Algorithmic scalability and load balancing

DOE/DOD Workshop on Emerging High Performance Architecture and Applications

Breakout summary

David A. Bader and John Johnson

Two types of communities

1. Existing codebases, typically physical simulations
   - Validated algorithms
   - Flywheel effect of developing architecture s that self-select these applications
2. Emerging applications in data and discrete sciences
   - Algorithms are still under development
   - Ample concurrency, yet not exploited
   - Community may not have the HPC expertise or investment

Application Scalability Challenges

- Bob Lucas
  - Multiphysics requiring two types of load balancing
  - Existing software base
- John Gilbert
  - Latency
  - Long sequential dependencies in graph algorithms
  - Lack of locality
  - Lack of parallel data structures
- Sam Williams
  - Complexity and portability; choosing optimizations
  - Memory performance
- Martin Berzins
  - Load balancing, MPI
  - Productivity / efficiency
- Sheri Li
  - More scalable algorithms are less effective
- David Bader
  - Parallel data structures / algorithms for dynamic, complex networks
  - Lack of locality
- Jean-Paul Watson
  - Non-local
  - Dynamic nature of graphs
  - Adaptive, Dynamic load balancing are problem sensitive
  - Parallel data structures
- Brad Miller
  - Combinatorial optimization
  - Gave up on communication between data and replicated data

Graphs & Complex Networks

Kernels that are mentioned as common to multiple domains:

- Problem-dependent Structure
  - Load balancing
- Connectivity queries
  - Sequential
- Betweenness and other characteristics
- Sparse matrix problems
- Clustering
- Subgraph isomorphism
- Belief cut
- Clique finding
- Bipartite network flow
- Partitioning / Decomposition
- Junction Tree Construction
- Feature extraction

Impediments to using today's HPC

- Parallel algorithms
- Unknown structure
  - Degree of Structure
  - Degree of Change (dynamic)
  - Lack of locality
- Partitioning / Load balance
- Temporal changing
- Large, shared memory
- Data preparation of noisy data
- End-to-end, streaming
- High degree vertices require sweeping large data sets
- Lack of standard body of "primitives", requires more prototyping and exploration of techniques
- Portability
- Existing infrastructure may be too complex to explore new algorithms
  - High cost of entry to HPC