Shoal: self-tuning for memory allocation and access

Auto-tune memory allocation based on access patterns and hardware characteristics → helps programmers to tackle the complexity of multicores

Problem

multicores: complex memory subsystems
- non-uniform memory access (NUMA)
- complex interconnects
- accelerators (Xeon Phi)
- features (DMA, large/huge page)
- global address space?
- cache coherence?

→ Hard to program

Ideas

Auto-Tune Memory Layout
- annotate memory allocation
- memory access patterns
  - from high-level languages
  - manually
- use hardware characteristics
- memory abstraction

Abstraction
- decouples memory from program logic
- specialized arrays (replication, partitioning ..)
- high-level operations (copy, memset ..)
- support for large pages, DMA engines

Results

throughput of memory controllers

scalability pagerank (Twitter)

scalability hop_dist (Twitter)

scalability triangle_counting (soc-LiveJournal)

DMA engine

performance array implementations

Conclusion

- automatic scalability
- good performance on various NUMA machines
- no programmer efforts

Future work

- synchronization
- scheduling
- accelerators (Xeon Phi)
- multiple address spaces