MPI Task Placement on Multicores

Douglas Doerfler
Contributions from: Kevin Pedretti, Mahesh Rajan, Carter Edwards, Courtenay Vaughan, Mike Heroux

SESS Seminar
April 8th, 2008

SAND2008-2311P
Unlimited Release
Printed April, 2008

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.
Introduction

- This is a follow-on talk to an earlier “multicore” seminar
- That talk focused on different processor architectures and some early performance evaluations focusing on “MPI everywhere” and some initial MPI vs. Threads comparisons
- Since then, we have been looking at task placement, and more MPI vs. Threads analysis (but not covered today)
- Question: What are the issues with MPI rank to core placement? Impacted features:
  - Performance
  - Power consumption
  - Runtime vs OS placement
Recap: Two “Common” Architectures

- **Common: Numbers Game**
  - 2 sockets
  - 8 cores
  - 4 memory channels
  - 4 FLOPS/clock

- **Not Common: Architecture**
  - Integrated MC vs Northbridge
  - Integrated SMP vs Northbridge
  - Unified LLC vs semi-unified LLC
  - 4 MB LLC vs 16 MB LLC

- **Not particularly comparing Intel & AMD here, but analyzing architecture tradeoffs**
Bisecting w/MPI

- **Worst Case**
  Communications is between sockets
  - For Barcelona, this is across HyperTransport
  - For Clovertown, this is through the Northbridge
- **HyperTransport Wins**
  - BW
  - Latency
**Intranode vs Internode MPI Communications**

- Intra - 1 socket/1 core
- Inter - 2 sockets/2 cores
- Clovertown can make use of “large” common L2 between cores
  - BW & Latency
  - However, very small practical benefit
- Dip at 1MB message size is real for Barcelona, why?
CTH Sensitivity to Core Placement

Rank <-> Core Mapping
HPCCG Sensitivity on Clovertown

- Only 4/8 cores useful
- Picking right 4 important
- ~50% difference
- Power savings
Conclusions

• MPI rank to core placement
  – Architecture matters, don’t be fooled by numbers
  – For the few applications we’ve examined,
    • if all cores are to be utilized, task placement may not matter as our applications generally only run as fast as the slowest core
    • If < N cores are used, e.g. to minimize power consumption, task placement DOES matter
  – For Linux, the OS managed placement provide the best performance

• For more information: