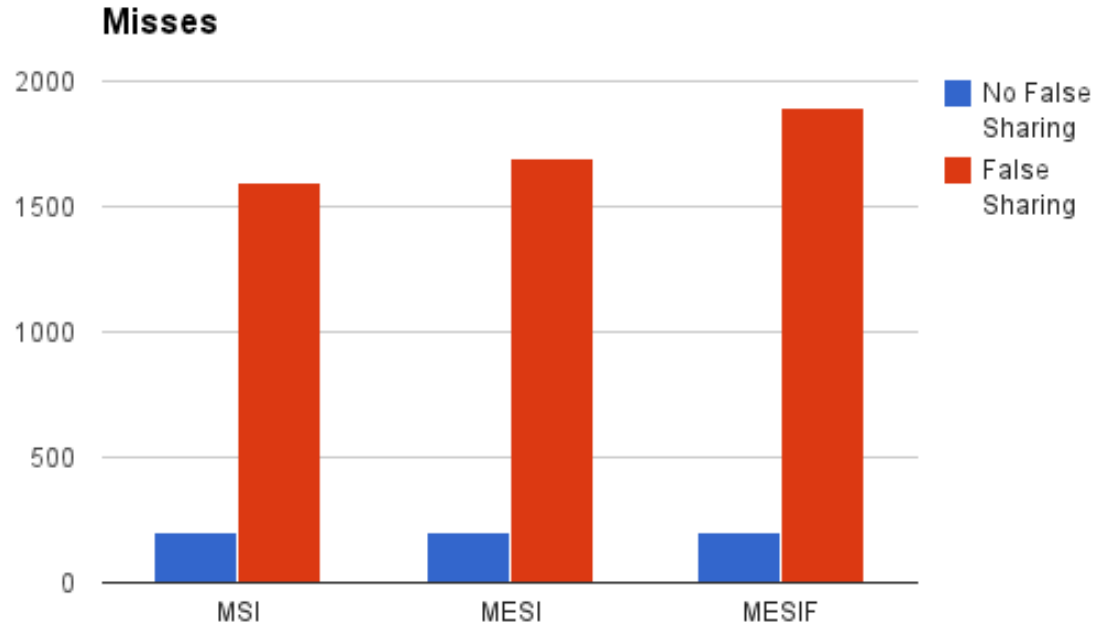
- Snooping-based
- Protocols supported: **MSI / MESI / MESIF**
- Shared atomic bus

Why is this challenging?

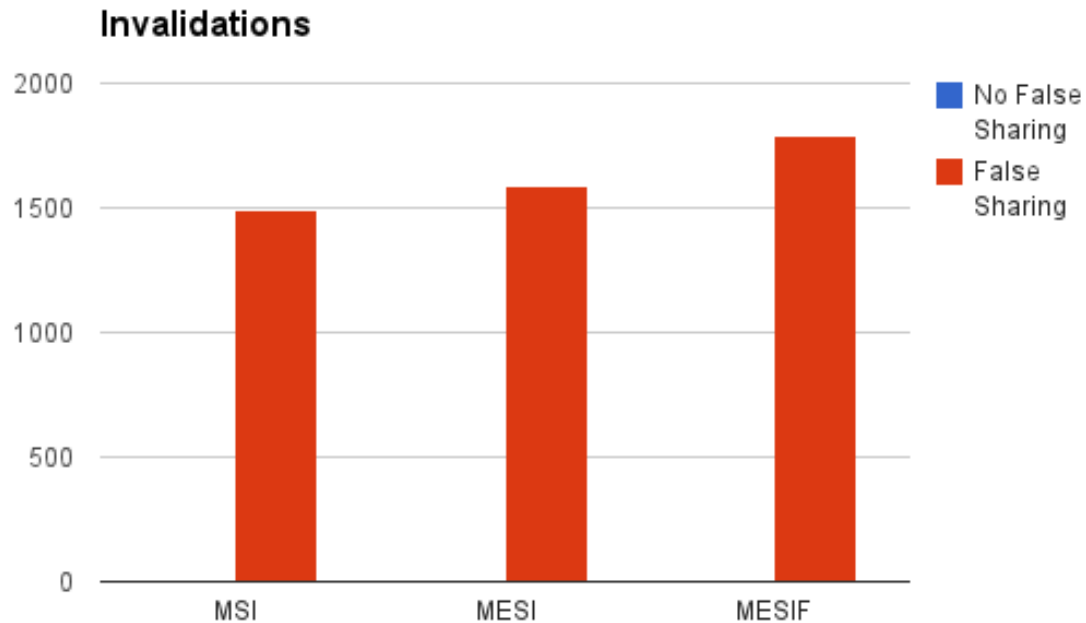
- Large memory trace size
 - ~50,000 lines for `int main() { return 0; }`
 - Full trace for BFS is on the order of ~10GBs
 - Solution: Pass in a list of the **important** functions
- Dealing with “race conditions” in the sequential simulator
- Ensuring correctness of cache coherence

An example: False sharing

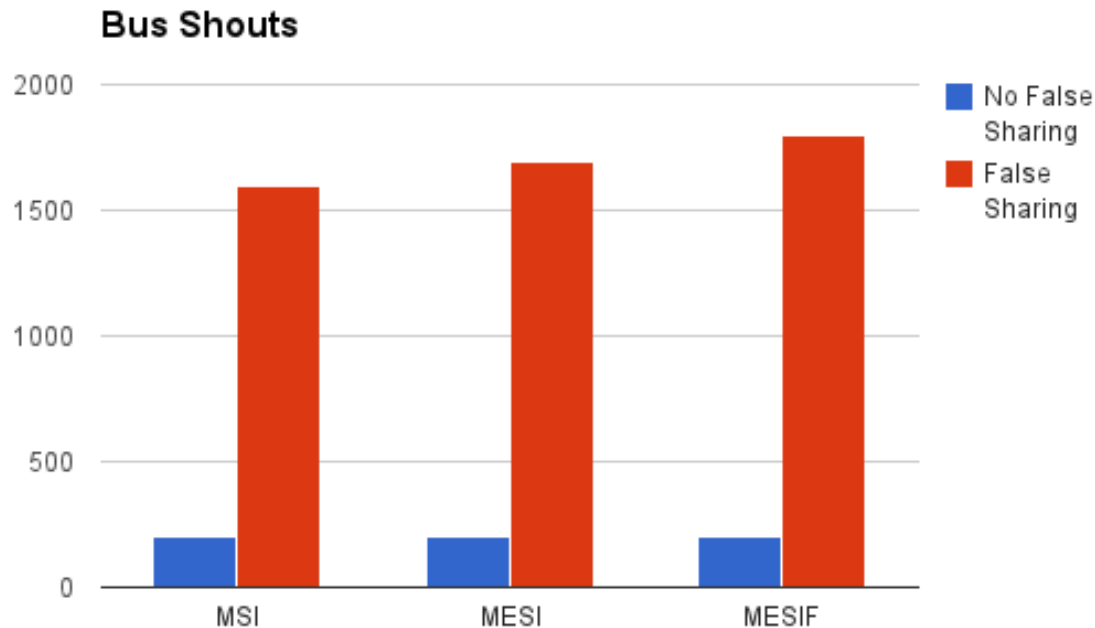
False sharing: Misses



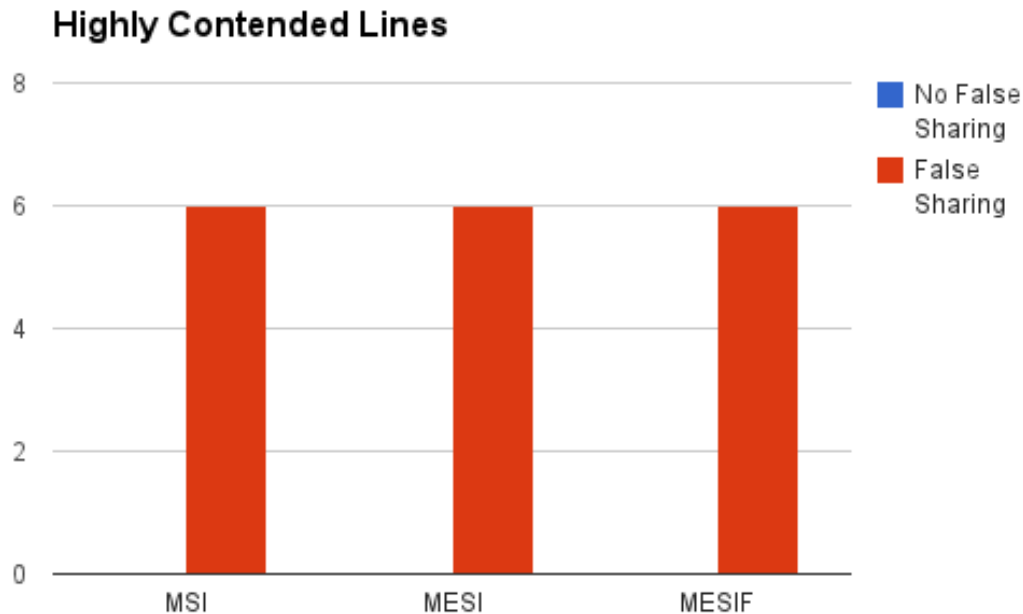
False sharing: Invalidations



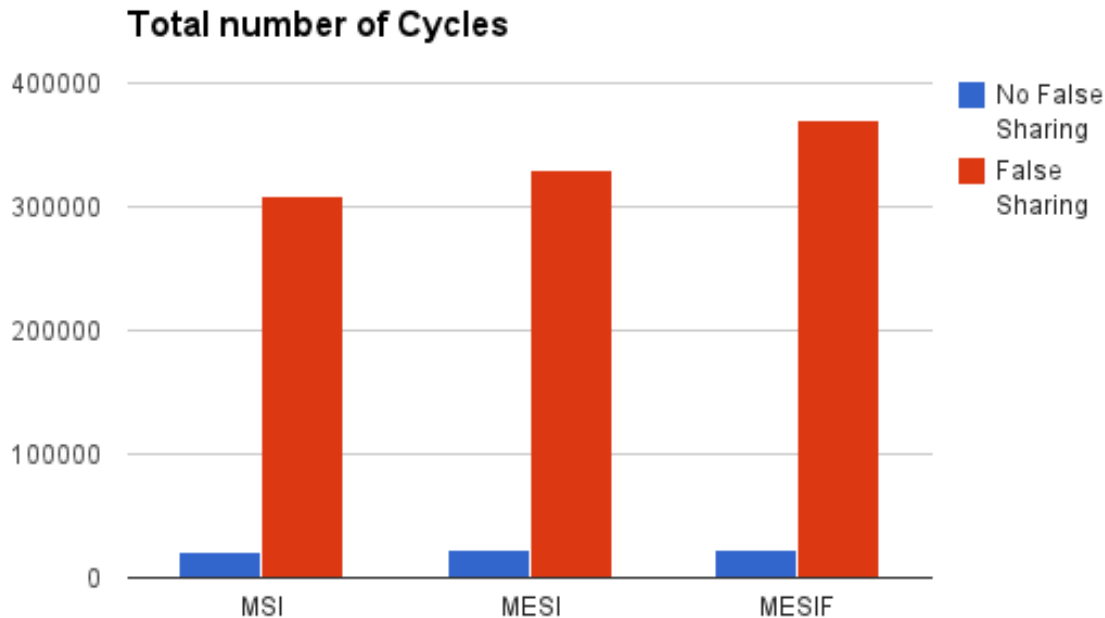
False sharing: Bus traffic



False sharing: Highly-contended lines

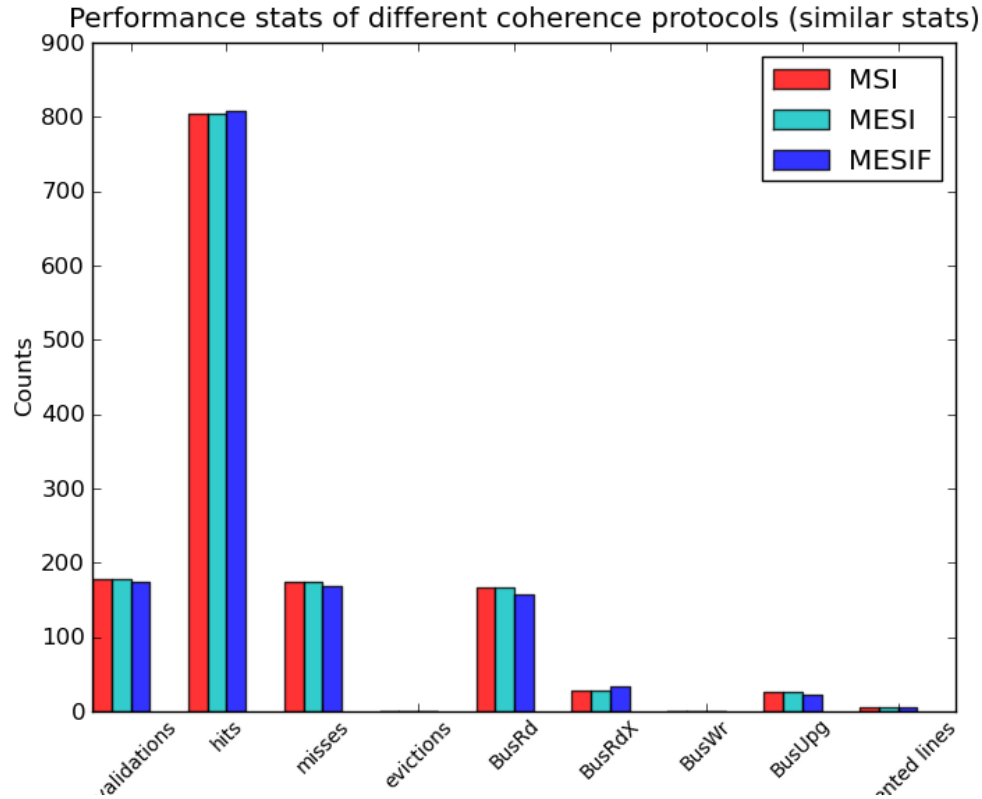


False sharing: Execution time

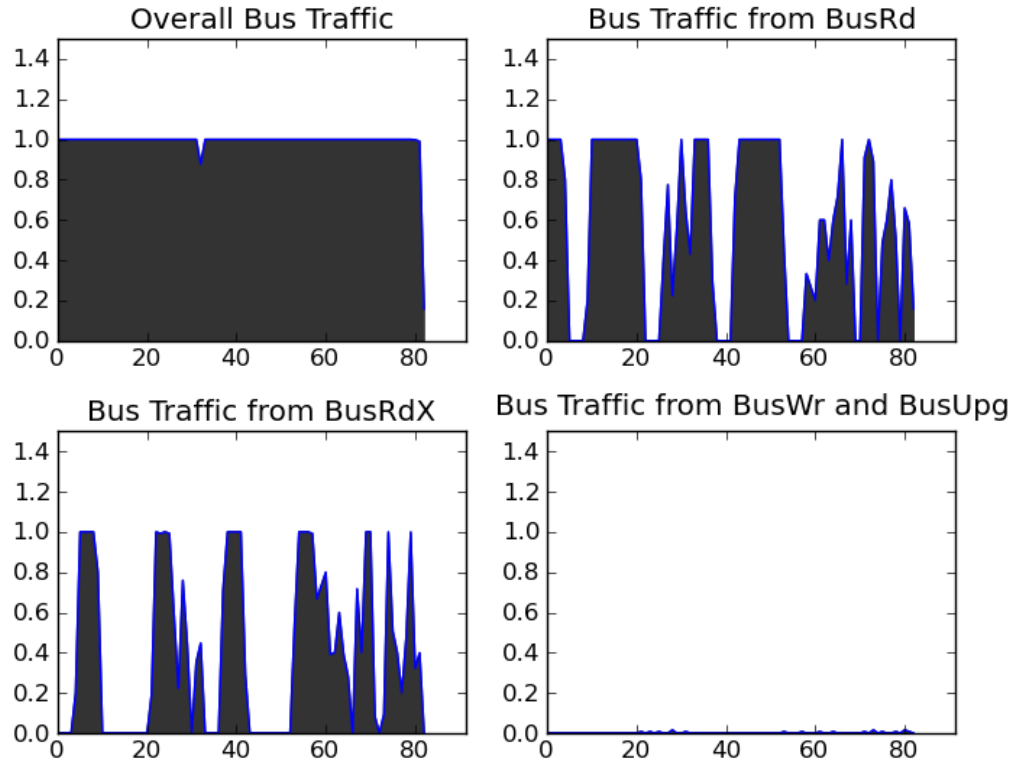


A larger example: OpenMP BFS (Asst 4)

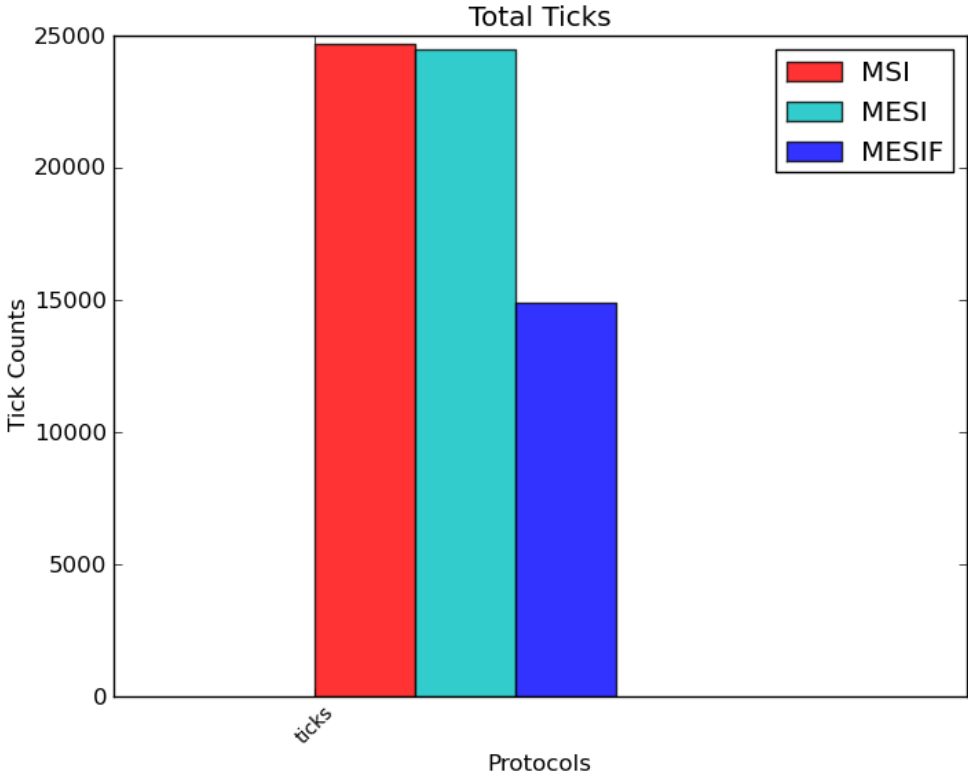
BFS: Performance



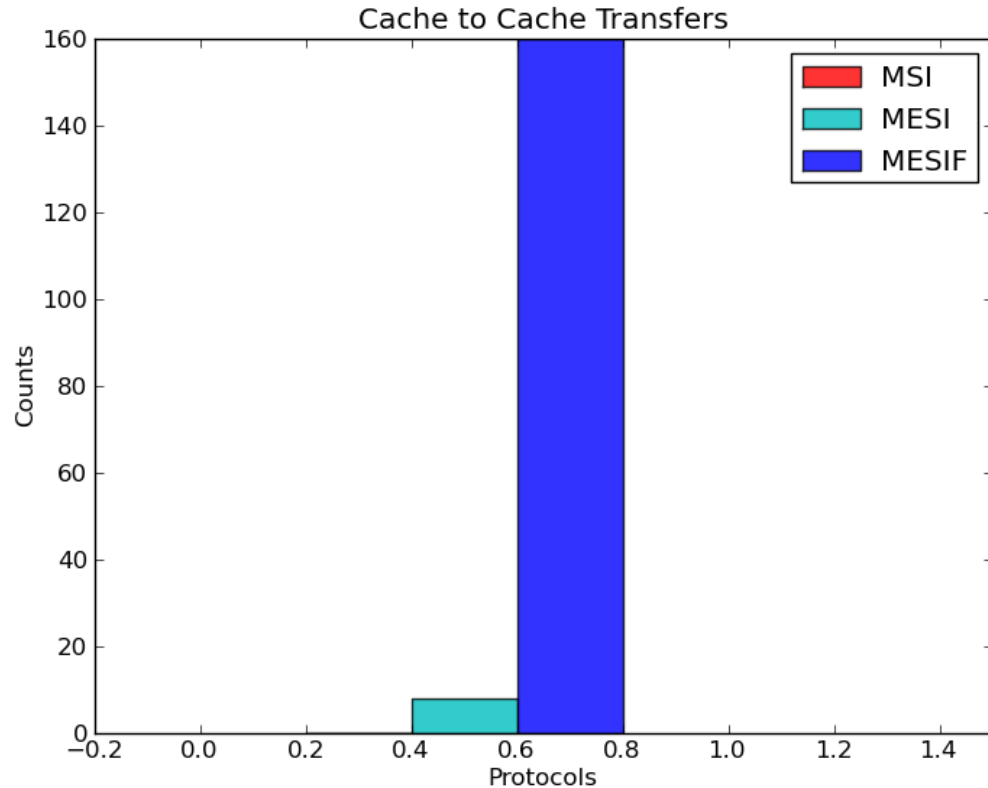
BFS: Bus traffic



BFS: Total Time



BFS: Cache-to-cache transfers



Thank you!

Cachemulator

Yuyang Guo (yuyangg@andrew.cmu.edu)

Isaac Lim (idl@andrew.cmu.edu)