Characterizing Data Locality in Parallel Graph Algorithms
Nicole Rodia and Kunle Olukotun
Stanford University

**Graph Analytics**
- Important component of data mining, machine learning, and scientific computation
- Graph algorithms are:
  - Memory intensive
  - Expensive, e.g. $O(n + m)$
  - Poor performance on multicore CPUs

**Simulation-based Analysis**
- Algorithms implemented in Green-Marl DSL [1]
- Social network, web link, road, FE mesh, and synthetic graphs datasets
- Multiprocessor simulation with multi-level memory hierarchy (zsim) [2]

**The Memory Wall**
- Increasing gap between fast computation and slow data access
- Modern machines use large multi-level caches to compensate for limited memory bandwidth
- **Execution time is dominated by data access**
  - Random access and large data size heavily utilize memory BW
  - Low computation-to-memory access ratio unable to hide latency

**Data Cache Performance**
- Cache lines poorly utilized $\rightarrow$ wastes memory BW
  - ↑ cache size $\rightarrow$ ↓ cache MPKI
  - Dataset size and structure affect cache MPKI
  - **Miss rates not reliable measure of data locality**

**Data Locality**
- Temporal locality: access same location again in future
- Spatial locality: access nearby location in future
- Long reuse accesses poor fit for LRU replacement policy

**Per-Data Structure Locality**
- Informatics Graph Properties
  - Small-world: $O(\log n)$ diameter
  - Scale-free: Power-law degree distribution
  - Many vertices with very small degree
  - Few vertices with very large degree
  - **Average reuse distance correlates with vertex degree** for data structures with non-uniform access patterns

**Future Work**
- Leverage algorithm, data structure, and graph dataset properties for a more effective cache replacement policy that better captures data locality and thus improves performance
- Incorporate fine-grained data access to reduce unused cache space and improve spatial locality

**References**