Benchmarking SMP Memory System Performance

Bronis R. de Supinski and Andy Yoo

Center for Applied Scientific Computing

Frank Mueller

North Carolina State University

Sally A. McKee

***University* of Utah**

October 11, 2001

Memory Microbenchmarks

- No existing benchmarks designed for SMPs
  - Parallelization method
  - Memory system contention
- STREAM
  - Memory intensive kernels
  - Limited to unit stride access patterns
  - Suggestion for OpenMP parallelization
- HBenchOS and lmbench
  - Simple memory intensive operations
  - Limited memory access patterns
  - No parallel implementations available

New Parallel Memory Microbenchmarks

- Parallelized version of HBenchOS
  - OpenMP parallel directives
  - Memory tests only: bw_mem_rd; bw_mem_wr; bw_mem_rdwr; bw_mem_cp, bw_mem_zero; and lat_mem_read
  - Extended access patterns
  - Reduced overhead with “macro-generator”
- New method for determining architectural features
  - Based on hardware performance monitors (PAPI)
  - Using direct deductions
  - Complements inference methods

HBenchOS Memory Read Bandwidth Test

```
for (many_iterations)
  while (more memory)
    for (10 times)
      acc += p[0]+p[1]+...+p[19];
      p += 20;
```

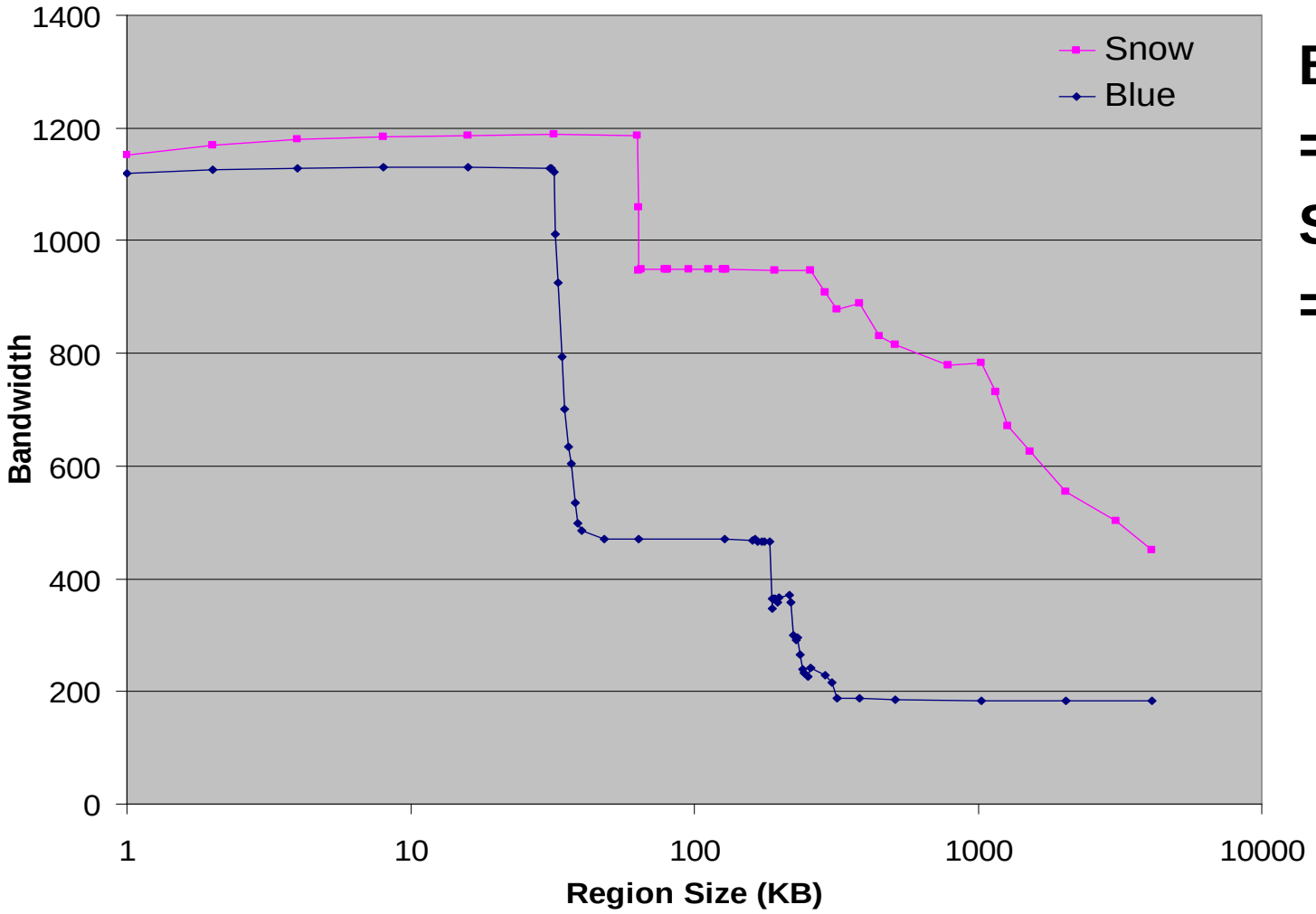
- Pluses
  - Clearly correspondence to memory size
  - Limited overhead for large memory sizes
- Drawbacks
  - Limited to unit stride
  - Limited to multiples of 200 * sizeof(int)
  - Sequential only

Parallel Memory Read Bandwidth Test

```
#pragma omp parallel
for (many_iterations)
  TEST_PARAM_BASED_MACRO
```

- Test parameters
  - Amount of memory to access (region size)
  - Stride
  - Number of threads
  - Miscellaneous things like alignment
- Scripted test procedure
  - Generate macro
  - Compile test
  - Run

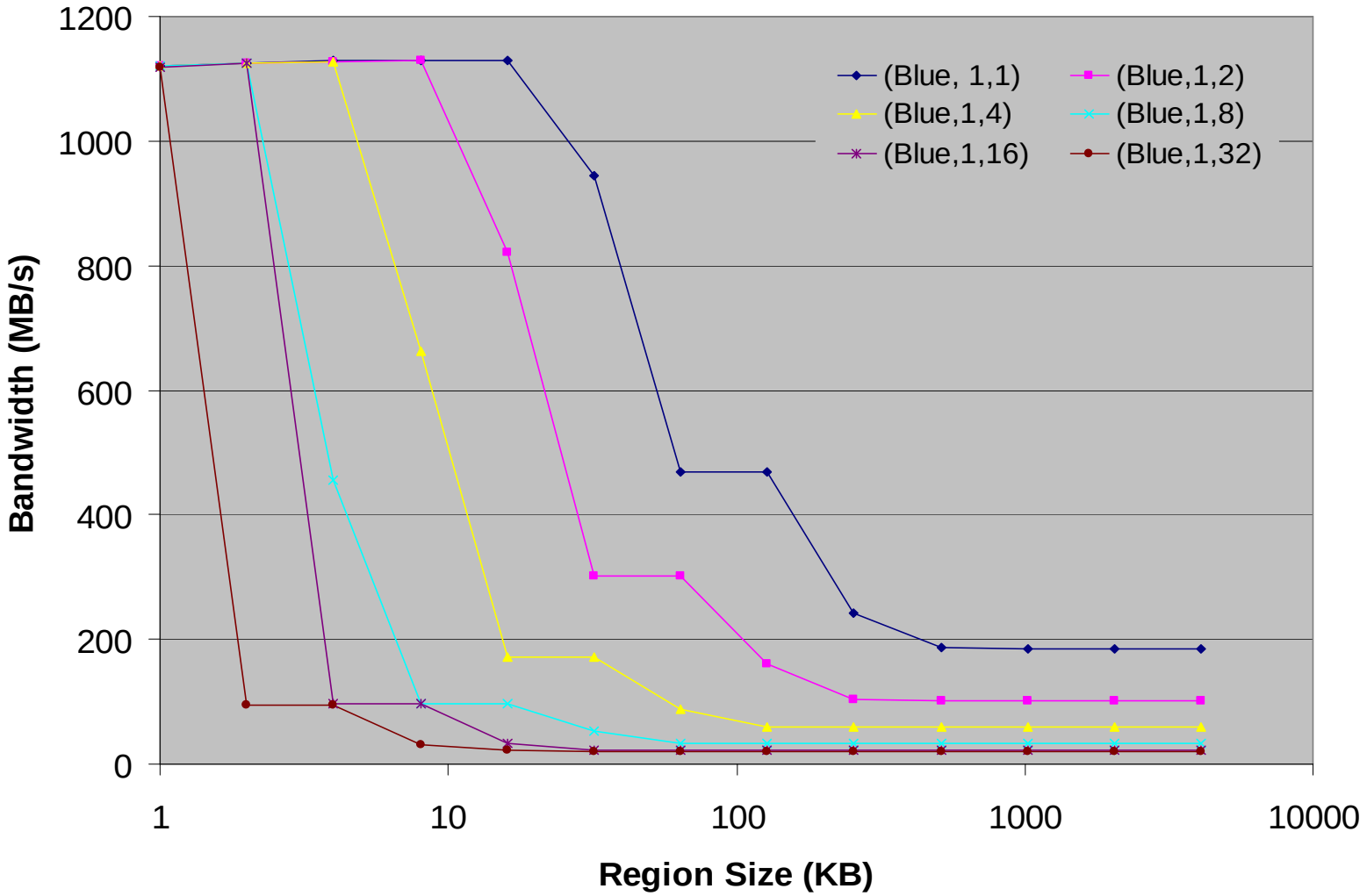
Single Thread Unit Stride Read Bandwidth



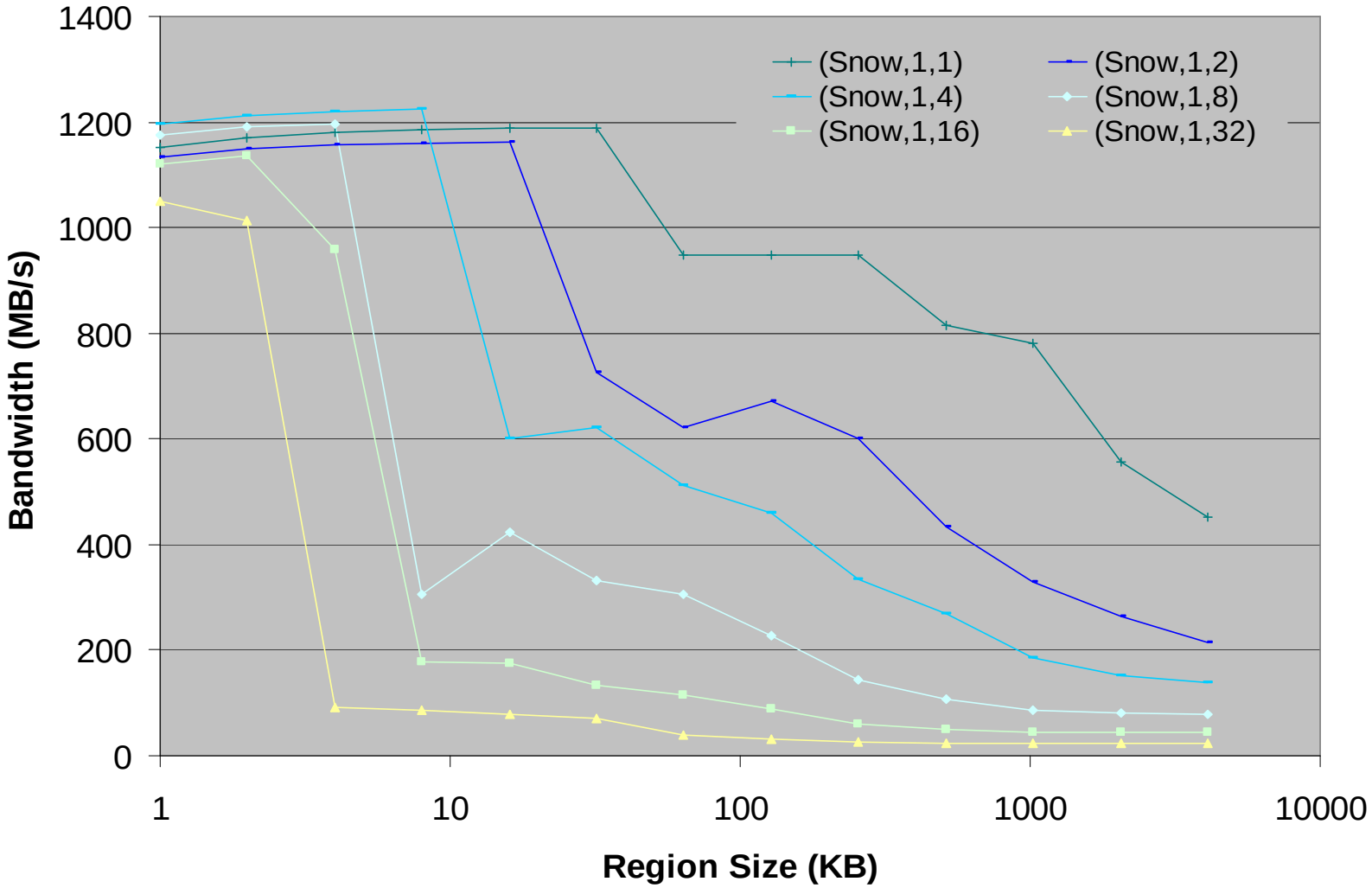
Blue L1 = 32KB

Snow L1 = 64KB

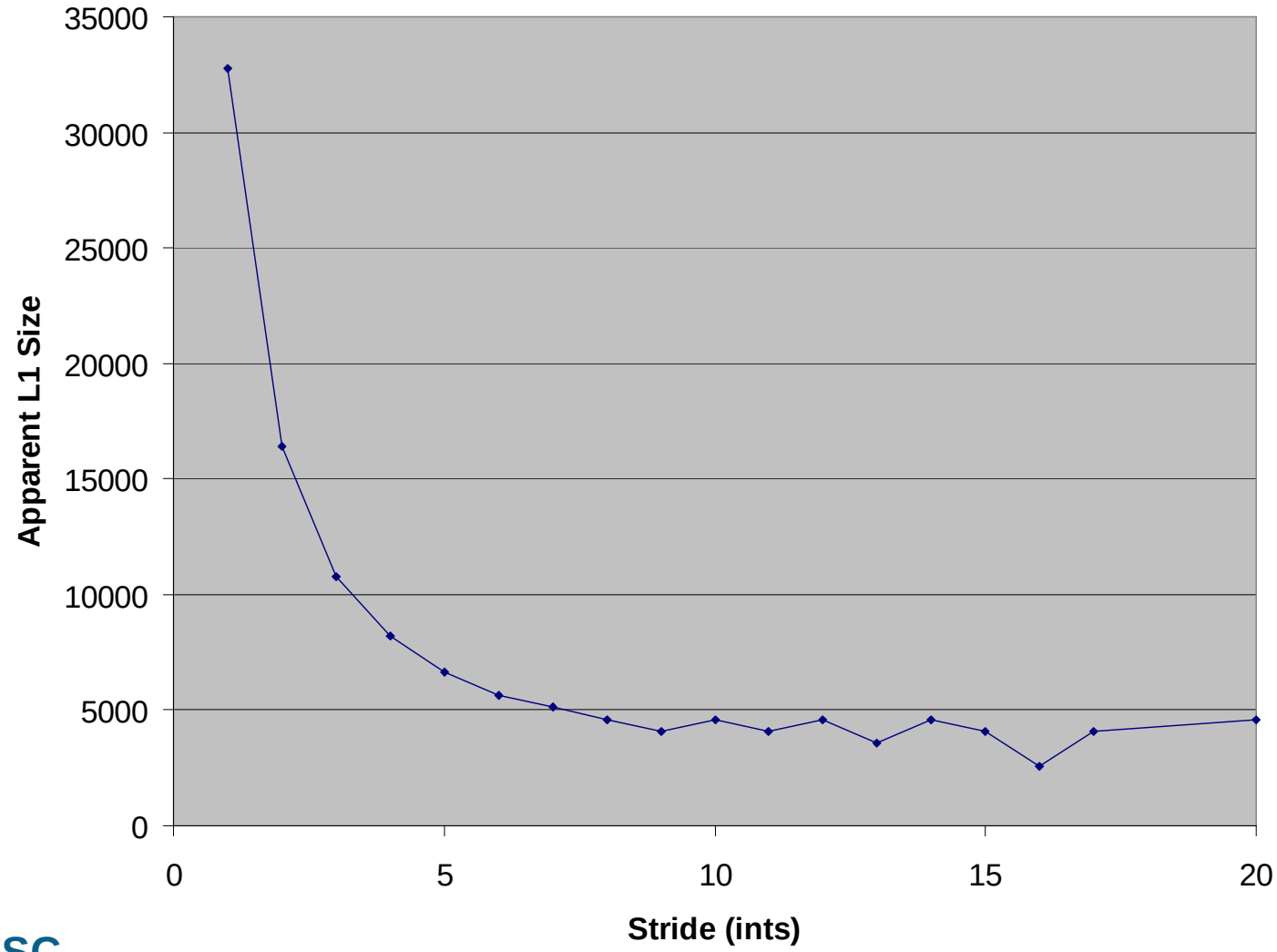
Effect of Varying Stride



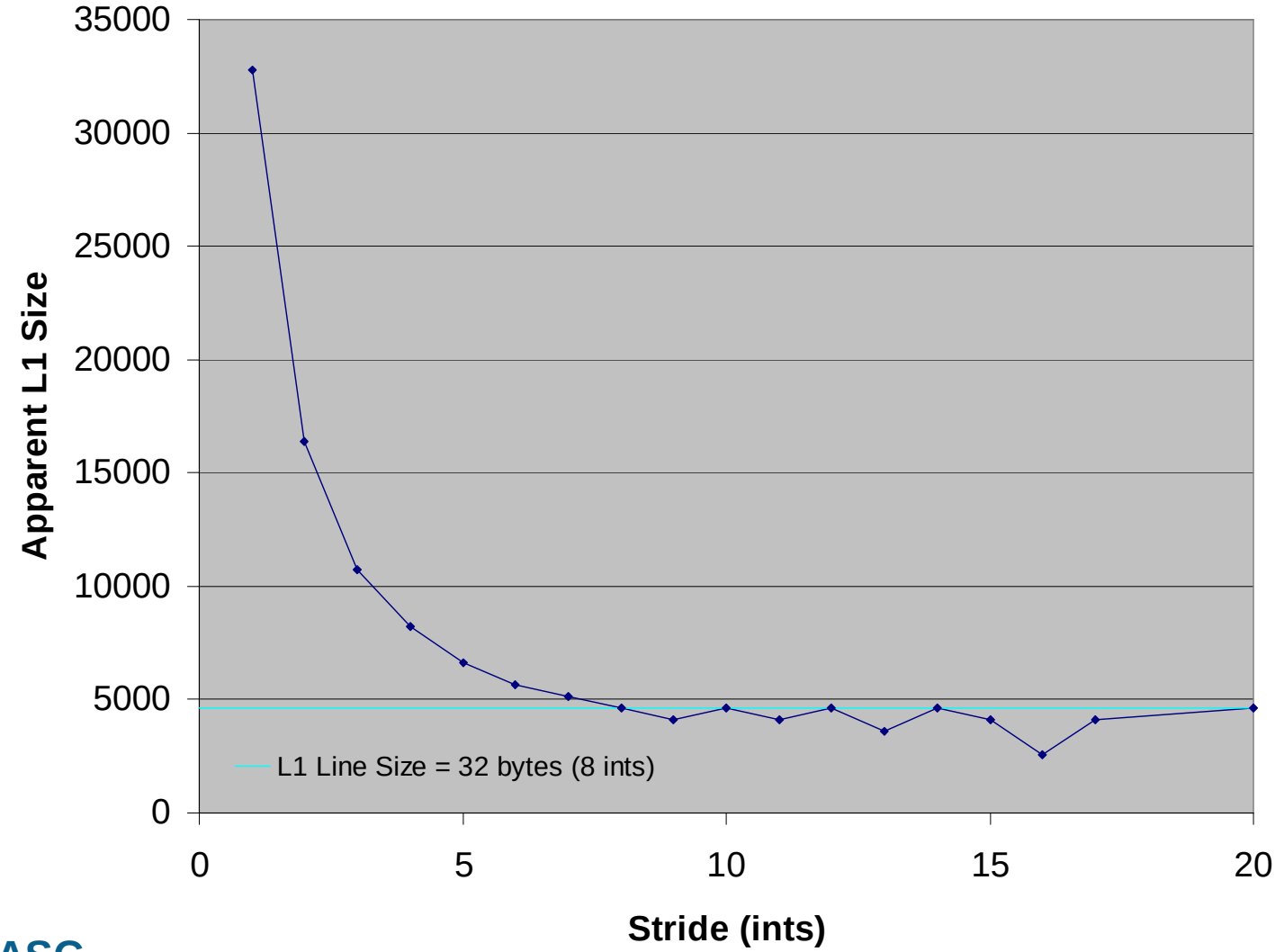
Effect of Varying Stride



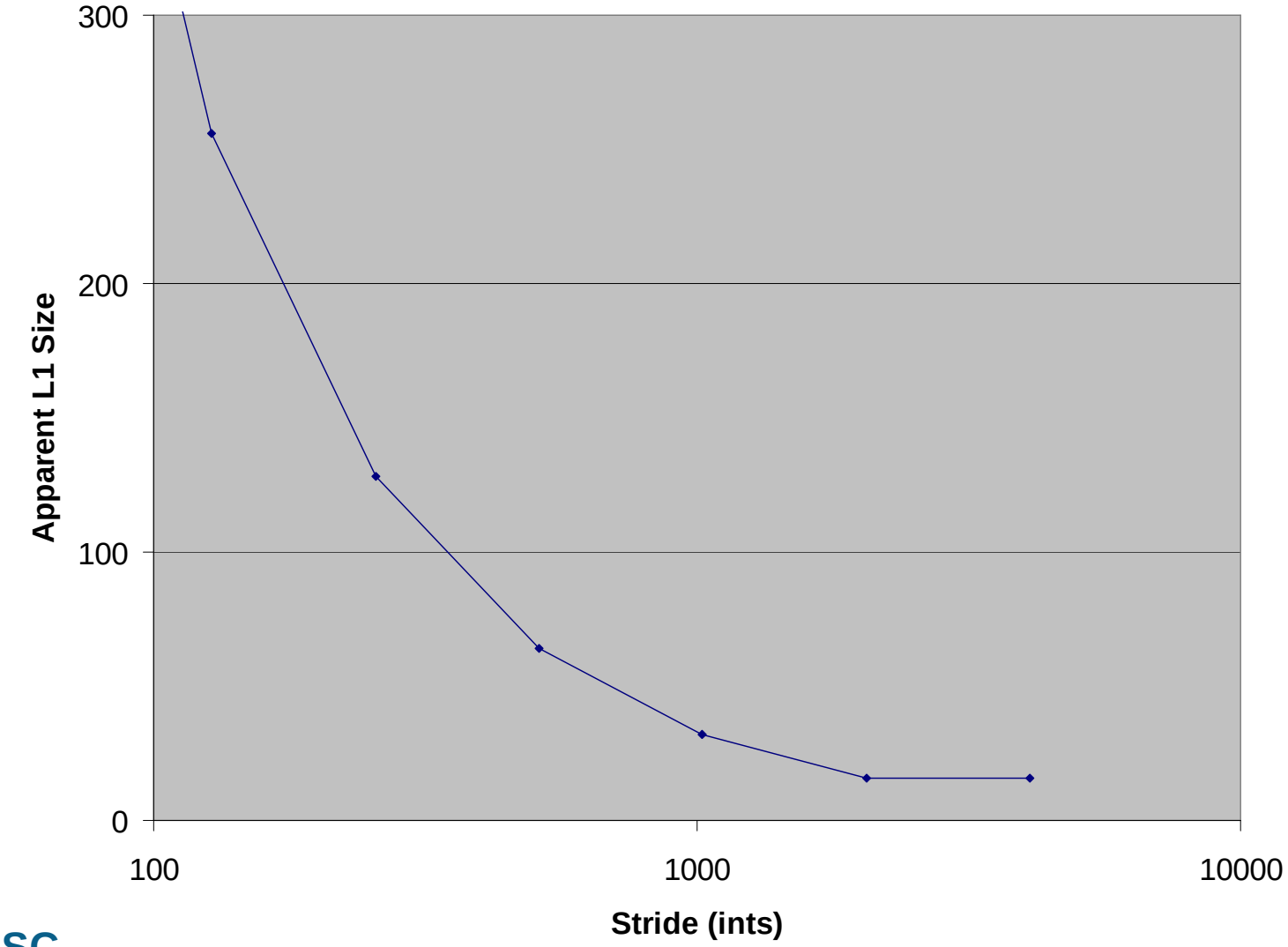
Inferring More Blue Architectural Features



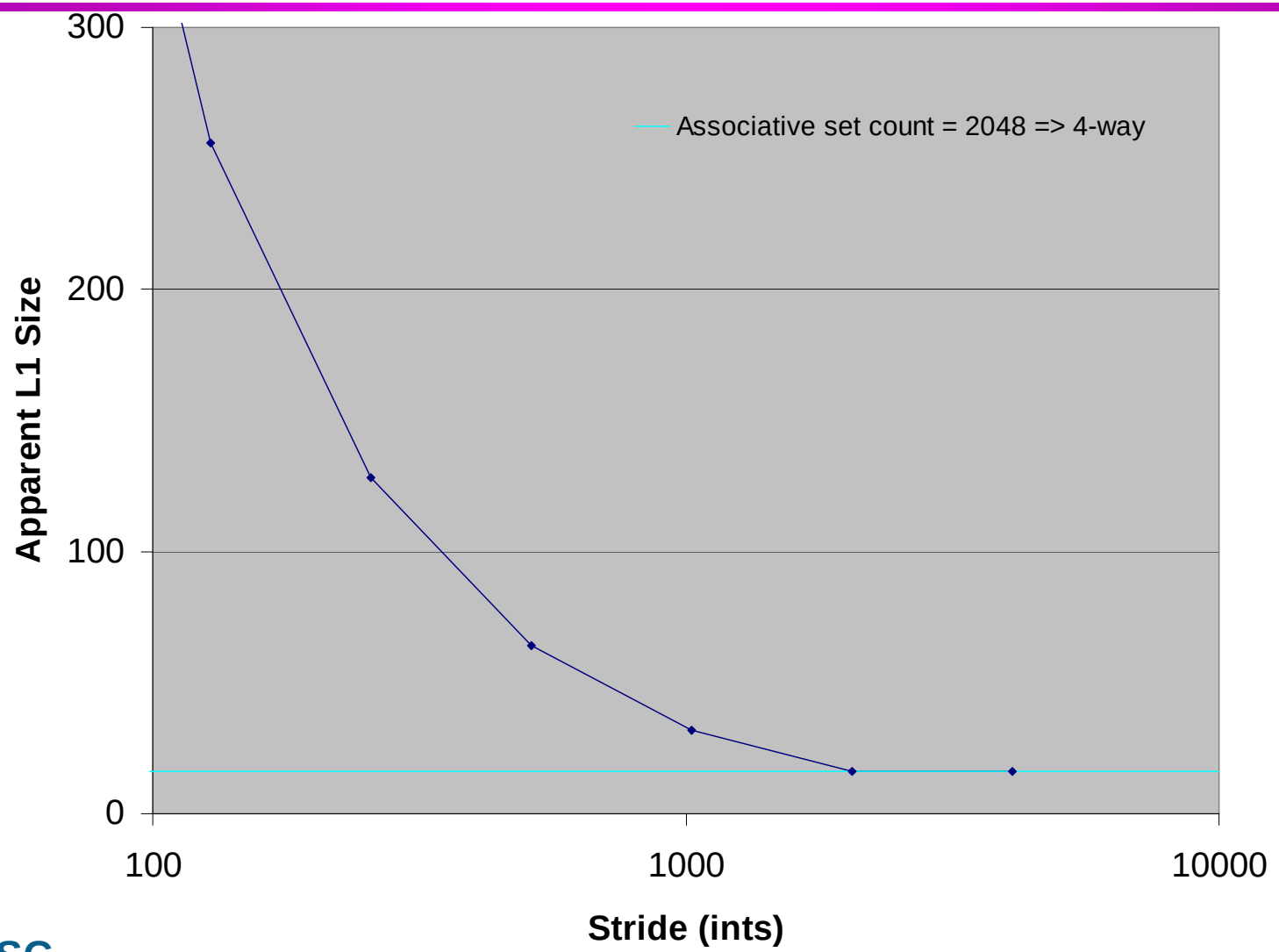
Inferring More Blue Architectural Features



Inferring More Blue Architectural Features



Inferring More Blue Architectural Features

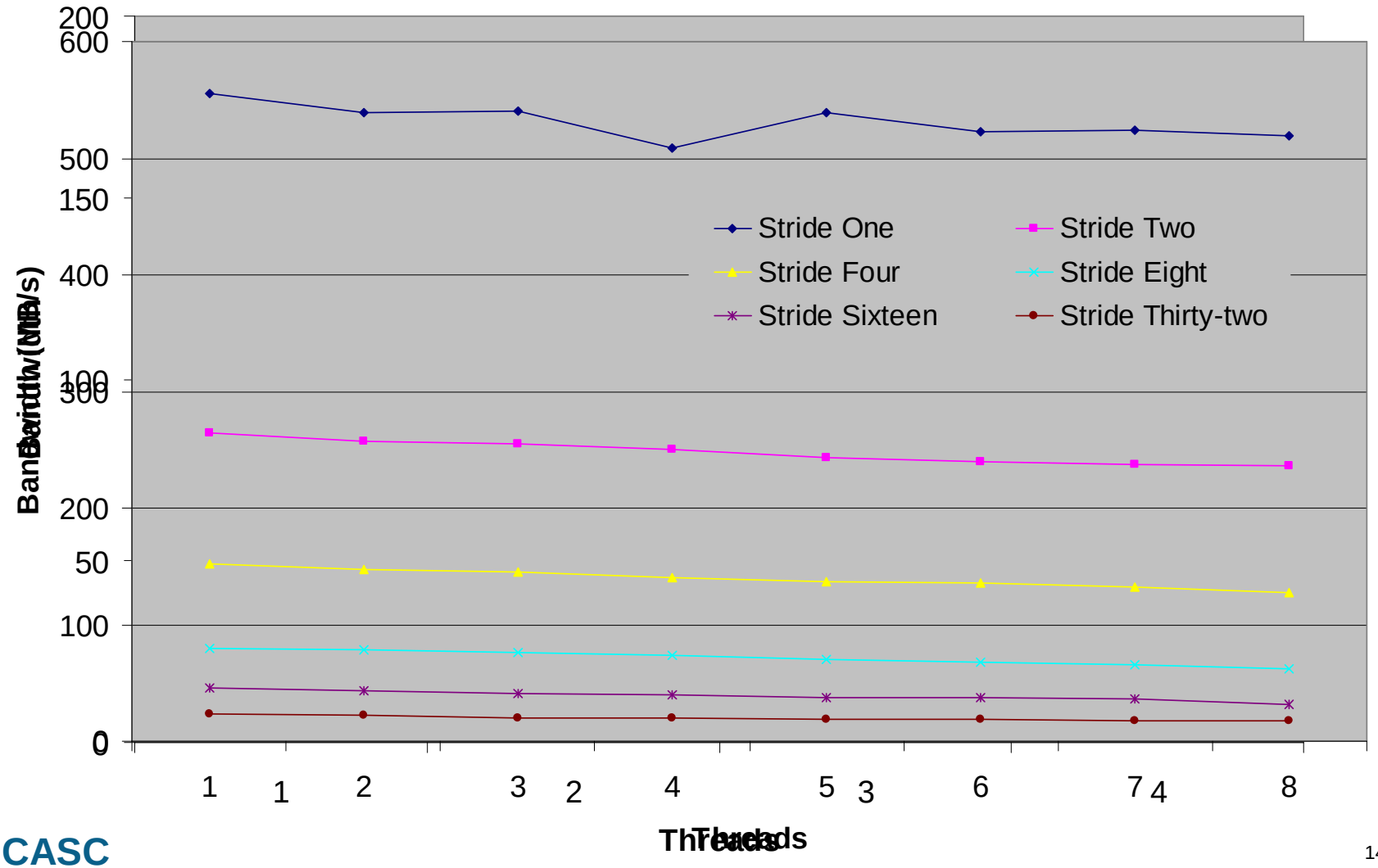


Finding Architectural Features

```
initialize_array (p, stride,...)
PAPI_Start (L1Dmisses)/* or whatever */
for (;;) p = (char **) *p;
PAPI_STOP, etc. in exception handler;
```

- Results for L1 on Blue agree with inferences
- Use hardware performance monitors (PAPI)
- Pluses
  - Clear connection to architectural features
  - Less subject to confusing factors
- Drawbacks
  - Does not provide performance measurement
  - Depends on availability and accuracy of HPM

Effect of Varying Number of Threads



Conclusion

- First parallel memory microbenchmarks
  - Extend HBenchOS
  - OpenMP parallelization
  - New access patterns
  - Infer more architectural features through stride
  - Significant bandwidth reduction due to contention
    - Between 11% and 40% on Blue
    - Between 7% and 30% on Snow
- New tests for architectural features
  - Use hardware performance monitors (PAPI)
  - Includes instruction cache tests

Current Focuses

- Pthreads version
- Instruction cache performance tests
- Testing on additional platforms
- Access pattern extensions
  - More tests with random access patterns
  - Application based access patterns
- Additional architectural features
  - Prefetching detection/characterization
  - Write buffers, etc.
- MPI memory microbenchmarks?

UCRL-VG-145223

Work performed under the auspices of the U. S. Department of Energy by University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48