Scalable Data Analytics: On the Role of Stratified Data Placement

Srinivasan Parthasarathy
Data Mining Research Lab
Ohio State University

The Data Deluge: Data Data Everywhere

Data Storage is Cheap

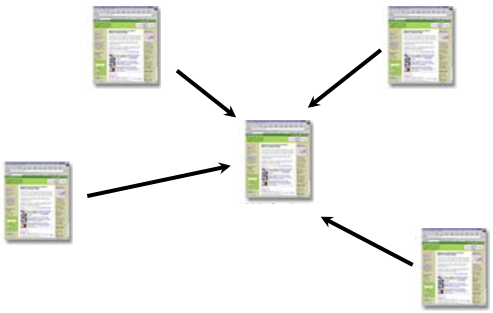
600$

to buy a disk drive that can store all of the world's music

[McKinsey Global Institute Special Report, June' 11]

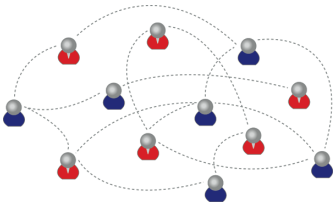
Data does not exist in isolation.

Data almost always exists in connection with other data – integral part of the value proposition.

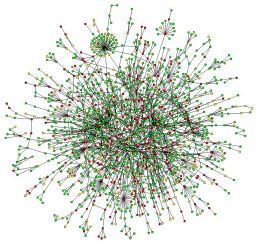


Google PageRank

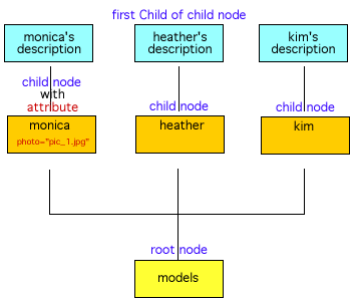
THE OHIO STATE UNIVERSITY

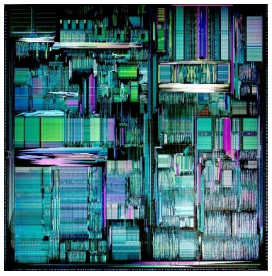


Social networks

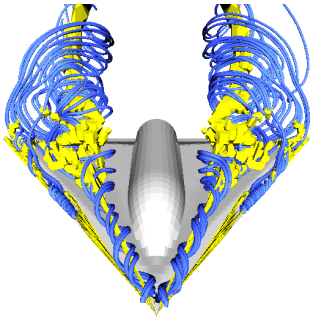


Protein Interaction

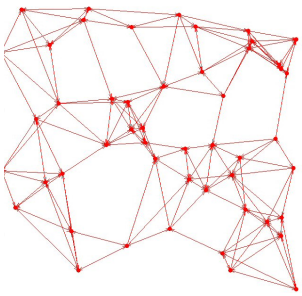




VLSI networks



Scientific Simulations



Neighborhood graphs

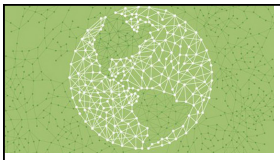
THE OHIO STATE UNIVERSITY

Big Data Problem: All this data is only useful if we can extract **interesting** and **actionable** information from large complex data stores **efficiently**

THE OHIO STATE UNIVERSITY

The Case for Stratified Data Placement of Complex Big Data

THE OHIO STATE UNIVERSITY

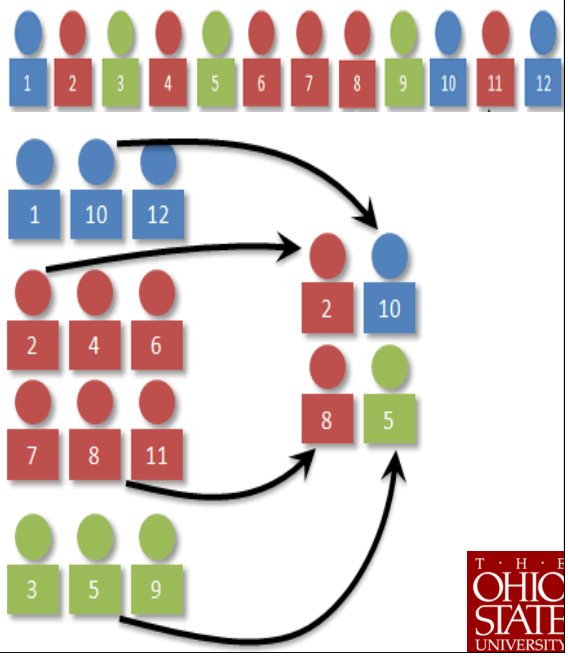


Importance of Data Placement

- Locality of reference
  - Placing related items in proximity improves efficiency
- Mitigating Impact of Data Skew
  - Must account for distributional effects on workload.
- Interactive Response Times
  - Operate on a sample with statistical guarantees
- Heterogeneity and Energy Aware
  - Heterogeneous compute and storage resources
  - Renewable energy capabilities may vary

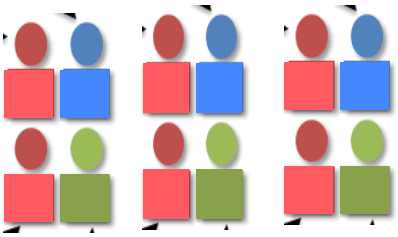
Stratified Sampling in a Slide

- Roots in Stratified Sampling (Cochran'48)
- Group related data into "homogeneous strata"
- Sample each strata
  - Proportional Allocation (shown)
  - Optimal Allocation

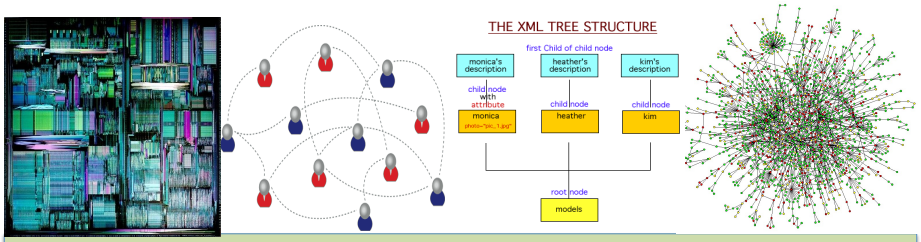


However, here we want to partition

- For Locality
  - Elements within a strata are placed together
- For Mitigating Skew
  - Each partition is a proportionally allocated stratified sample
- For Interactivity
  - Optimally allocate one partition
  - Proportionally allocate the rest
- Accounting for Energy/Heterogeneity
  - More on this later -- time permitting



Our Vision: Stratified Data Placement

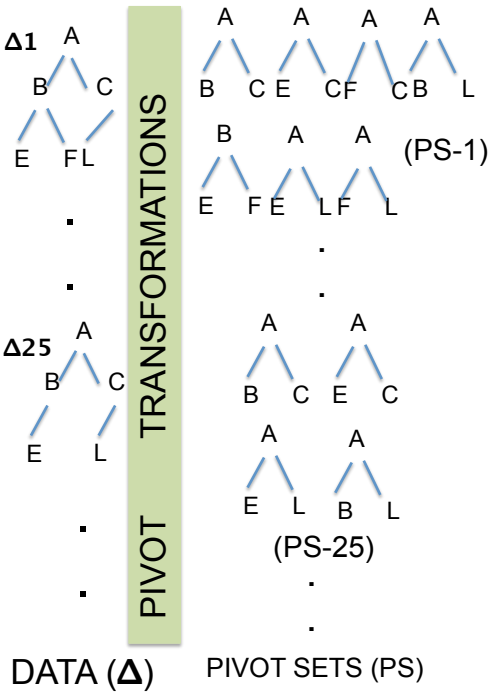


STRATIFIED DATA PLACEMENT		
HADOOP/SHARK/ Azure (HDFS/RDD/Blob)	Key Value Stores e.g. Memcached Redis	MPI & Partitioned Global Addresses Space Systems (PGAS) e.g. Global Arrays

Key Challenge: Creating Strata (of Complex Data)

- What about Clustering?
  - Non-trivial for data with complex structure
  - Potentially expensive
  - Variable sized entities

- 4-step approach [ICDE'13]
  1. Convert complex data into a (multi-)set of pivotal elements that *capture features-of-interest*
  2. Compute sketch of set (minwise hashing)
  3. Use sketches to group into strata (sketchsort/sketchcluster)
  4. Partition strata according to application needs (e.g. skew, balance)



Step 1: Pivotization

Problem: Need to simplify complex representation.

Key Idea: Think Globally Act Locally

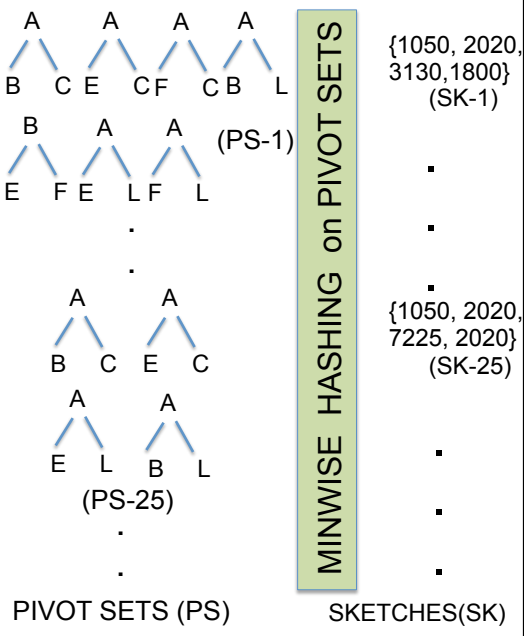
– Construct set or vector of local features that collectively captures global datum

Solution: Specific to Data & Domain

- Documents/Text
  - Shingling [Broder 1998]
- Trees (XML, Linguistic data)
  - Wedge pivots [Tatikonda'10]
- Graphs (Web, Social, Molecules)
  - Adjacency lists [Buehrer'08], Wedge Decompositions [Seshadri'11], Graphlets [Pruzlj'09]
- Spatial/vector data
  - LSH[Indyk'99, Chariker'02, Satuluri'12]
- Images/Simulation/Sequential data
  - Kernels (Leslie'03), KLSH (Kulis'2010)

Step 2. Sketching

- **Problem:** Pivot sets may be variable length, similarity computation is expensive: O(n^2)
- **Key Idea:** Use Sketching
- **Solution:** Locality Sensitive Hashing *[Broder'98, Indyk'99, Charikar'01]*
  - *Resulting representation is fixed-length (k)*
  - *Tradeoff: Representation Fidelity vs. Sketch size*
  - *Can handle kernel functions [Kulis'09] and Bayesian priors [Satuluri'09]*



Minwise Hashing (Broder et al 98)

Universe ⟶ { dog, cat, lion, tiger, mouse}

π_1 ⟶ [cat, mouse, lion, dog, tiger]

π_2 ⟶ [lion, cat, mouse, dog, tiger]

A = { mouse, lion }

$mh_1(A)$ = min (π_1{ mouse, lion }) = mouse

$mh_2(A)$ = min (π_2 { mouse, lion }) = lion

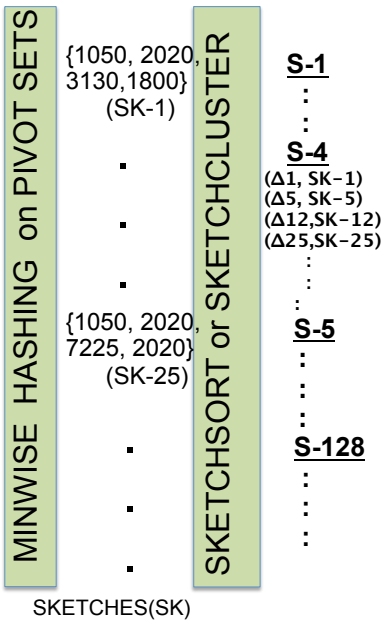
Key Fact

For two sets A, B, and a min-hash function $mh_i()$:

$$Pr[mh_i(A) = mh_i(B)] = Sim(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Unbiased estimator for *Sim* using *k* hashes:

$$\hat{Sim}(A, B) = \frac{1}{k} \sum_{i=1:k} I[mh_i(A) = mh_i(B)]$$



Step 3: Stratification

Problem: Group related entities into strata

Key Idea: Inspired by W. Cochran's work on stratified sampling [1940s]

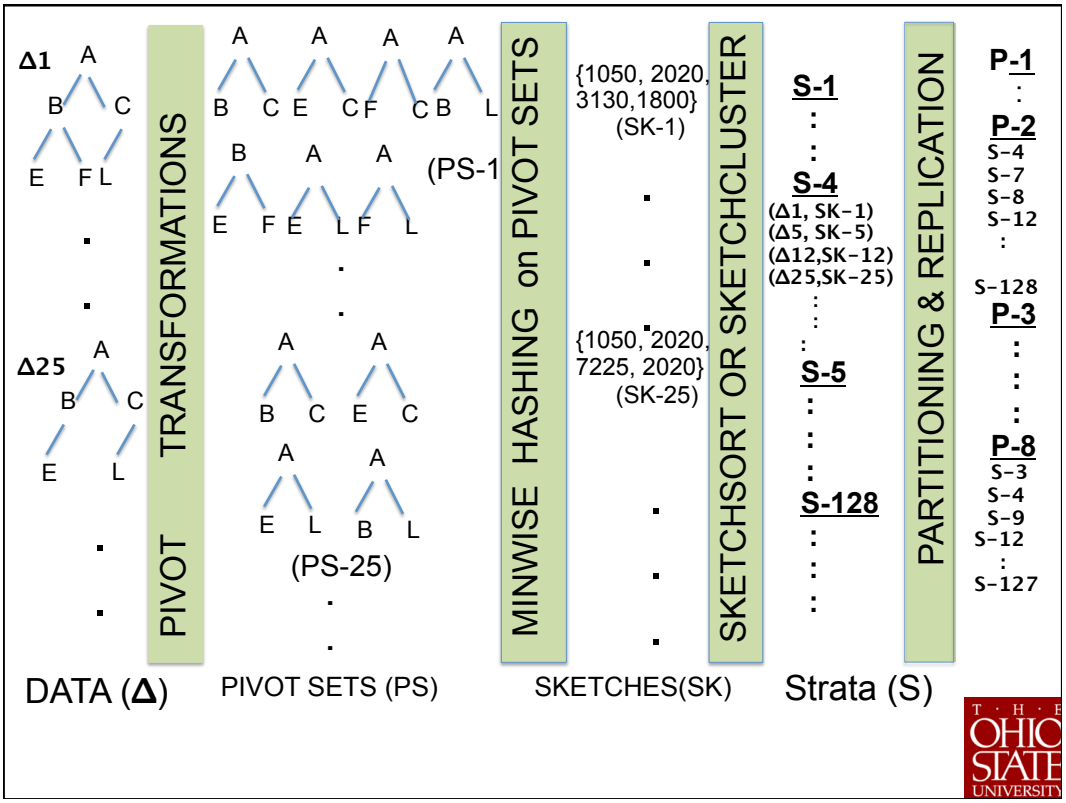
Solutions:

- Sort pivot sets directly (skip sketch step) – **Pivot Sort**
- Directly use output of LSH/Minwise Hash – **SketchSort**
- Cluster sketches with fast variant of k-modes – **SketchCluster**

Step 4: Partitioning

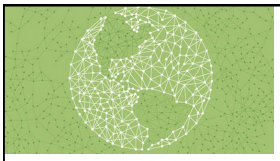
- **Problem:** How to partition stratified data?
- **Key Ideas:** Guided by application hints and system state.
- **Solutions:**
  1. **Proportional Allocation**: Split each stratum uniformly proportionally across all partitions → mitigates skew
  2. **Optimal Allocation** for first strata, proportional for rest [C77]
  3. **All-in-One** : Place each stratum in its entirety within a partition

IMPORTANT NOTE: We use sketches to create strata – but partitioning happens on original data.



Empirical Evaluation

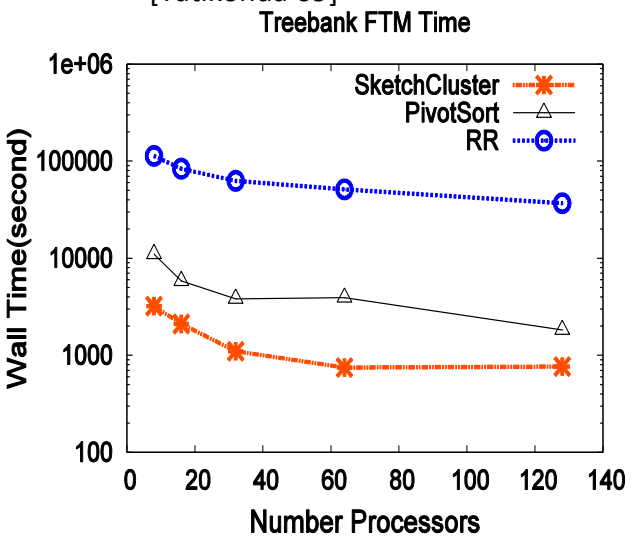
- We report wall clock times
- All times include cost of placement
- Evaluations on several key analytic tasks
  - Top-K algorithms [Fagin], Outlier Detection [Ghoting'08, Otey'06], Frequent Tree[Zaki'05, Tatikonda'09] and Graph Mining [Buehrer'06, Yan'02, Nijlsson'04], XML Indexing [Tatikonda'07], Community detection in Social/Biological data [Ucar'06, Satuluri'11], Web Graph Compression [Chellapilla'08-09; Vigna'11, LZ'77], Itemset Mining [Buehrer-Fuhry'15]
  - All applications are run straight out of the box – the only thing the user specifies relates to locality, skew, and interaction.



Frequent Pattern Mining

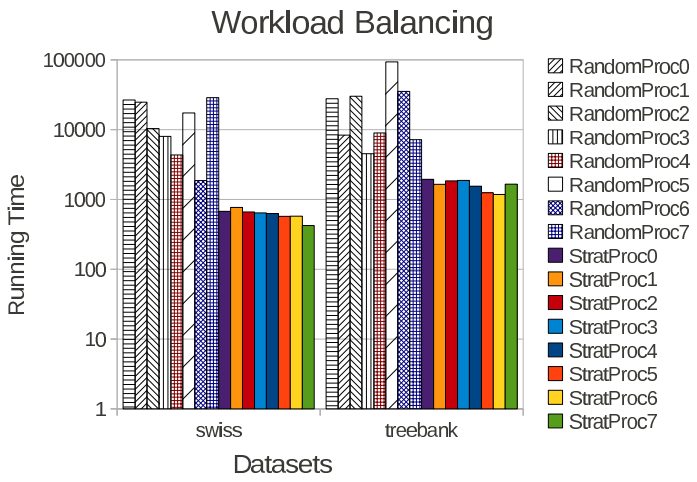
[Tatikonda'09]

- Used Widely
  - Transactions, graphs, trees
- Approach
  1. Distribute Data
     - Proportional Allocation
  2. Run Phase 1
  3. Exchange Meta Data
  4. Run Phase 2
  5. Final Reduction
- Placement mainly impacts steps 1-3. Steps 3 and 5 are sequential.



Proposed approaches shows 100X gains

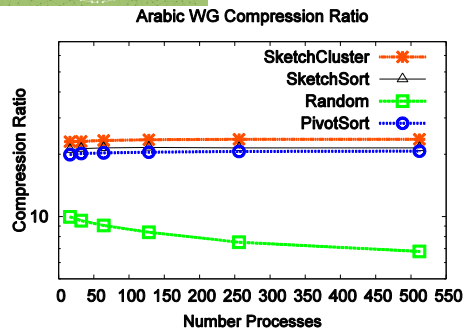
FTM Phase 1: Drilling Down

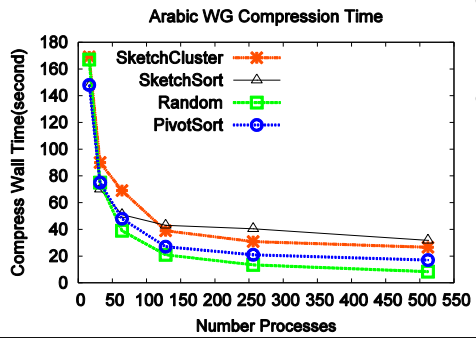


- Data Dependent Workload Skew is mitigated
- Payload-aware partitioning helps!

WebGraph Compression

[Vigna et al 2011]





Critical application for search companies

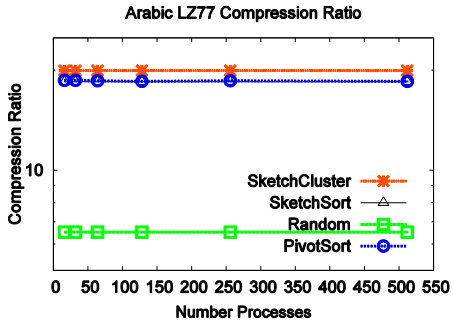
Key Requirement: Locality

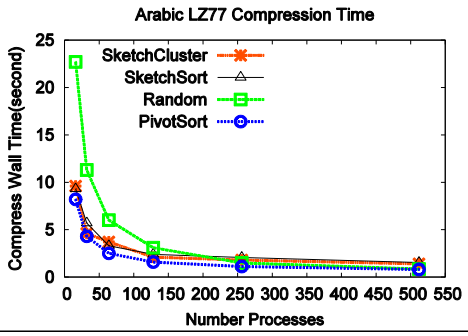
Approach:

- Distribute data via placement
- Run compression algorithm in parallel
- Parameters (similar to FTM)
  - Use adjacency/triangle pivots
  - Use All-in-one partitioning

Web data Compression (cont)

[Lempel Ziv '77]



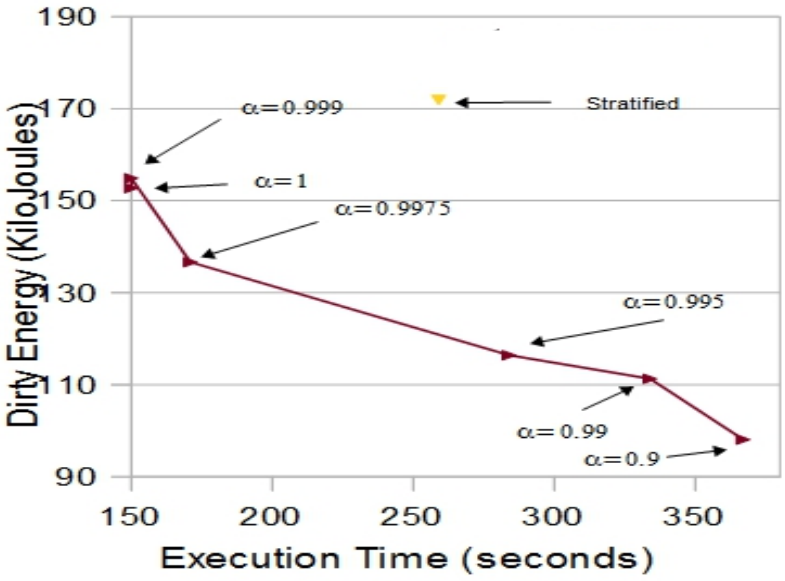


- Similar results on a different compression algorithm
- Strawman solutions offer low compression, and sometimes also more expensive.
- Our methods win on both quality and efficiency

Step 4: Energy- and Heterogeneity-Aware Partitioning

- Modern Datacenters are increasingly heterogeneous
  - Computation
  - Storage
  - Green Energy Harvesting
- Partitioning and placement while accounting for heterogeneity is challenging
  - Pareto Optimal Model

Pareto Frontier



Take Home Message

- In todays analytics world data has complex structure
- Stratitifed Data Placement has a central role to play

STRATIFIED DATA PLACEMENT		
HADOOP/SHARK/ Azure (HDFS/RDD/Blob)	Key Value Stores e.g. Memcached Redis	MPI & Partitioned Global Addresses Space Systems (PGAS) e.g. Global Arrays

  - Over 2 orders of magnitude improvement over state-of-art for a multitude of analytic tasks. First to explore this idea for placement.
  - Preliminary results on heterogeneous- energy-aware systems show significant promise!