Knowledge and Cache
Conscious Data Mining: Algorithms and Systems Support

Srinivasan Parthasarathy
srini@cse.ohio-state.edu

Joint work with A. Ghoting, G. Buehrer, M. Goyder,
S. Tatikonda, T. Kurc and J. Saltz

NSF NGS CNS0406386
Motivation

• Advances in technology $\rightarrow$ huge data collections
  – Sensor networks
  – Massive legacy data in business or financial settings
  – Large scale simulations
  – Homeland security applications
  – Biomedical imaging
  – Bioinformatics
Knowledge Discovery Process

- Knowledge discovery and data mining
  - Goal: extracting useful and actionable information (models, rules, patterns) from such massive data stores.
  - Multi-billion dollar industry
- Time consuming process – Compute and Data Intensive
- Human-in-the-loop (verification) – Interactive
- Impedence Mismatch!
Next Generation Data Analysis

• Potential Solution: Leverage commodity high performance computing solutions to resolve this impedance mismatch.
  – Services oriented architecture
    • Scheduling Services
    • I/O and Data Services
    • Knowledge and Data Caching Services
  – **Algorithms that can leverage such services**

• Challenges
  – Highly irregular – very data and application dependent
  – Often rely on large meta-data housed in dynamic data structures
    • Used to prune search space (pointer-based)
    • May be out-of-core!
  – Data is also often dynamic (time varying)
Key Idea: Predicting and Exploiting Re-Use at Multiple Levels

• **Cache Conscious Data Mining**
  – At the algorithmic level
  – Improve spatial and temporal locality through careful understanding of (repetitive) access patterns
    • Leverage memory placement and data structure partitioning
  – Leverage architectural features (e.g. SMT) effectively to hide latency
    • Co-schedule threads that work on same data (different tasks)

• **Knowledge Conscious Data Mining**
  – At the methodological level
  – Leverage the iterative and interactive nature of process
  – Store and re-use previously computed knowledge to drive future requests
  – Effective in collaborative data analysis tasks but also across iterations of same algorithm
Cache Conscious Tree Mining

- Applications: bioinformatics, linguistics, program analysis, bug detection, web mining etc.
- Essentially converted pointer-based trees to sequences (housed in arrays) and operated on sequence space (bijection)
- Up to 355 speedup, using 40% less memory over state-of-art

Itanium, 1.3 Ghz, 4GB Memory, CSLOGS – weblog dataset
Knowledge Conscious Clustering

- Fundamental approach with a host of applications
- Single client system. Benefits include
  - Re-use knowledge across iterations of algorithm
  - Remote (Client-side) caching of KO
  - Up to 10 fold improvement across the board
Knowledge Caching System Overview

Client Process

Knowledge Cache

Put(KO) → Get(QO)

KO – Knowledge Object
Metadata – used to determine re-use potential given QO
  linearize(…)
delinearize(…)
Knowledge – encoding of actual information
  linearize(…)
delinearize(…)

QO – Query Object
Specified by application or user
  CanReuse(KO)
  ReuseScore(KO)

Key Features
• Replacement Policy
• Associative LRU
• Supports distributed caching of KO
• Supports partial caching

Additionally we also support Data Objects (DO) – data subsets
Summary and Current Status

- Designed Cache Conscious Solutions
  - Frequent Pattern Mining (VLDB Journal 06, KDD 06)
    - Tera-scale mining (PPOPP 2007)
  - Tree Mining (CIKM 2006)
    - Parallelization in progress
- Designed Knowledge Conscious Solutions
  - Clustering (PKDD 2006)
  - Frequent Pattern Mining & Classification (in progress)
- Systems Support
  - Design in place, implementation being debugged (in progress)

Weak Scalability on Frequent Pattern Mining
- Stripped down linearize/delinearize
  - 10 fold reduction in communication
- Efficient even when meta-data is out-of-core
- Order of magnitude over state-of-art
Acknowledgements

• Other grant acknowledgements
  – NSF CAREER IIS-0347662
  – NSF RI CNS-0403342
  – IBM PhD Fellowship (A. Ghoting)

• For more information
  – http://www.cse ohio-state.edu/~srini
  – http://dmrl.cse ohio-state.edu