# Wildfire

## A Scalable Path for SMPs

Alex Edelsburg

Duke University
Pratt School of Engineering

February 12, 2010

# Table of contents

# Overview

Distributed Shared Memory (DSM) prototype
Multiple SMPs (MSMP)

## Key ideas

- Replicate shared data across SMPs
- Schedule sharers in same SMP

# Motivation

SMP bandwidth will eventually peter out
NUMA - poor performance on real comms patterns: producer/consumer & migratory data

Different question - SMPs spanning multiple boxes/cabinets

- Want to plug together a few large SMPs
- Binary compatibility with the SMPs
- Little extra cost, same OS

# Goals

Efficiently tie together large SMPs (MSMP)
Maintain compatibility & performance
Create extra node locality
SMP-like performance
No NUMA-specific optimizations

# Goals

## Application-transparent locality using

1. Coherent Memory Replication (CMR)
   Replicate shared data across SMPs
2. Hierarchical Affinity Scheduling (HAS)
   Schedule sharers in same SMP

# The System

- Connect 2 to 4 Sun SMP servers
- Custom interconnect hardware - leverage hooks in architecture
- Switch between CMR and cc-NUMA at node or page granularity

# Cache Coherence

Traditional MOSI write-invalidate
Fully mapped directory
1 bit for each node in system
2 bits for owner node

# Cache Coherence

## Deterministic Directory

Directory & cache state always in agreement

- Simple, no corner cases
- Easy to verify
- Implemented in silicon

# CMR

- Version of Simple Cache-Only Memory Architecture (S-COMA)
- Allows SMP to allocate local, “shadow” physical pages that correspond to remote physical pages
- Promoted from default (cc-NUMA) using hardware counters that detect sharing patterns

# CMR

If remote page is changed, "shadow" becomes stale
Must translate local addresses to global before sending request

- LPA2GA Local Physical Address to Global Address
- GA2LPA Global Address to Local Physical Address

# HAS

Intelligent assignment of threads to processors/nodes

| | |
|---|---|
| Same processor | Warm local caches |
| Same node | Node locality |
| Different node | Worst case<br>Some data still replicated by CMR |

# Latency Comparison

**Bottom line**

Does almost as well as a tuned NUMA machine

Fig 3 Best performance with less than 28 nodes. No non-local accesses

Fig 4 Large directory cache in SRAM. Allows for faster lookup across nodes

Fig 5 Less likely to have cache-to-cache misses across large nodes

# Application Comparison

Large commercial OLTP benchmark

- Intense shared-data updates
- High cache-miss ratios
- High memory traffic

**Results**

With optimizations on, Wildfire is *within 13% of ideal!*

# Application Comparison

- Although more inter-node messages required on a miss, less bandwidth due to intra-node locality
- 75% created locality - indicates effective procedures
- CMR is a win even if limited only to certain “hot” regions

# Questions

- Why were the authors concerned with bandwidth of SMP backbone?
  Increased faster than Moore's Law for 10 years
- How fair are their thin and fat NUMA estimates?
  Seem fair. Show behavior close to ideal
- Could a more complex inter-node coherence protocol net even better performance?
- Can we expect to see ideas like this in future products (e.g. Larrabee)?