GPU-Accelerated Incremental Correlation Clustering of Large Data with Visual Feedback

Eric Papenhausen and Bing Wang
(Stony Brook University)

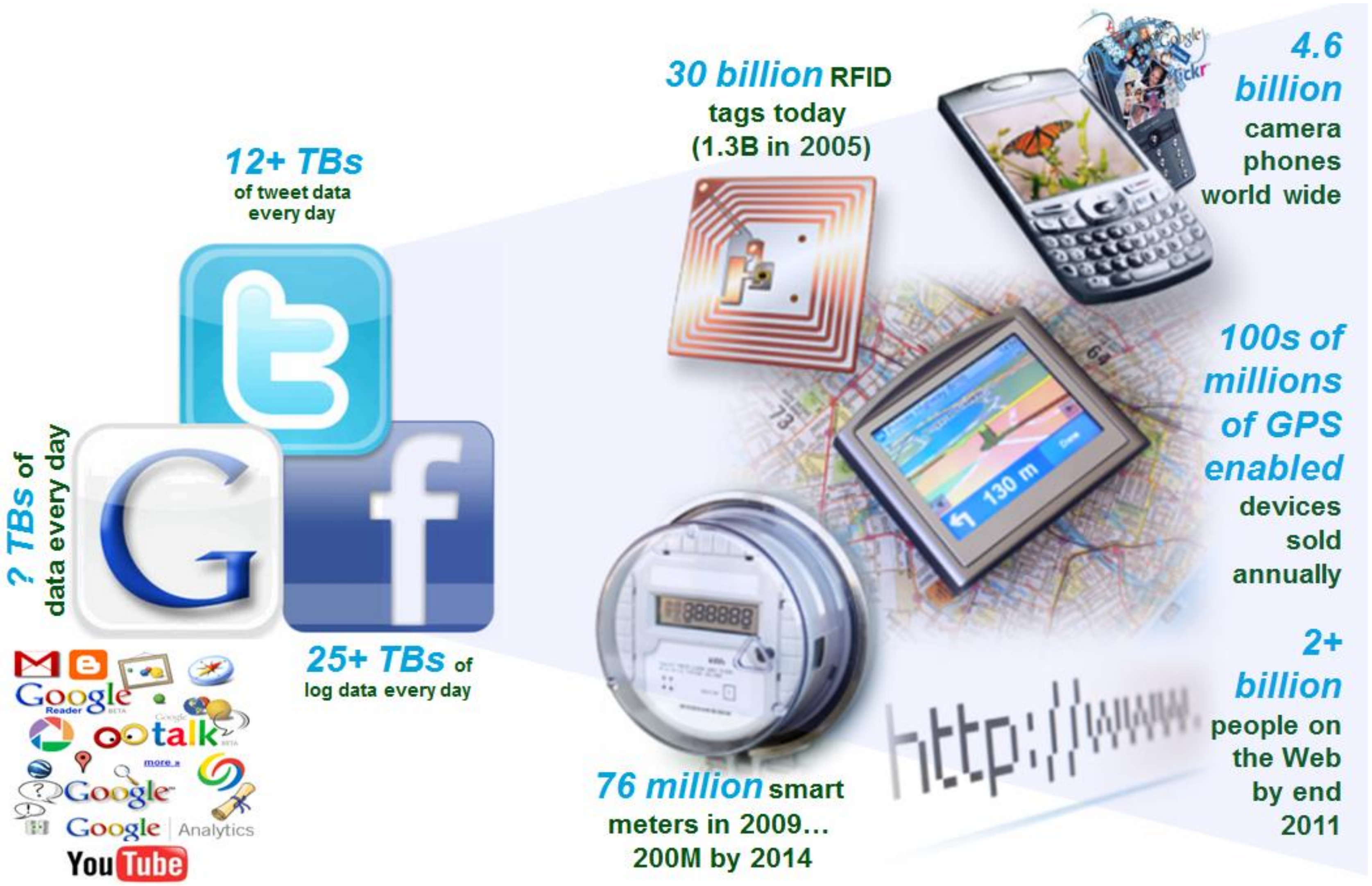
Sungsoo Ha (SUNY Korea)

Alla Zelenyuk (Pacific Northwest National Lab)

Dan Imre (Imre Comsulting)

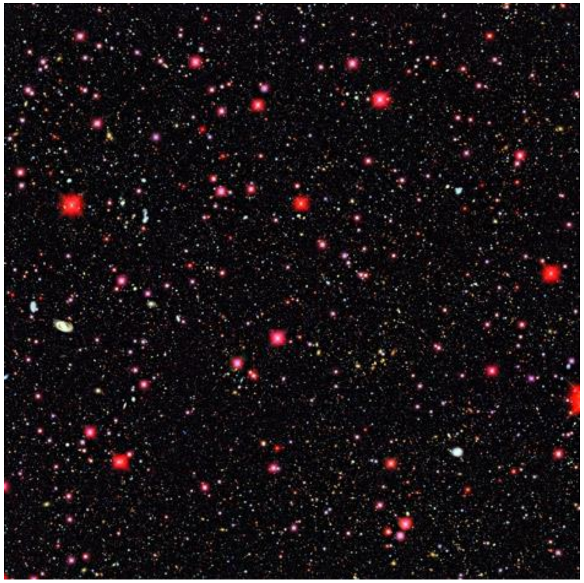
Klaus Mueller
(Stony Brook University and SUNY Korea)

The Internet of Things and People



The Large Synoptic Survey Telescope

Will survey the entire visible sky deeply in multiple colors every week with its three-billion pixel digital camera



Probe the mysteries of Dark Matter & Dark Energy

10 x more galaxies than Sloan Digital Sky Survey

Movie-like window on objects that change or move rapidly

Our Data – Aerosol Science

Acquired by a state-of-the-art single particle mass spectrometer (SPLAT II) often deployed in an aircraft

Used in atmospheric chemistry

- understand the processes that control the atmos. aerosol life cycle
- find the origins of climate change
- uncover and model the relationship between atmospheric aerosols and climate

Our Data – Aerosol Science

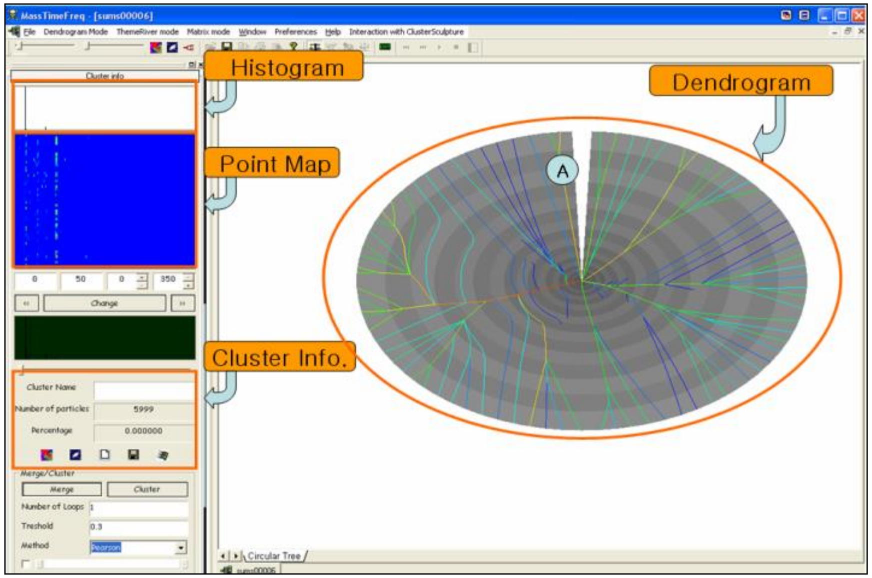
SPLAT II can acquire up to 100 particles per second at sizes between 50-3,000 nm at a precision of 1 nm

- Creates a 450-D mass spectrum for each particle

SpectraMiner:

- Builds a hierarchy of particles based on their spectral composition
- Hierarchy is used n subsequent automated classification of new particle acquisitions in the field or in the lab

SpectraMiner



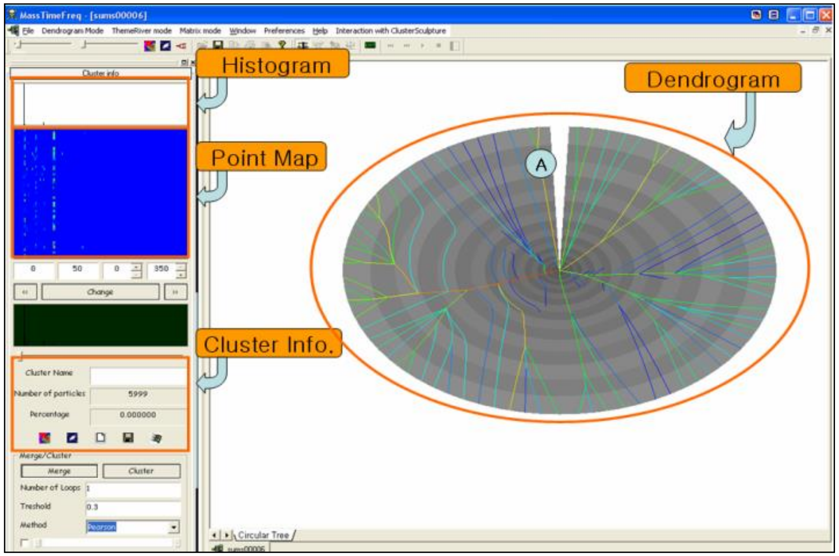
SpectraMiner

Tightly integrate the scientist into the data analytics

- Interactive clustering – cluster sculpting
- Interaction needed since the data are extremely noisy
- Fully automated clustering tools typically do not return satisfactory results

Strategy:

- Determine leaf nodes
- Merge using correlation metric via heap sort
- Correlation sensitive to article composition ratios (or mixing state)



SpectraMiner – Scale Up

CPU-based solution worked well for some time

SPLAT II and new large campaigns present problems

- At 100 particles/s, the number of particles gathered in a single acquisition run can easily reach 100,000
- This would take just a 15 minute time window

Large campaigns are much longer & more frequent

- Datasets of 5-10M particles have become the norm

Recently SPLAT II operated 24/7 for one month

- Had to reduce acquisition rate to 20 particles/s

CPU-based solution took days/weeks to compute

Interlude: Big Data – What Do You Need?

#1: Well, data !!

- data = $$
- look at LinkedIn, Facebook, Google, Amazon

#2: High performance computing

- parallel computing (GPUs), cloud computing

#3: Nifty computer algorithms for

- noise removal
- redundancy elimination and importance sampling
- missing data estimation
- outlier detection
- natural language processing and analysis
- image and video analysis
- learning a classification model

Interlude: Big Data – What Do You Need?

#1: Well, data !!

- data = $$
- look at LinkedIn, Facebook, Google, Amazon

#2: High performance computing

- parallel computing (GPUs), cloud computing

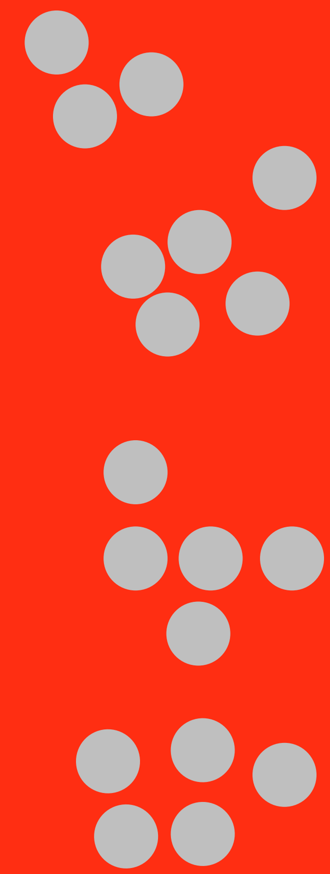
#3: Nifty computer algorithms for

- noise removal
- **redundancy elimination and importance sampling**
- missing data estimation
- outlier detection
- natural language processing and analysis
- image and video analysis
- learning a classification model

Incremental k-Means – Sequential

Basis of our trusted CPU-based solution (10 years old)

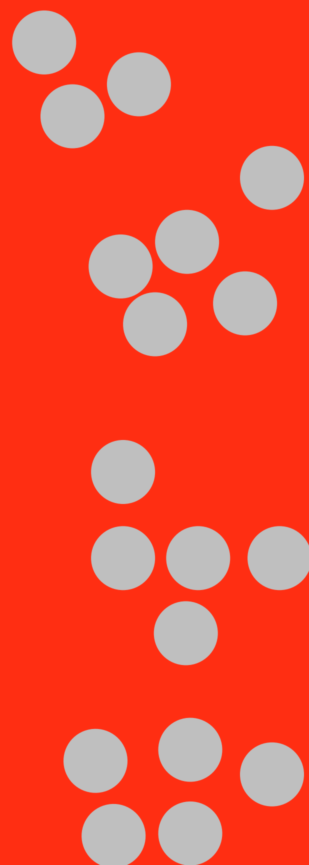
```
Make the first point a cluster center
While number of unclustered points > 0
   Pt = next unclustered point
   Compare Pt to all cluster centers
   Find the cluster with the shortest distance
   If(distance < threshold)
      Cluster Pt into cluster center
   Else
      Make Pt a new cluster center
   End If
End
Second pass to cluster outliers
```



Incremental k-Means – Parallel

New parallizable version of the previous algorithm

Do
  Perform sequential k-means until C clusters emerge
  Num_Iterations = 0
  ***While** Num_Iterations < Max_iterations*
    In Parallel: Compare all points to C centers
    In Parallel: Update the C cluster centers
    Num_Iterations++
  End
  Output the C clusters
  ***If** number of unclustered points == 0*
    End
  ***Else** continue*
End



Comments and Observations

Algorithm merges the incremental k-means algorithm with a parallel implementation (k=C)

Design choices:

- C=96 good balance between CPU and GPU utilization
- With C>96 algorithm becomes CPU-bound
- With C<96 the GPU would be underutilized
- A multiple of 32 avoids divergent warps on the GPU
- Max_iterations = 5 worked best

Advantages of the new scheme:

- Second pass of previous scheme no longer needed

GPU Implementation

Platform

- 1-4 Tesla K20 GPUs
- Installed in a remote ‘cloud’ server
- Future implementations will emphasize this cloud aspect more

Parallelism

- Launch N/32 thread blocks of size 32 x 32 each
- Each thread compares a point with 3 cluster centers
- Make use of shared memory to avoid non-coalesced memory accesses

GPU Implementation – Algorithm

```
c1 = Centers[tid.y]          // First 32/96 loaded by thread block
c2 = Centers[tid.y + 32]  // Second 32/96 loaded
c3 = Centers[tid.y + 64]  // Final 32/96 loaded
pt = Points[tid.x]

[clust, dist] = PearsonDist(pt, c1,c2,c3)   // d_xy=1-r_xy
[clust, dist] = IntraColumnReduction(clust,dist)

//first thread in each column writes result
If(tid.y == 0)
    Points.clust[tid.x] = clust
    Points.dist[tid.x] = dist
End If
```

Quality Measures

Measure cluster quality with the Davies-Bouldin (DB) index

$$DB = \frac{1}{n}\sum_{i=1}^{n} \max_j\left(\frac{\sigma_i + \sigma_j}{M_{ij}}\right)$$

σ_i and σ_j are intra-cluster distances of clusters i, j

M_{ij} is the inter-cluster distance of clusters i, j

DB should be as small as possible

Acceleration by Sub-Thresholding

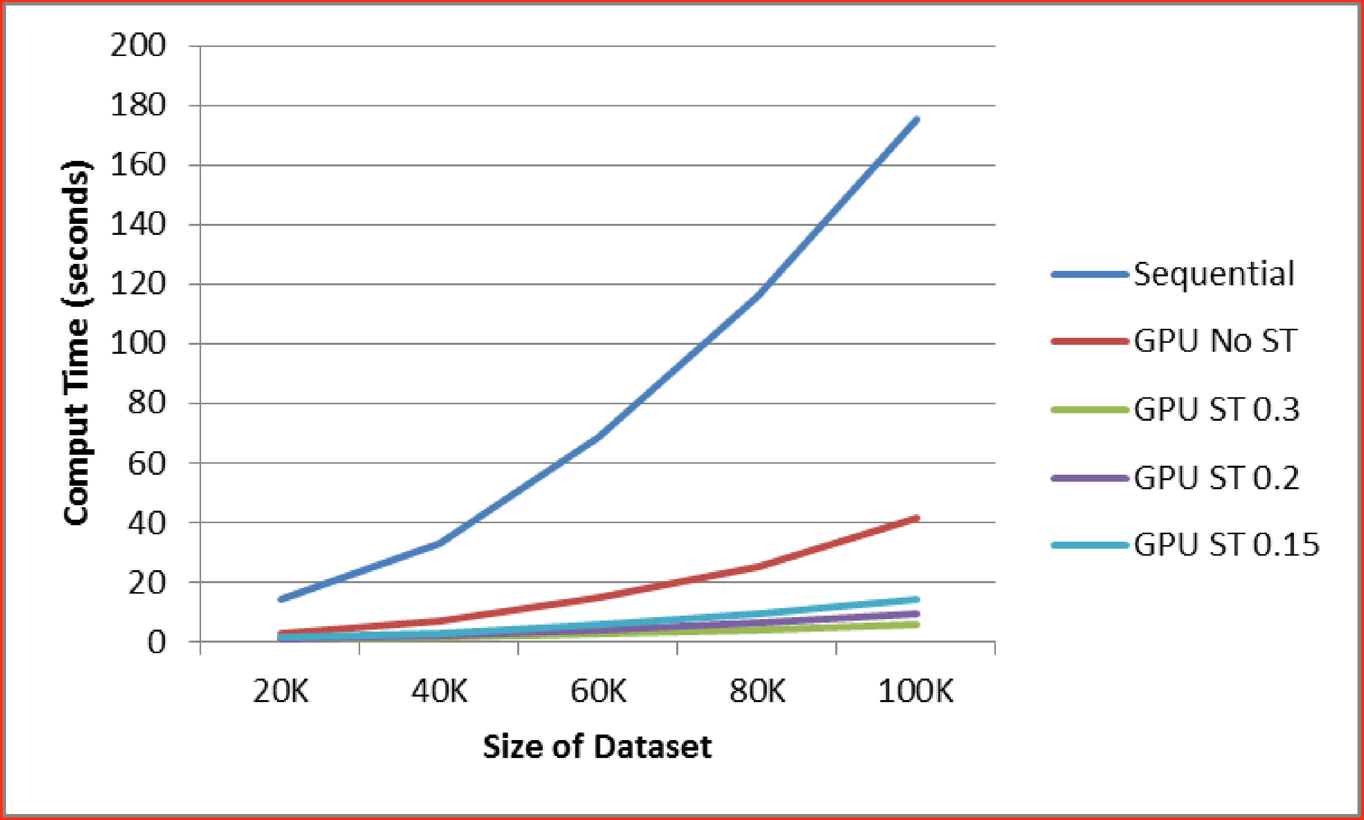
Size of the data was a large bottleneck

- Data points had to be kept around for a long time
- Cull points that were tightly clustered early
- These are the points that have a low Pearson's distance

This also improved the DB index

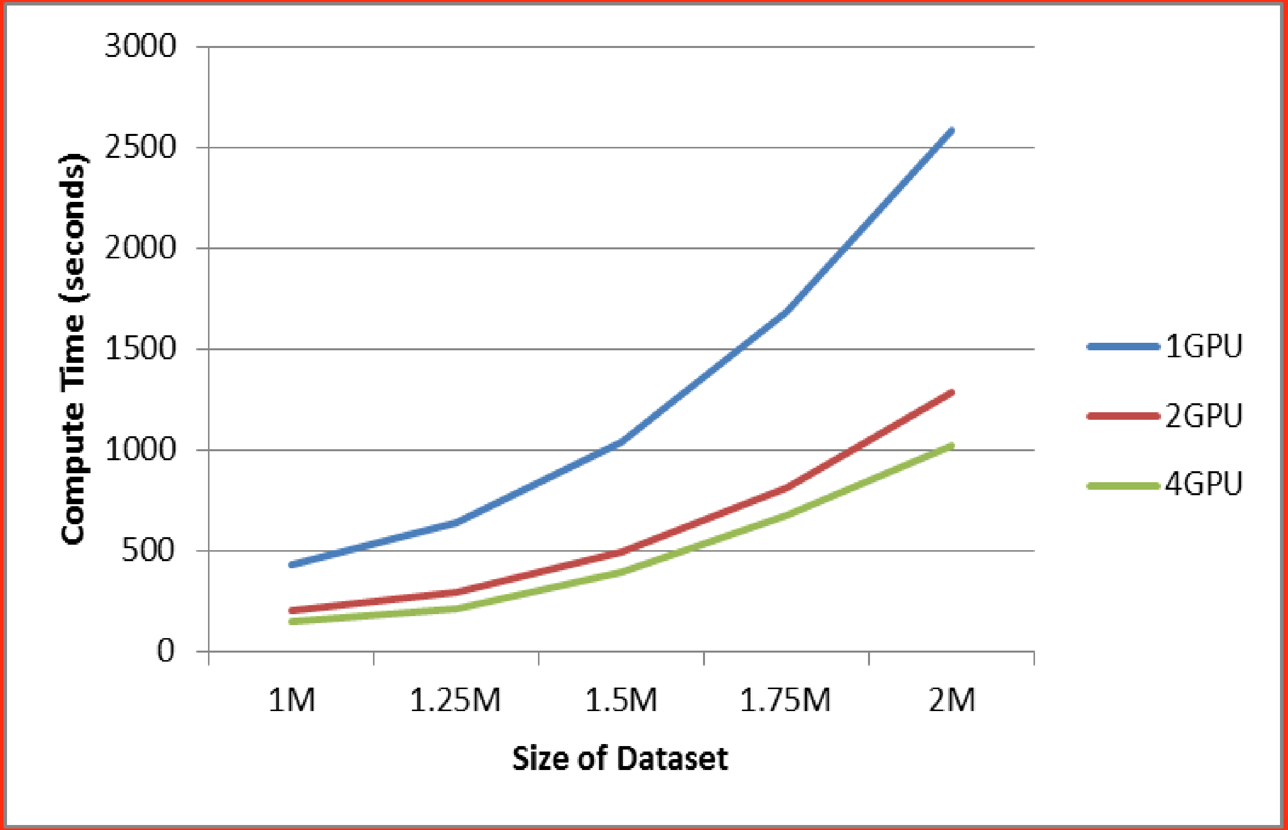
Size	Seq.	Par.	ST: 0.3	ST: 0.2	ST: 0.15
10k	0.527	0.539	0.540	0.537	0.529
50k	0.546	0.590	0.548	0.554	0.539
100k	0.550	0.584	0.600	0.570	0.544
200k	0.564	0.587	0.640	0.593	0.564

Results – Sub-Thresholding



About 33x speedup

Results – Multi-GPU



4-GPU has about 100x speedup over sequential

In-Situ Visual Feedback (1)

Visualize cluster centers as summary snapshots

- Glimmer MDS algorithm was used
- Intuitive 2D layout for non-visualization experts

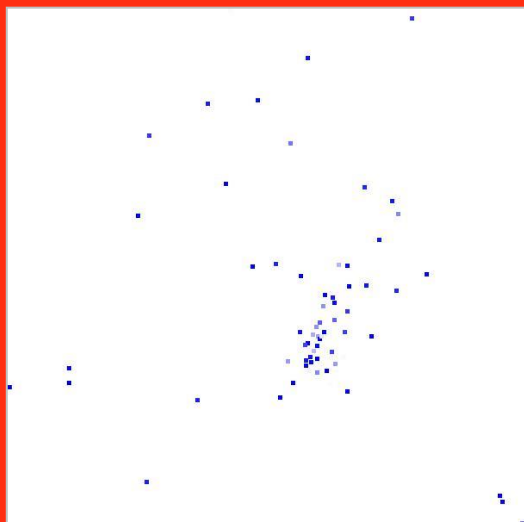
Color map:

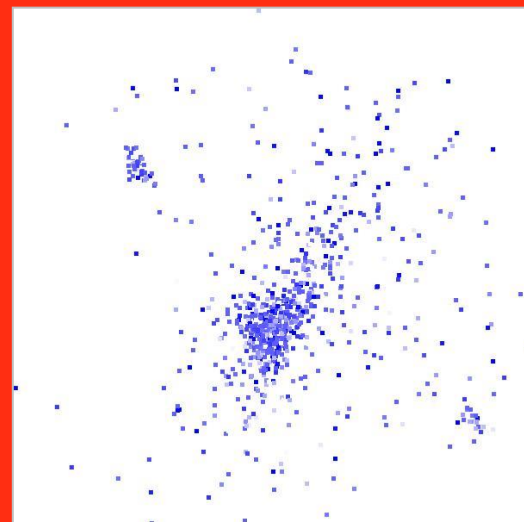
- Small clusters map to mostly white
- Large clusters map to saturated blue

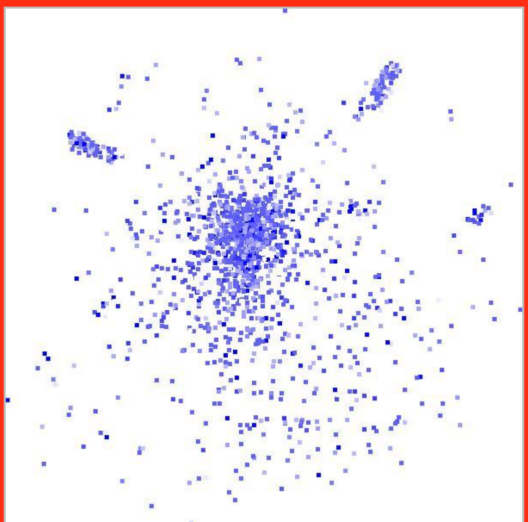
We find that early visualizations are already quite revealing

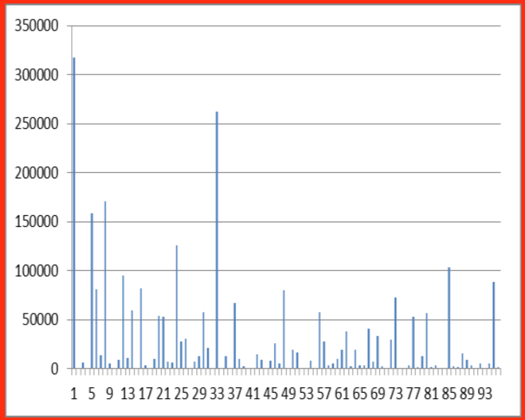
- This is shown by cluster size histogram
- Cluster size of $M>10$ is considered significant

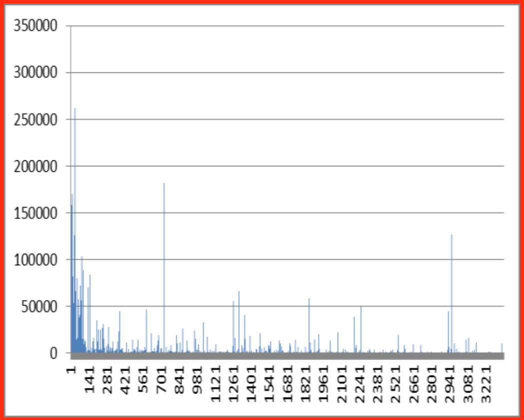
In-Situ Visual Feedback (2)

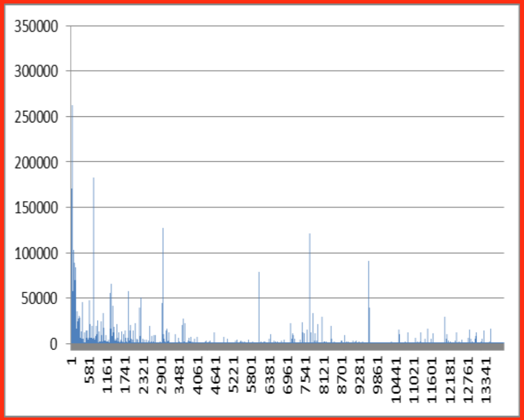










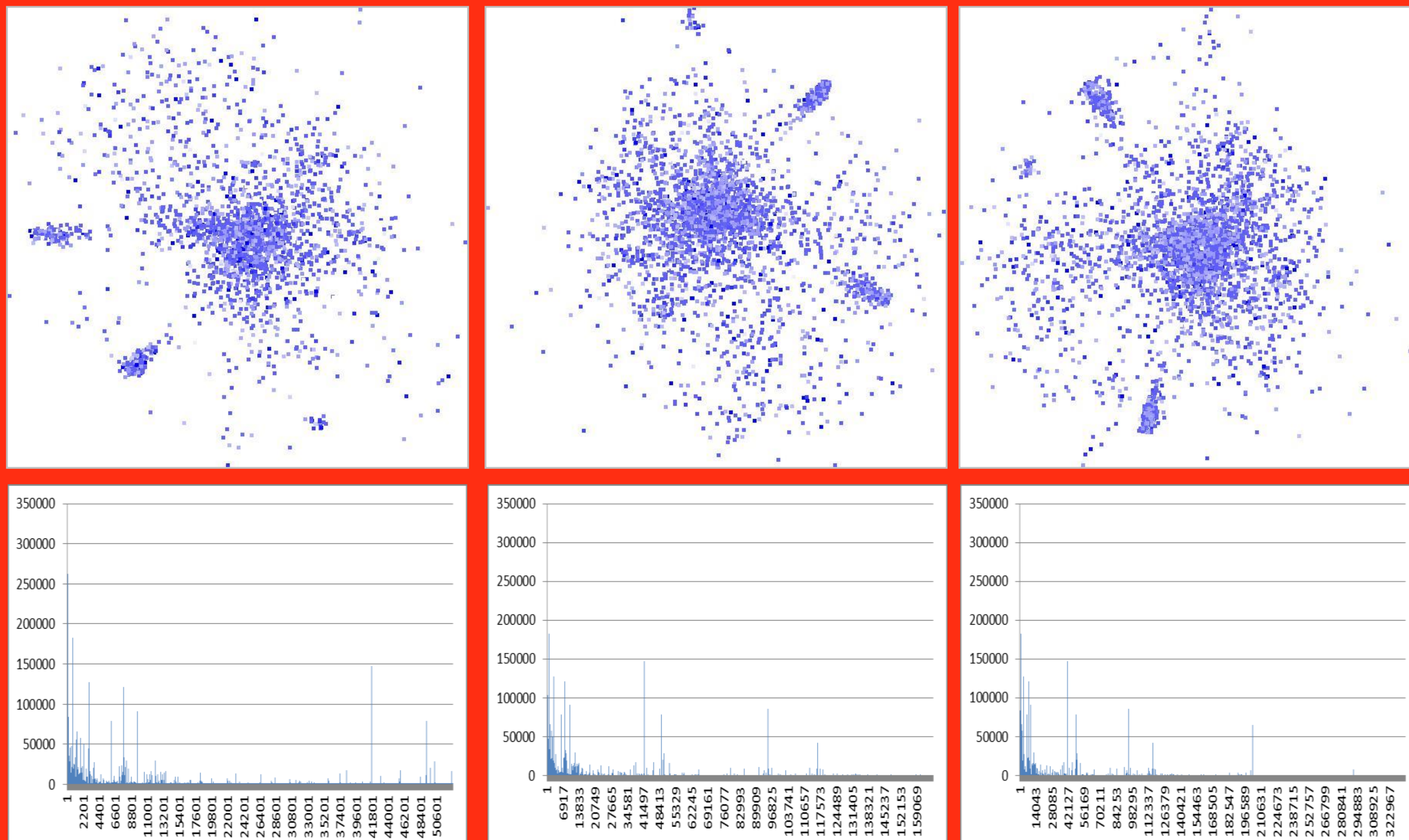


79/96

998/3360

2004/13920

In-Situ Visual Feedback (3)



3001/52800 4002/165984 4207/336994

Relation To Previous Work (1)

Main difference

- We perform k-means clustering for data reduction

Previous work often uses map-reduce approaches

- Connection most often with MPI/OpenMP
- Distribute points onto a set of machines
- Compute (map) one iteration of local k-means in parallel
- Send the local k means to a set of reducers
- Compute their averages in parallel and send back to mappers
- Optionally skip the reduction step and instead broadcast to mappers for local averaging

Relation To Previous Work (2)

GPU solutions

- Often only parallelize the point-cluster assignments
- Compute new cluster centers on the CPU due to low parallelism

Conclusions and Future Work

Current approach quite promising

- Good speedup
- In-situ visualization of data reduction process with early valuable feedback

Future work

- Load-balancing point removal for multi-GPU
- Anchored visualization so layout is preserved
- Enable visual steering of point reduction
- Extension to streaming data
- Also accelerate hierarchy building

Final Slide

Thanks to NSF and DOE for funding

Addional support from the Ministry of Korea Knowledge Economy (MKE)

Any questions?