

Tools for Memory Performance Analysis

Goals

- Expose the hierarchy
- Show the placement and movement of data throughout runtime
 - a data-centric view may be preferable to app developers
- Provide app developers knobs to place data and control data movement
 - e.g., to avoid cache conflicts on many-cores
- Develop best practices for placing data within and using deep memory hierarchies
- Use knowledge of memory behavior to guide resource allocation and scheduling

Problems

- Vendors seem to be trying really hard to hide the deep memory hierarchy, but we expect many performance problems will be related to “poor” or “inefficient” memory use
- NUMA issues for multi- and many-cores
- Data transfer in heterogeneous architectures
- Experts can use existing tools and techniques to identify and resolve memory-related problems, but application developers and computing center support staff need significant training

A Path to Better Memory Tools

- Develop a taxonomy of memory problems (see Georg's paper) and a methodology (or workflow) for:
 - starting with a high-level performance profile,
 - identifying memory-bound code,
 - analyzing data structure access patterns,
 - and producing suggestions for remediations, data transformations, or algorithmic changes
- More (and better) training for app developers
 - my idea - clone Georg and Dave Lowenthal
- Ground truth microbenchmarking using mini-apps or kernels to derive "ideal" memory hierarchy performance
 - studies to show how performance is affected by scaling and concurrency
- Encourage use of scientific libraries and have the experts optimize them