Architecture Conscious Data Mining

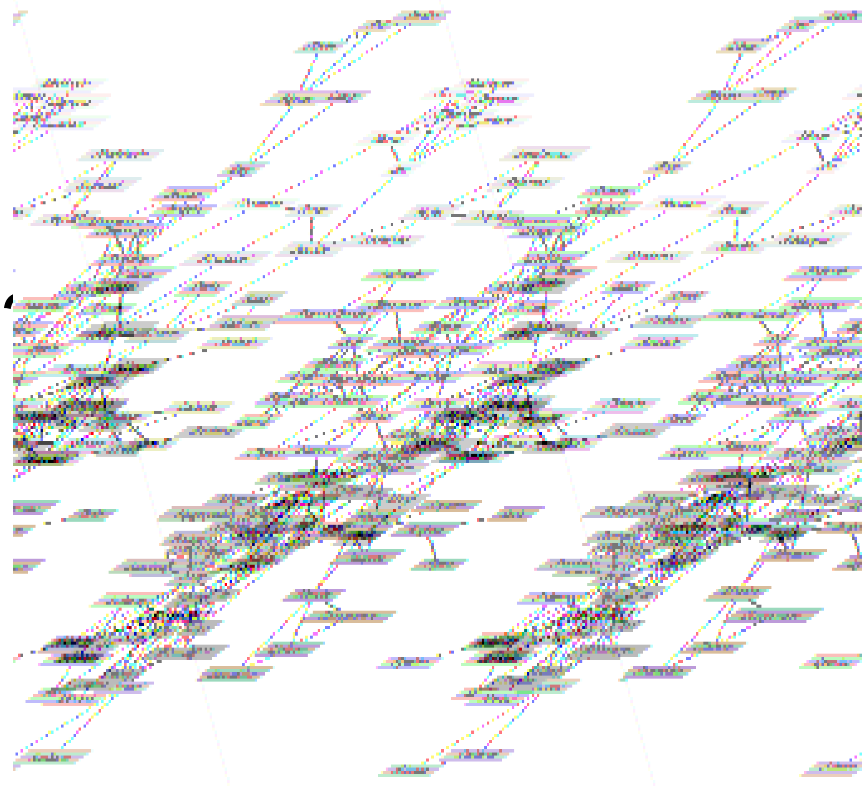
Srinivasan Parthasarathy
Data Mining Research Lab
Ohio State University

KDD & Next Generation Challenges

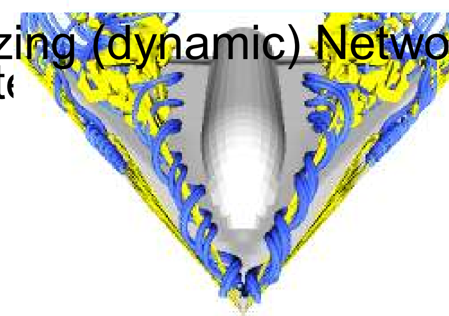
- KDD is an iterative and interactive process the goal of which is to extract **interesting** and **actionable** information from potentially large data stores **efficiently**
- Young field, long laundry list of technical challenges
  - Theoretical foundations in various sub-fields
  - Interestingness and Ranking
  - New and Exciting Applications
    - Embedding domain knowledge effectively
  - Visualization for data & model understanding
  - Efficient and scalable algorithms (focus of this talk)
- Other challenges
  - Educational (talk a bit about this at the end)
  - Reproducability (need for benchmarks)
  - Socio-Political

Efficiency in the KDD process

- Why is it important?
  - Interactive nature of KDD
  - Real-time constraints
- What makes it challenging?
  - Dataset properties (large, heterogeneous, distributed)
  - Computational complexity
- Example Applications
  - Clinical data
  - Biological data
  - Large scale simulation data
  - Social network data
  - Sensor data, WWW data….



Analyzing (dynamic) Networks
Prote k (yeast)



Toward Efficient Realizations

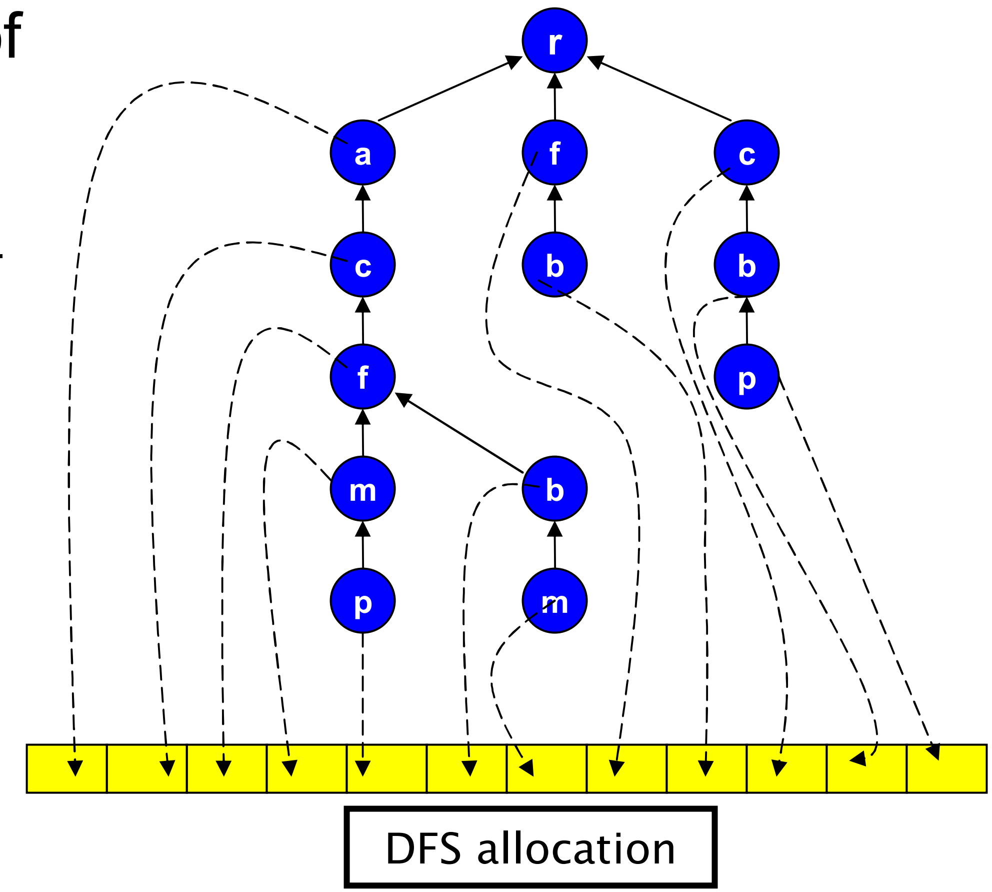
- Data driven approach
  - Compression, Sampling, Dimensionality Reduction, Feature Selection, Matrix Factorization etc.
- Computational driven approach
  - Intelligent search space pruning to reduce complexity
  - Approximate algorithms, streaming algorithms
  - Parallel and distributed algorithms
- Architecture-Conscious approach (this talk)
  - Largely orthogonal to the above alternatives
  - Objective is to understand limitations and novel features of modern and emerging architecture(s)
  - Subsequently, re-architect algorithms to better utilize system resources.

Houston, do we have a problem?

- Turns out we do
  - Many state-of-the-art data mining algorithms grossly under-utilize processor resources [Ghoting 2005]
- Why?
  1. Data intensive algorithms – lots of memory accesses – high latency penalty.
  2. Mining algorithms are extremely irregular in nature – data and parameter driven – hard to predict
  3. Use of pointer-based data structures – poor ILP
  4. Do not leverage important features of modern architectures – automated compiler/runtime systems are handicapped because of 1, 2 and 3.

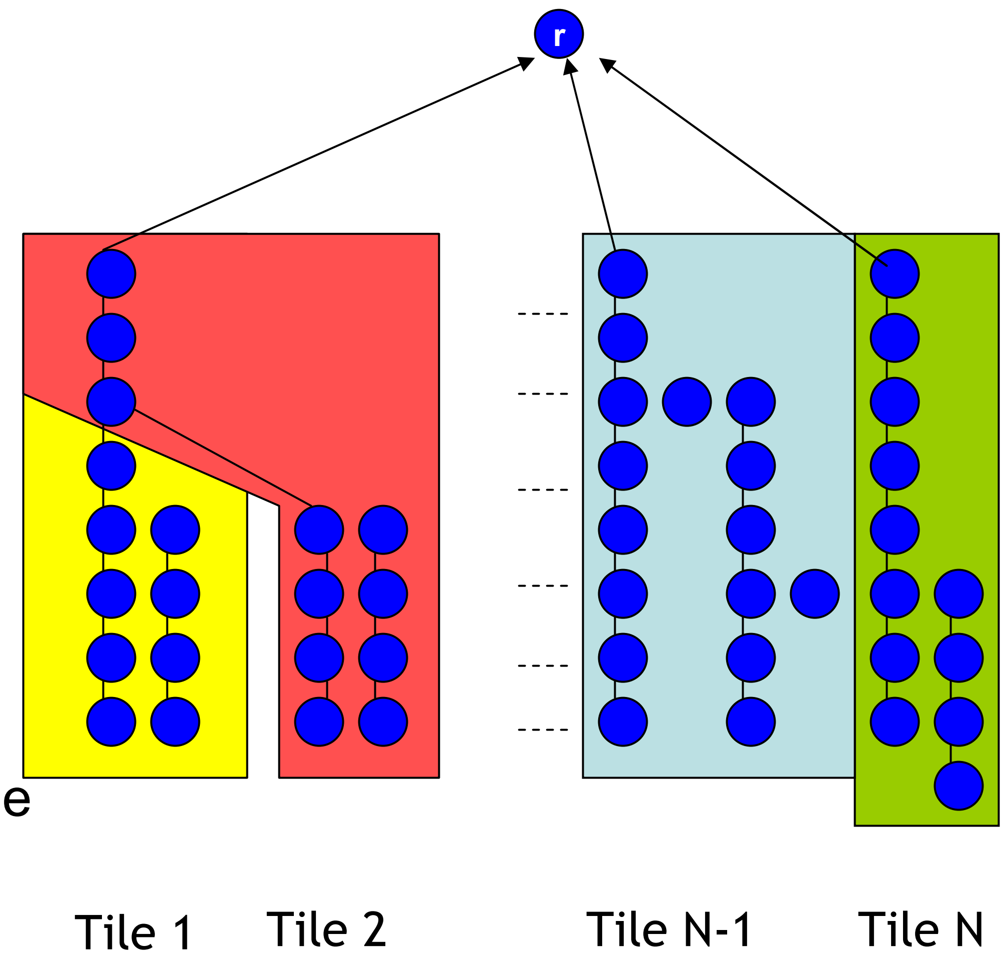
Spatial Locality

- Improve spatial locality of dynamic data structures
  - Memory pooling
  - Loss-less compression – store only data that is needed – allows for more data per cache line
  - Memory placement to match dominant access order
  - Side benefit – enables effective hardware prefetching (latency alleviating mechanism)



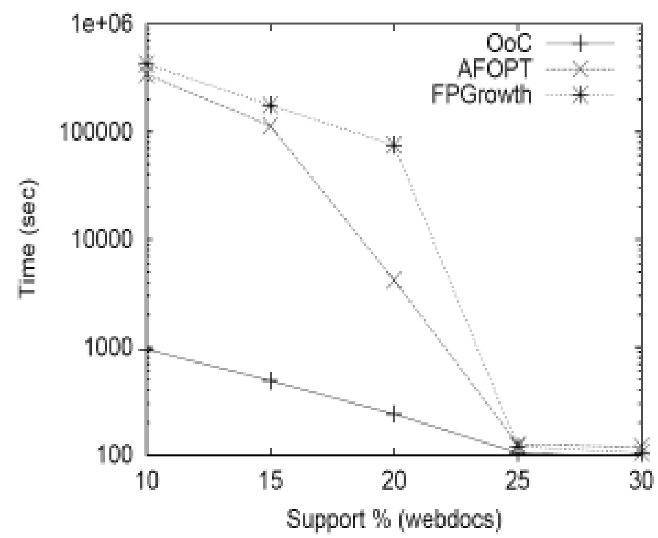
Temporal Locality and Leveraging SMT

- Data Structure Tiling
  - Operate on a tile-by-tile basis
    - Non-overlapping (traditional)
    - Overlapping
- Smart data partitioning
  - Jigsaw puzzle analogy
- SMT
  - Co-schedule tasks that operate on same data tile helps improve performance



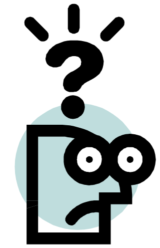
Sample Benefits

- Gains in performance can be staggering
  - Frequent patterns (itemsets, trees, graphs)
  - Outlier Detection
  - Clustering
- Benefits to end applications
  - Scientific simulation data
  - Web data
  - Molecular and Clinical data
- For network of workstations
  - minimize communication and leverage remote memory
  - Enables mining of terabyte scale distributed datasets efficiently.



VLDB'05, KDD'06, VLDBJ07
PPOPP'07

CMPs (next frontier)



- Why the push from industry?
  - Increasing clock frequencies is not returning improved IPC, and it is increasing power costs and thermal issues
- Two new PCs in my den, no need for the heat vent!
  - Great for winters!
- Importantly
  - Parallel Computing meets mainstream commodity market
- Challenges
  - Existing applications, they need to be rewritten to use multiple threads of execution
  - Compiler and runtime techniques have a hard time already – application must help
  - Fine-grained sharing of processor resources (cache, bus/channel etc.)
  - Memory hierarchy issues are even more challenging
- Potential solution
  - Adaptable algorithms

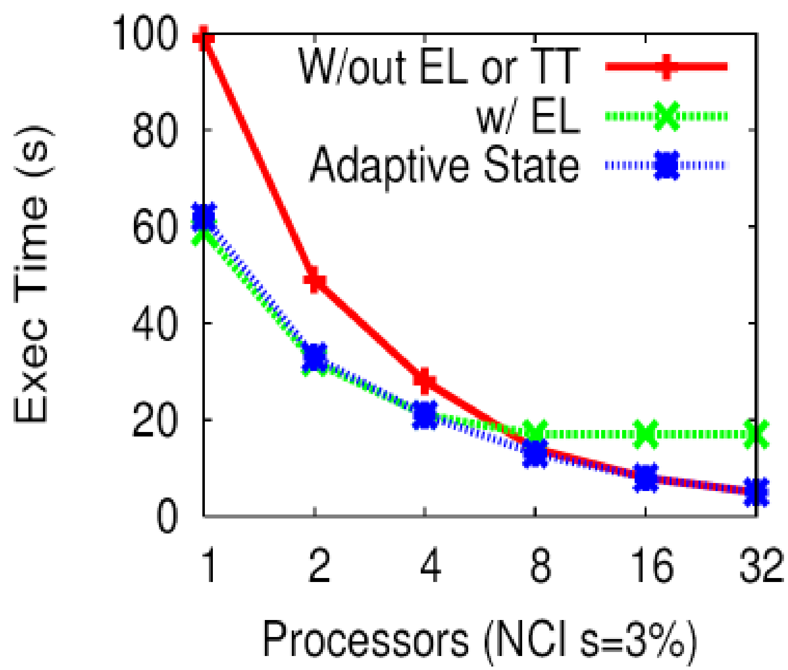
Adaptive algorithms

- Key idea:Trading off memory for redundant computation
- Benefits:
  - Reduced working set sizes
  - Likely to have reduced bandwidth pressure
  - Utilizing strengths of the CMP
- Challenge:
  - Sensing the problem
  - Re-architecting algorithm to reduce memory consumption

- Key idea: Moldable partitioning and adaptive scheduling of tasks
- Benefits
  - Better CPU utilization
  - If co-scheduling – reduced cache miss rates
- Challenges:
  - Sensing the problem
  - Re-architecting algorithm
    - Moldable task decomposition
    - Pass on enough state to move task to another core

Adaptive algorithms performance

- Graph mining
  - Gaston vs. Gspan vs. Hybrid (adaptive)

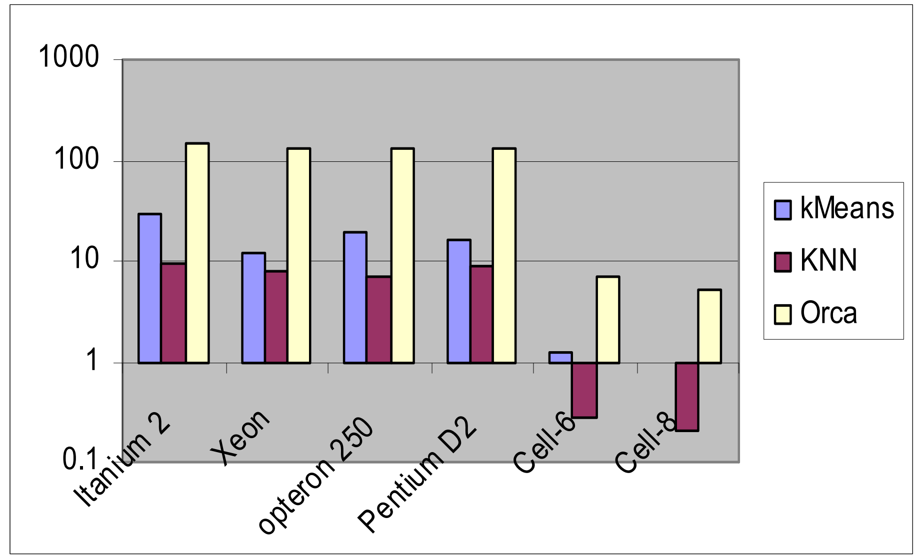


- Tree Mining
  - Converted to sequence space (dynamic arrays)
    - Better locality, ILP
  - Reduced memory LCS matching + structure checks
  - Leveraged hybrid scheduling
  - Sequential Performance
    - **2 order reduction in memory footprint**
    - **3 orders improvement in processing time**
  - Parallel Performance
    - Linear scalability on a 4-core dual chip (8 cores)
    - Adapted similar idea to XML indexing with similar results!

ICDM’06, CIKM’06, VLDB’07

Esoteric CMPs (CELL)

- Interesting design point on commodity CMP space
  - 25 GB/s OC bandwidth
  - 8 cores (SPUs) + 1 PPU
  - FP computation 200 GFlops
  - Breakthroughs in commodity processing
- Challenges
  - Hard to program
  - Need to explicitly manage memory and data transfers between PPU and SPUs
  - Probably not suitable for all programs
  - Interesting class of algorithms and kernels can benefit significantly!



Cell-6 on Sony Playstation
Cell-8 is simulated
All cases codes optimized and
Implemented on appropriate compiler

Mining on Clusters

- Heavily researched over the last 15 years
  - DDM Wiki (a very nice start point resource)
- What are the "new" challenges?
  - Non-homogeneous "hybrid" clusters – (e.g. Roadrunner)
  - Multi-level parallelism (on chip, on node, on cluster)
  - Leveraging features of high end systems networking
    - Infiniband makes it feasible and cheaper to access remote memory than local disk – how to leverage?
  - KDD may be particularly amenable to pipelined parallelism – a largely ignored approach
  - KDD and the grid (heard about this yesterday)
  - Application specific challenges -- e.g. astronomy, folding@home etc.

Discussion

- KDD is an iterative and interactive process the goal of which is to extract **interesting** and **actionable** information from potentially large data stores **efficiently**
- This talk was primarily about the last but all 3 are important.
- Architecture conscious data mining is a viable orthogonal approach to achieve efficiency (references in paper)
  - Tangible benefits to applications, algorithms and kernels
  - Lower memory footprints + significantly faster performance
  - Adaptive algorithms are necessary for emerging architectures
  - Whats next? Services oriented architecture
    - Plug-and-Play naturally connects with KDD process
    - An effective mechanism to keep cores busy.

Broadly Speaking

- Education
  - As an aside parallel algorithms and high performance computing has to be a part of basic CS curriculum.
  - We as data-intensive science need to understand the key systems issues better from OS and architecture friends
- Broader Scientific Impact
  - Interactions between Systems and Data Mining
    - Data mining for software engineering, invariant tracking, testing, bug detection in sequential and parallel codes
    - Data mining for performance modeling
    - Leveraging systems features for data mining

Thanks

- Students
  - A. Ghoting, G. Buehrer, S. Tatikonda
- Collaborating Colleagues
  - OSU-Physics, OSU-Biomedical Informatics, Intel, IBM
- Funding agencies
  - NSF CCF0702587, CNS-0406386, CAREER-IIS-0347662, RI-CNS-0403342.
  - DOE Early career principal investigator grant
  - IBM Faculty partnership
- Organizers of this workshop
- Additional Information: dmrl.cse.ohio-state.edu or srini@cse.ohio-state.edu