Latency-Tolerant Runtime System for Large Graph Computations

Jacob Nelson, Brandon Myers, Andrew Hunter, Luis Ceze, Carl Ebeling, Dan Grossman
University of Washington

Simon Kahan
Pacific Northwest National Laboratory

http://softxmt.cs.washington.edu

Pacific Northwest NATIONAL LABORATORY

sampa

University of Washington Computer Science & Engineering

Modern processors depend on **locality** for performance

Many important problems have little locality, leading to inefficient execution at scale

Example: **Social networks**

Multithreading helps hide latencies that lead to inefficient execution

Example: Madduri et al used special-purpose latency-tolerant hardware (500MHz Cray XMT) to do a centrality analysis of the IMDb movie actor database (1.54M vertices, 78M edges)

Result: 4.75x speedup over 2.4GHz Xeon system

Can we get similar speedups with commodity hardware?

Example: **parallel graph traversal**

```
explore (Person p) {
  for each f in p.friends {
    if (atomic_increment(&f.visited) == 0) {
      /* f not yet visited */
      explore (f)
    }
  }
}
```

For large graphs, this is almost always a cache miss

Key Components

Lightweight synchronization

Problem domain has high communication to work ratio

Approach: associate a lock with every word;
provide operations to manipulate data and lock atomically

Fast Context Switching

Enable memory concurrency by overlapping computation with memory requests from multiple threads

Approach: switch at user level and save minimal state
schedule based on memory request completion

Memory Concurrency

Support enough outstanding memory operations to saturate processor memory bandwidth

Approach: prefetch before context switch;
use coprocessor to extend prefetch depth

Implementation Tradeoffs

Software-only

- methods to provide lightweight synchronization:
  - non-standard datatype
  - multiple atomic operations per sync op

Hardware/Software

- coprocessor can provide synchronization ~ cost of remote memory reference
- coprocessor can help with prefetch and reschedule, or support more remote outstanding memory ops