Storage Management and Data Mining Problems in High Energy Physics Applications

Arie Shoshani
Doron Rotem
Henrik Nordberg
(http://www.lbl.gov/DM.html)

Scientific Data Management R&D Group
Lawrence Berkeley National Laboratory

June 26, 1997

Data Organization and Indexing of Large High Energy Physics Data

Collaboration	# members /institutions	Date of first data	# events/year	total data volume/year-TB
STAR	350/35	1999	10^7-10^8	300
PHENIX	350/35	1999	10^9	600
BABAR	300/30	1999	10^9	80
CLAS	200/40	1997	10^{10}	300
ATLAS	1200/140	2004	10^9	2000

STAR: Solenoidal Tracker At RHIC
RHIC: Relativistic Heavy Ion Collider

Data Organization for Efficient Retrieval of Very Large Datasets

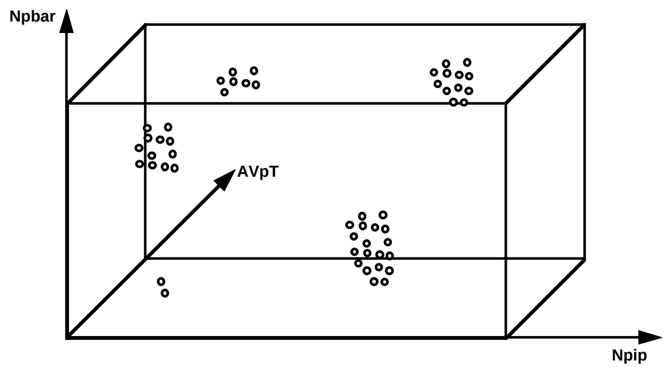
- **General problem area**
  - **how to cluster data in physical storage according to expected access patterns**
- **Observation**
  - **on parallel disks: distribute clusters to maximize parallel reads**
  - **on tape storage: keep cluster together to minimize tape mounts**

HENP Mass Storage Access

- **After event reconstruction, event properties (features) are extracted: called "level 1 n-tuples"**
- **Number of properties is large (50-100)**
- **e.g. momentum, no. of pions, transverse energy**
- **Multidimensional space is highly skewed and sparse**
- **Need to access events based on partial properties specification (usually ranges)**
- **Problem: re-organize event clusters on mass storage according to the property space**

Clusters in the M-Dim property space

Computer **S**cience **R**esearch and **D**evelopment **D**epartment
Computing **S**ciences **D**irectorate, **LBNL**



EXAMPLE OF EVENT PROPERTY VALUES

Computer **S**cience **R**esearch and **D**evelopment **D**epartment
Computing **S**ciences **D**irectorate, **LBNL**

I event 1
I N(1) 9965
I N(2) 1192
I N(3) 1704
I Npip(1) 2443
I Npip(2) 551
I Npip(3) 426
I Npim(1) 2480
I Npim(2) 541
I Npim(3) 382
I Nkp(1) 229
I Nkp(2) 30
I Nkp(3) 50
I Nkm(1) 209
I Nkm(2) 23
I Nkm(3) 32
I Np(1) 255
I Np(2) 34
I Np(3) 24
I Npbar(1) 94
I Npbar(2) 12
I Npbar(3) 24
I NSEC(1) 15607
I NSEC(2) 1342
I NSECpip(1) 638
I NSECpip(2) 191
I NSECpim(1) 728
I NSECpim(2) 206
I NSECkp(1) 3
I NSECkp(2) 0
I NSECkm(1) 0
I NSECkm(2) 0
I NSECp(1) 524
I NSECp(2) 244
I NSECpbar(1) 41
I NSECpbar(2) 8
R AVpT(1) 0.325951
R AVpT(2) 0.402098
R AVpTpip(1) 0.300771
R AVpTpip(2) 0.379093
R AVpTpim(1) 0.298997
R AVpTpim(2) 0.375859
R AVpTkp(1) 0.421875
R AVpTkp(2) 0.564385
R AVpTkm(1) 0.435554
R AVpTkm(2) 0.663398
R AVpTp(1) 0.651253
R AVpTp(2) 0.777526
R AVpTpbar(1) 0.399824
R AVpTpbar(2) 0.690237
I NHIGHpT(1) 205
I NHIGHpT(2) 7
I NHIGHpT(3) 1
I NHIGHpT(4) 0
I NHIGHpT(5) 0

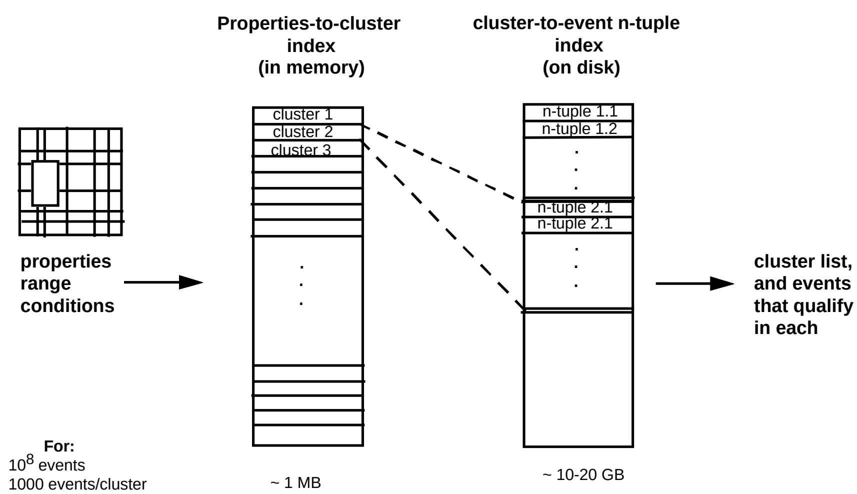
54 Properties, eventually 10^8 events

Size of HENP datasets on tape

- **STAR experiment**
  - **10^8 events over 3 years**
  - **1-10 MB per event: reconstructed data (DST tapes)**
  - **10^{15} total size**
  - **10,000 tapes per year (30 MB tapes)**

Manage Cluster Access

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL



Size of indexes for STAR

- **Index size**
  - **property space: 10^8 events x 200 bytes = 20 GB**
  - **index space: 10^5 clusters x 100 bytes = 10 MB (assume 1000 events/cluster)**
- **Problem**
  - **how to organize property space index**

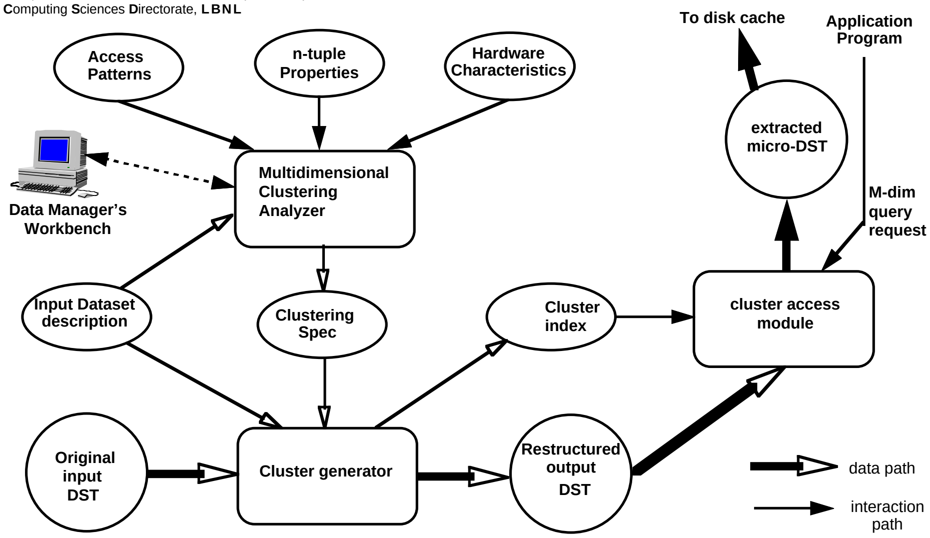
Main Tasks

Computer **S**cience **R**esearch and **D**evelopment **D**epartment
Computing **S**ciences **D**irectorate, **LBNL**

- **Discover event clusters**
  - **based on natural distribution - Data Mining**
  - **based on access patterns - consult physicists**
  - **simulate performance - data manager's workbench**
- **Manage cluster access**
  - **given a query, determine clusters to access, use multi-dimensional indexes to select events that qualify**
- **Reorganize DST tapes according to clusters**
  - **long process - done initially, then rarely**
  - **flow control - restart after interruption**
- **Cache management**
  - **determine if in cache, which incremental clusters to cache, which clusters to purge from cache**

Events Clustering and Access: Main Components

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL



Discover events clusters

Computer Science Research and Development Department
Computing Sciences Directorate, LBNL

- **Top down approach**
  - **partition each dimension into "bins" (e.g. 1-2 GEV, ..., 1-3 pions, ...)**
  - **select subset of dimensions based on physicist's experience**
  - **analyze which events fall into the same "cell" (i.e. m-dim rectangles formed by the bins)**
  - **eliminate empty cells**
  - **combine cells to form similar size clusters**
- **Assumption**
  - **most queries are "range queries"**

Discover events clusters

- **Bottom up approach**
  - **select a subset of the dimensions (using physicist's hints initially)**
  - **partition space on each dimension successively**
  - **determine suitability of various indexing methods to high dimensionality and skewed distributions: K-D trees, Quad-trees, R-trees,...**
  - **Iterate for other dimension combinations**
- **Consult physicists**
  - **do cluster correlations matter?**
  - **get additional hints on preferred dimensions**

Top Down Cell Management

- **Assume: 7 dimensions, 10 bins each**

  -- Number of cells: 10^7, 4 byte counters

  -- Number of bytes: 4×10^7, 40 MB

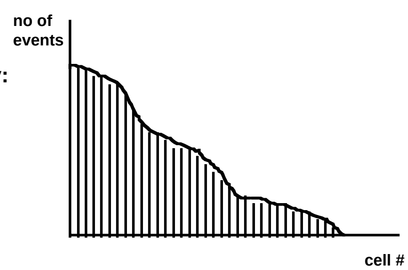
- **For e.g. small dataset 97% of cells are empty:**

  -- store only populated cells

  -- use hash tables to locate existing cells

  -- use 2 bytes for bin_id per property: ratio for p% full is: 200/ (n+2)p

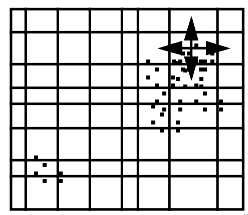
  -- No of bytes: for 7 dim, 3% => 5.4 MB



- **Sort cells by size (number of events)**

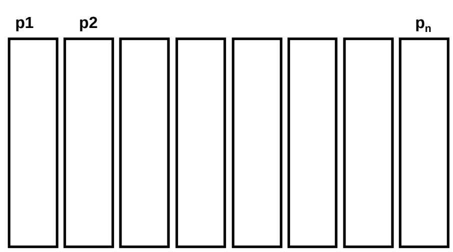
Cluster Identification

- **sort cells by size**
- **pick larger cell to start forming a cluster**
- **find all neighbors of "Manhattan distance" equal to 1**
- **include cells above a threshold**
- **iterate for all cells in cluster**
- **when no more cells above threshold, pick larger remaining cell and start forming a new cluster**
- **Display cluster distributions**



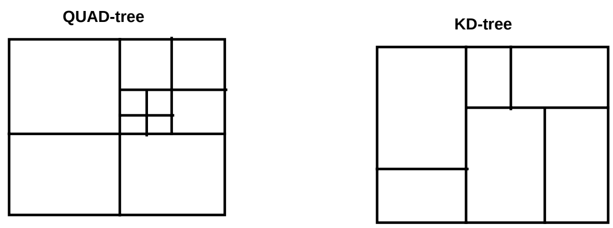
indexing over all properties

- **Assume 150 properties**
  - **– Any combination of range queries**
  - **– want to compute number of events**
- **possible solution: vertically partitioned file**
  - **– idea: touch only properties in queries**
  - **– each partition 10^8 x 4 bytes = 400 MB per partition**
  - **– too expensive in space and time**



indexing over all properties

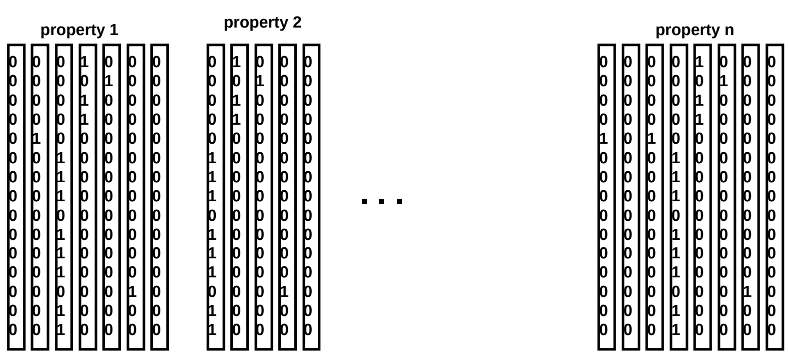
- **other possible solutions**
  - **partitioning MD space (KD-trees, n-QUAD-trees, ...)**
  - **for high dimensionality - either fanout or tree depth too large**
  - **e.g. symmetric n-QUAD-trees require 2^{150} fanout**
  - **non-symmetric solutions are order dependent**



Bit-Sliced indexing

Computer **S**cience **R**esearch and **D**evelopment **D**epartment
Computing **S**ciences **D**irectorate, **LBNL**

- **partition each property into bins**
- **for each bin generate a bit vector**
- **compress each bit vector (run length encoding)**



Compression method

Uncompressed:
0000000000001111000000000 ......0000001000000001111111100000000 000000

Compressed:
12, 16, 1016,1017,1025,2025

Advantage:

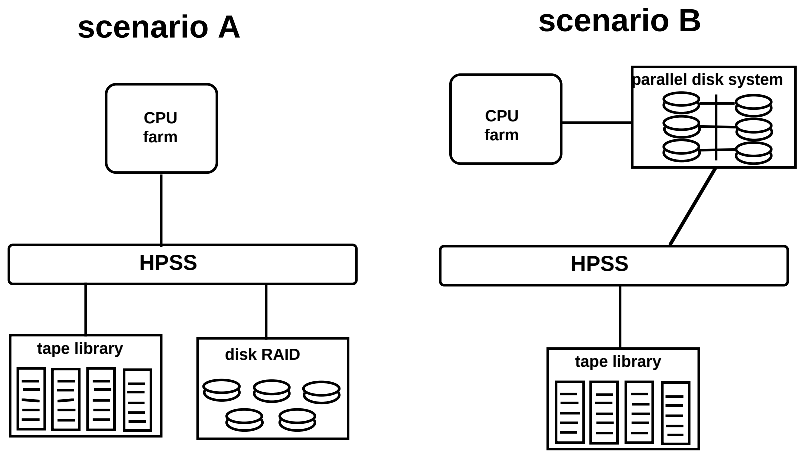
Can perform: AND, OR, COUNT operations on compressed data

Bit-Sliced indexing

- **Estimated size**
  - **100 properties X 10 bins X 10^8 bits = 10^{11} bits**
  - **compression factor (avg run length) = 1000**
  - **total size = 10^8 bits ~ 10 MB**
- **Advantage**
  - **only bit partitions need to be accessed (multiple bins per property are “or”ed, result for each property requested are “and”ed)**
  - **operations can be performed on compressed bit-slices**
  - **compressed bit-slices can be processed in paprallel**
- **Disadvantage**
  - **results can be given on bin boundaries only**

Cache Management

Hardware scenarios

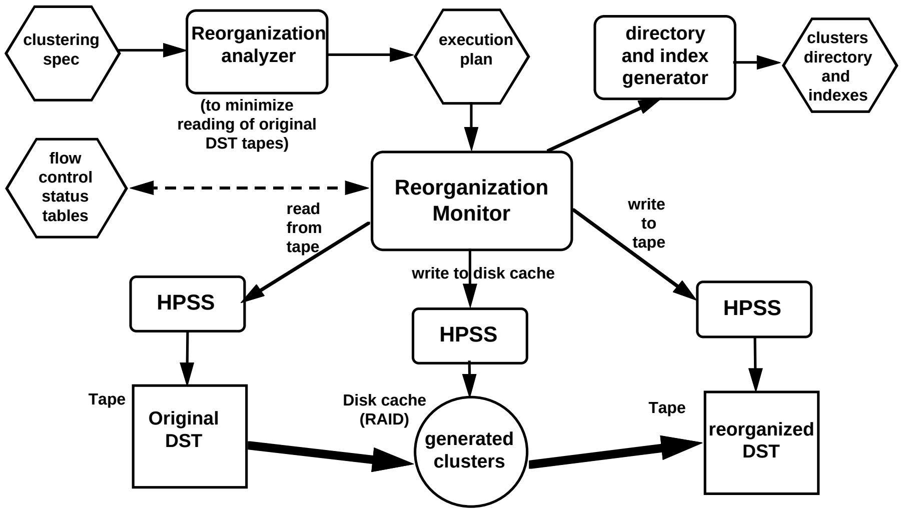


Cache Management Issues

Computer **S**cience **R**esearch and **D**evelopment **D**epartment
Computing **S**ciences **D**irectorate, **LBNL**

- **Scenario A**
  - RAID more expensive than a Parallel Disk System (factor 2-3)
  - but, rely on HPSS to manage disk
  - storage management simplified
- **Scenario B**
  - a Parallel Disk System is cheaper, does not depend on RAID vendor
  - but, need to manage disk allocation
  - has control over placement of events on cache
- **Planned initial pilot**
  - Scenario A under NERSC

Reorganize DST tapes according to clusters



Open Problems

- Discover event clusters in sparse high M-dim space given analyst guidance on bining
- Analyze the natural clustering of events by properties
- develop an efficicient index on high M-dim space
- develop a cache mangement policy for job mixes
- Simulate and test the effect of distributing events on disk cache by blocks vs. one event per disk
- The benefit of partial event data replication to accommodate conflicting access patterns

Open Problems (cont'd)

- **The ability of HPSS to store files on tapes according to external specifications**
- **The ability of HPSS to perform "partial file reads" from tape storage**
- **The possibility and effectiveness of parallel tape management under HPSS**
- **The benefit of partial event data replication to accommodate conflicting access patterns**