Collaborative Research: Adaptive I/O Stack for High End Computing

Xiaosong Ma*, Vincent Freeh, John Blondin (NC State U.)
Yuanyuan Zhou (U. of Illinois)
Anand Sivasubramaniam (Penn State U.)

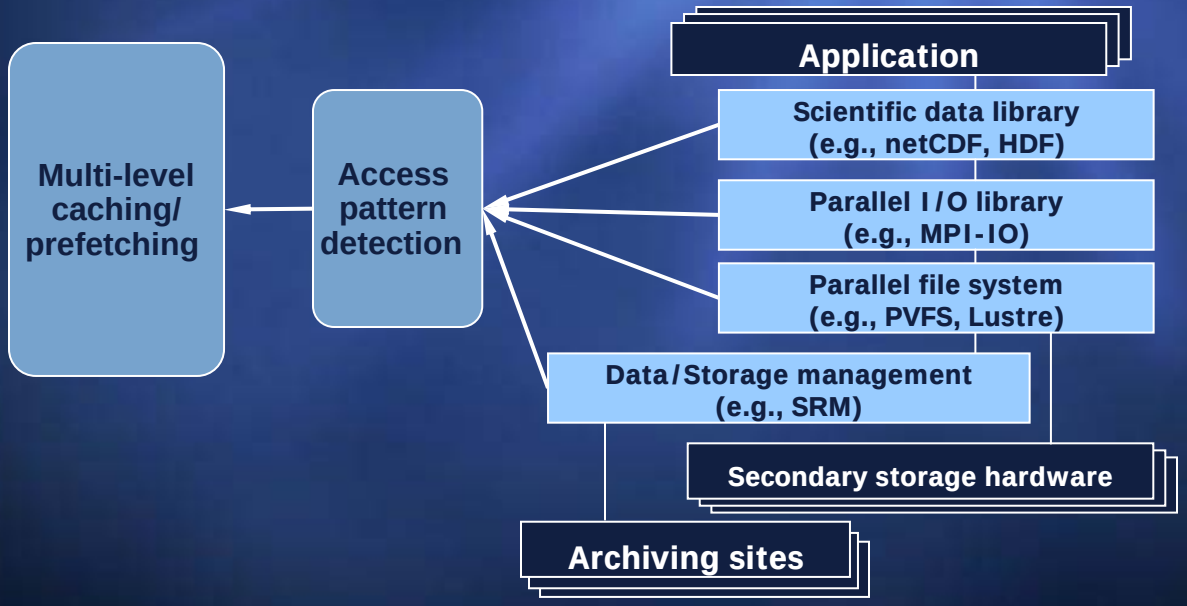
(* Joint faculty with Oak Ridge National Lab)

Problem Overview

- Rapidly increasing I/O needs of scientific applications
  - Increasing performance and scalability gap
  - I/O and storage often limiting factor of application scale
- HEC I/O stack lacks flexible application support
  - System software and tools extended from sequential counterparts
  - Little optimizations/customizations targeting individual apps
  - Little coordination between I/O stack layers
    - High-level I/O libraries
    - Parallel I/O libraries
    - Parallel file systems
    - Storage/data management systems
- Result: unsatisfactory performance + wasted resources

We Propose

- PATIO (Parallel AdapTive I/O) Framework
- Focus
  - Automatic access pattern recognition
  - Multi-layer caching/prefetching



Methodology

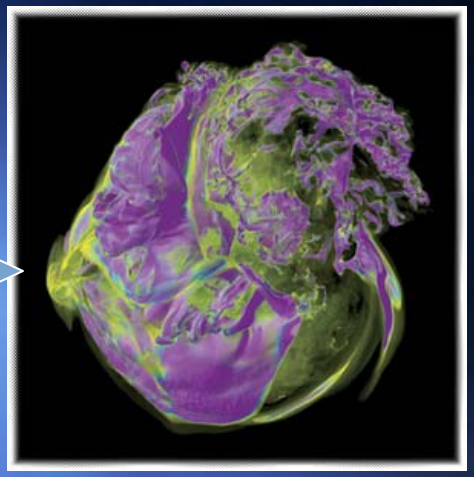
- Automatic access pattern recognition
  - Mining regular, complex, and correlated patterns
  - Needs to be adapted for HEC
    - Different data granularities at different I/O stack layers
    - Per-job analysis
    - Good news: long running jobs, repetitive behaviors
- Multi-layer caching/prefetching
  - Memory utilization increasingly crucial
  - Different I/O stack layers have their own strategies
  - Problem: varied lower-level access pattern, redundant operations
  - Proposed solution: vertical collaborative caching
    - Adaptive selection of algorithms and cache configurations
    - Effective inter-layer communication

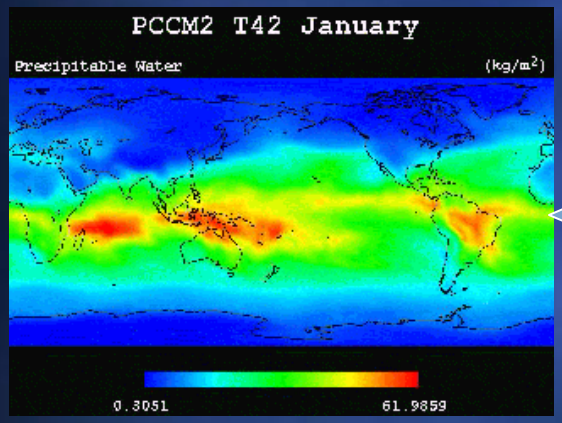
Evaluation

Common application characteristics:

- Use deep I/O stack
- Fast growing data needs

Terascale Supernova Initiative (TSI)
- NCSU & others, DOE SciDAC
- 3D multi-scale, multi-physics simulation
- Substantial allocation at ORNL's NLCF
- 300-timestep run generates 6TB
- Sequential & non-sequential accesses
- netCDF files





The Climate Consortium
- ORNL & others, DOE SciDAC
- IPCC model runs
- Variety of access patterns
- netCDF files

Spallation Neutron Source (SNS)
- ORNL
- PBs of data by 2010
- Diversed user community and access patterns
- Data center
- NeXus files

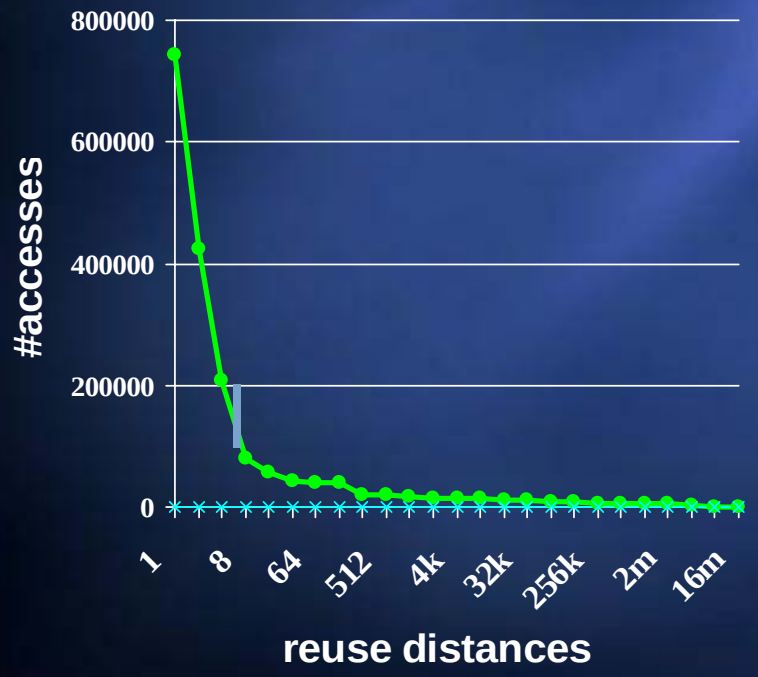
Prior Work: Level 2 Cache Access Characteristics

For commercial workloads [USENIX'01]

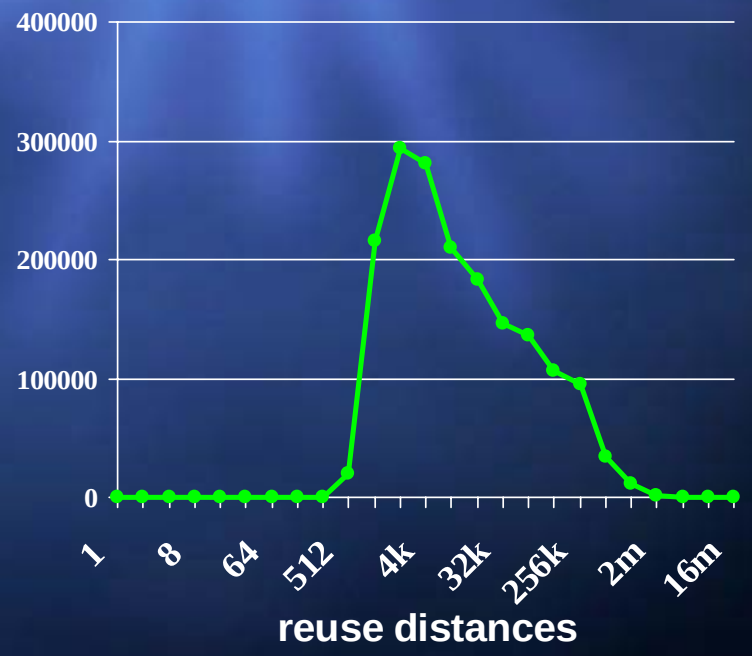
(1) Accesses to 2nd level buffer cache have poor temporal locality

(2) Accesses are not even distributed among blocks

1st level Buffer Cache



2nd Level Buffer Cache

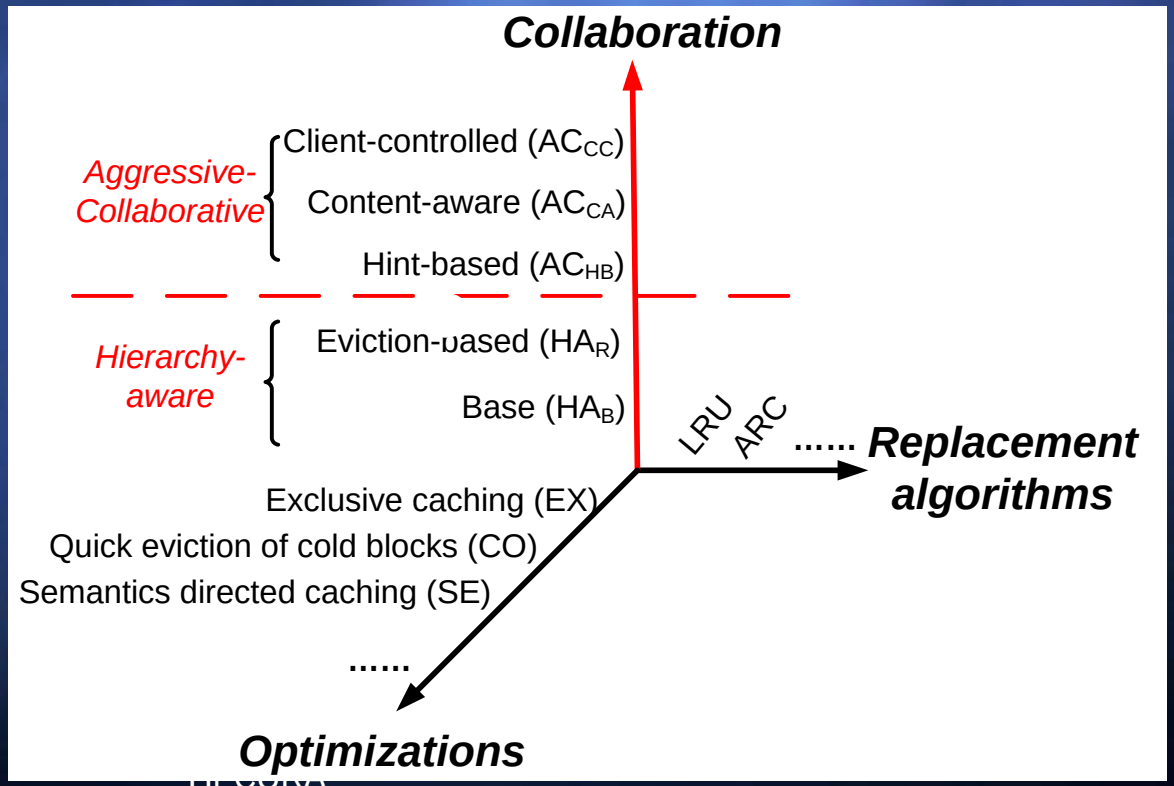


Prior Work: Multi-level Buffer Caching

- Multi-level buffer cache management
  - MQ cache replacement algorithm [USENIX'01]
  - Eviction based cache placement [USENIX'03]
- Goal: manage the buffer caches collaboratively as if they are in a unified buffer cache
  - i.e, equivalent to a global LRU managing all buffer caches
- Results:
  - MQ outperforms all previous replacement algorithm
  - Eviction + MQ gives the best buffer cache hit ratios

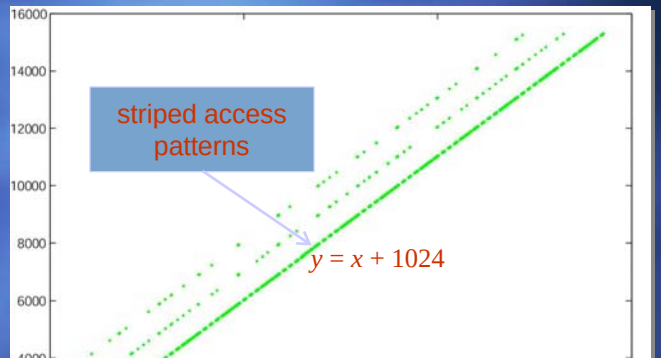
Prior Work: Collaborative Buffer Caching

- Empirical evaluation of 248 combinations of collaborative buffer caching for commercial workloads [SIGMETRICS'05]
  - Hierarchy-aware caching is enough, and there is not much need for aggressive collaborative buffer caching



Prior Work: Mining for Access Patterns

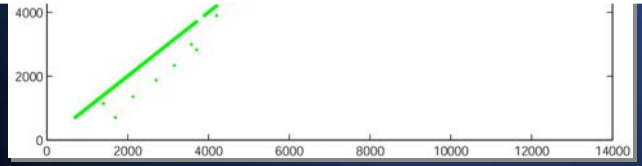
- Idea: using data mining & statistic ideas to automatically extract access patterns from I/O traces
  - Simple patterns: temporal/spatial locality, hot blocks, etc
  - Complex patterns: block correlations, etc



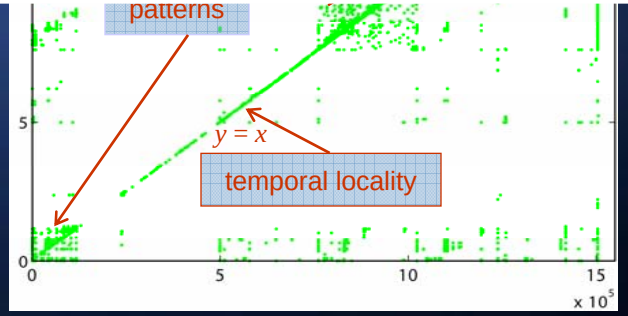
A major research question:

(1) Are these ideas and findings applicable to scientific I/O workloads?

(2) If not, how to revise them?



Loop-sequential Trace (1X10^6 requests)



Cello Trace (6.6X10^6 requests)