CloudVista: Visual Cluster Exploration for Extreme Scale Data in the Cloud

Keke Chen
Data Intensive Analysis and Computing (DIAC) Lab
Ohio Center of Excellence in Knowledge Enabled Computing
(Kno.e.sis Center)
Dept. Computer Science & Engineering
Wright State University

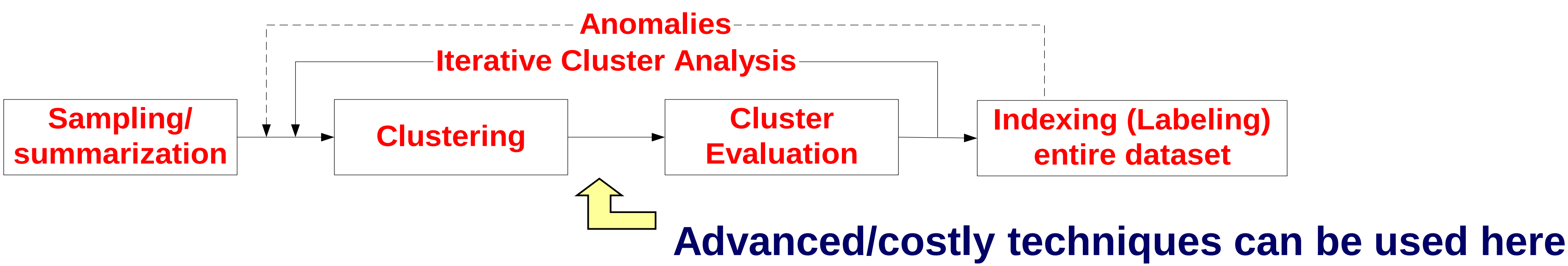
Motivation

- Applications are outsourced, moved to the cloud
- Very large data are generated and stored in the cloud
  - **Remote extreme-scale data** is now the problem not only with scientific computing, but also with many applications

→ *Need effective cloud-based data analytics tools*

Motivation

- Traditional tools are not sufficient for handling datasets in such a scale
  – We are talking about scales in millions-billions of records
  – The well known three-phase framework does not work effectively for clustering analysis

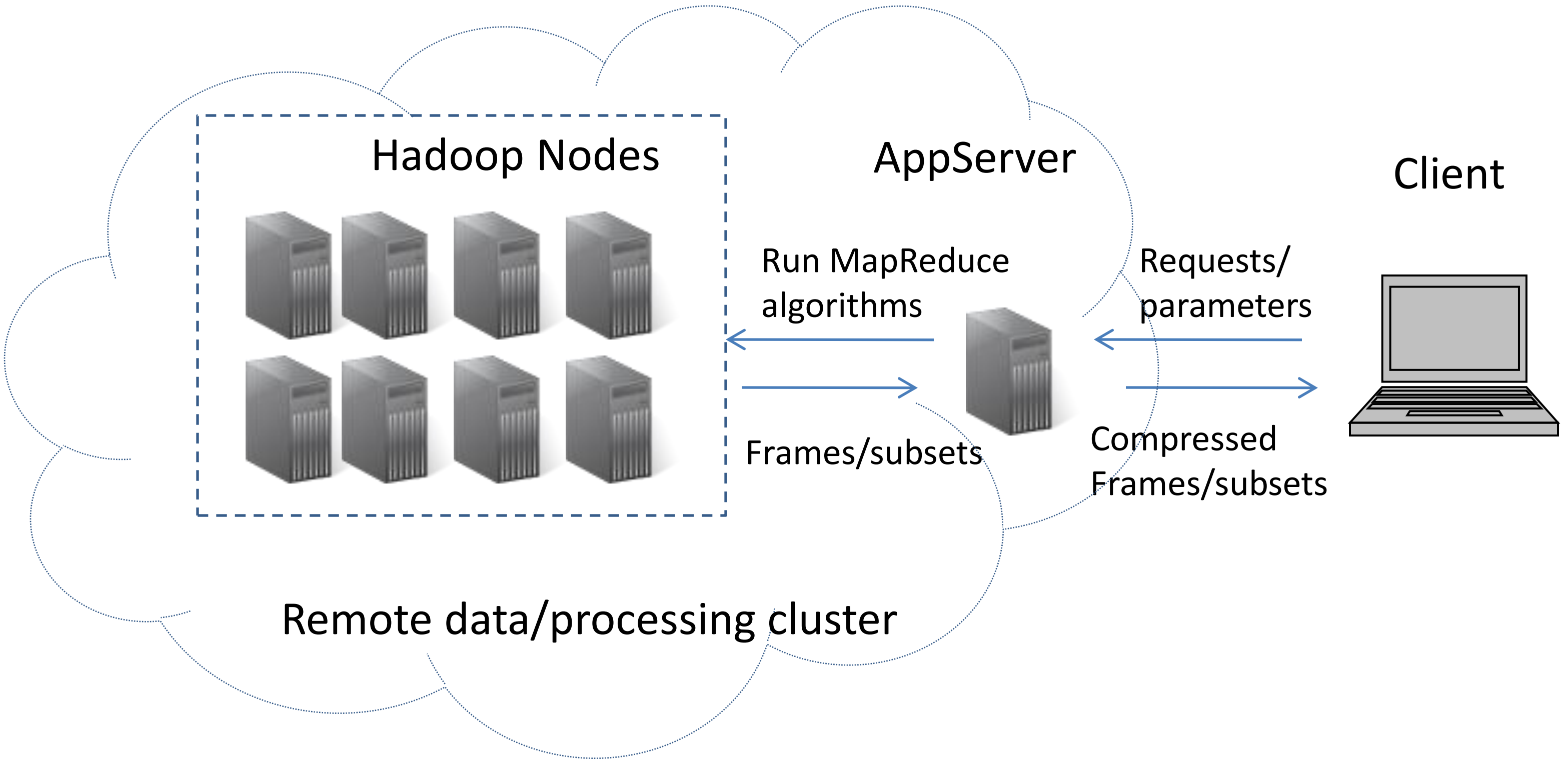


  – Mismatch between intermediate data and entire data (caused by the low sample rate)
  – Need expensive iterative exploration to validate the results

Proposed approach

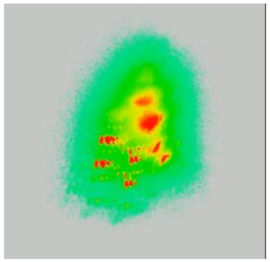
- Interactive visual cluster exploration on ***remote*** extreme scale data
  - Good, but is it possible?
- Challenges
  - Interaction does not work in the same way as we do with desktops – latency!
  - Batch processing – how to predict and generate useful visualization?
  - Visualization model needs to support parallel processing
- Contributions
  - Implement an effective batch-interactive processing strategy
  - Use VISTA visualization model + MapReduce for massive parallel processing
  - Use RandGen algorithm for meaningful batch processing (generating **useful** visual frames in batch)
  - Analyze the number of cloud-side operations → important for pay-as-you-use

System architecture



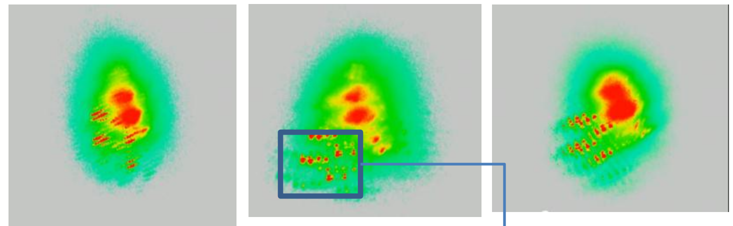
Major components:
- "Frames"
- MapReduce algorithms
- interaction model

workflow



Starting frame

RandGen in the cloud



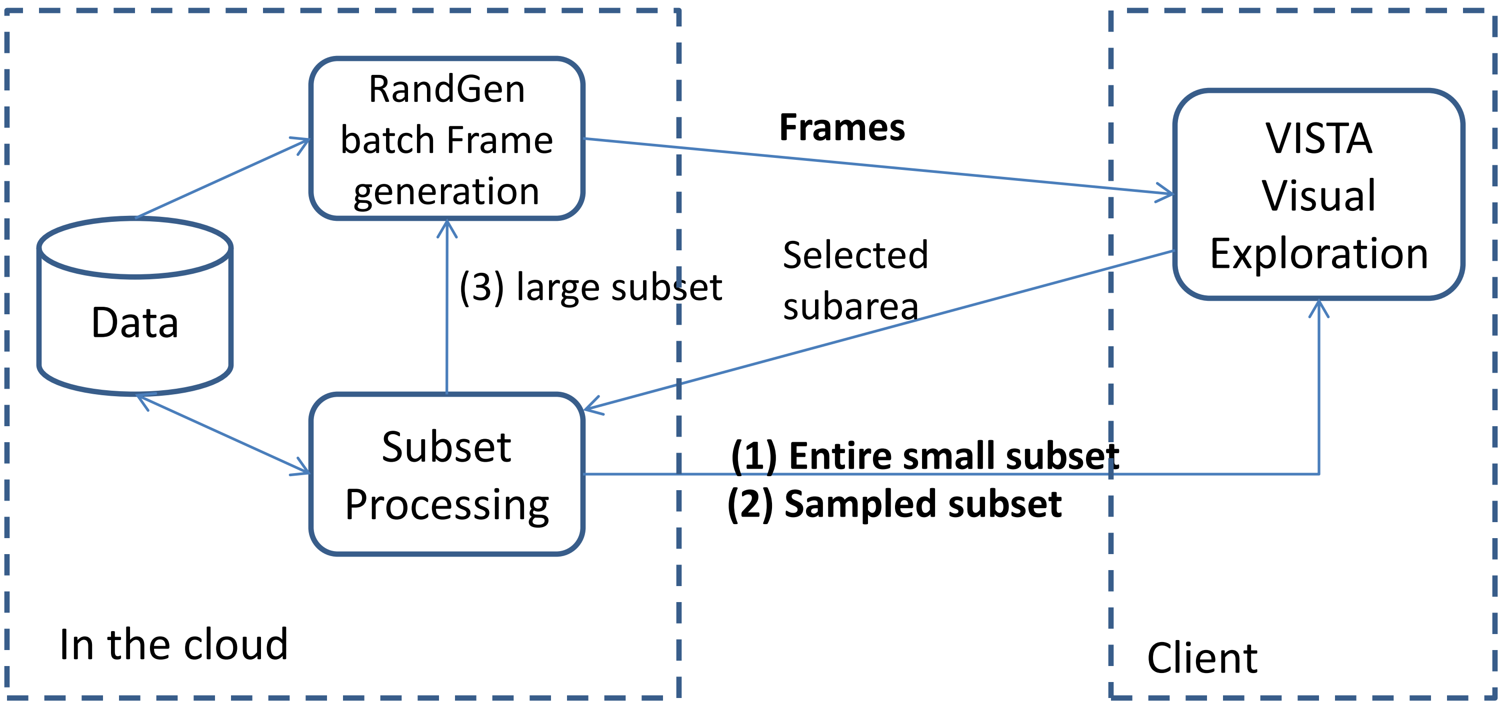
A batch of related frames for local exploration

or generate another batch Starting from *frame i*

Subset Processing

(1) Small enough to fit local machine
(2) Medium size that sampling works
(3) Too large, use RandGen again

Interaction model



The user selects a visual area → converted to a selected subset

(1) # of records <= client manageable size → return the whole subset

(2) Acceptable sample rate * # records <=manageable size

(3) Too large → RandGen

Key questions

- How to generate useful visual frames in batch
  - Usefulness: do they help exploring clustering structures?
- How often is the cloud-side processing needed?
  - $$$

VISTA visualization model

Zooming factor

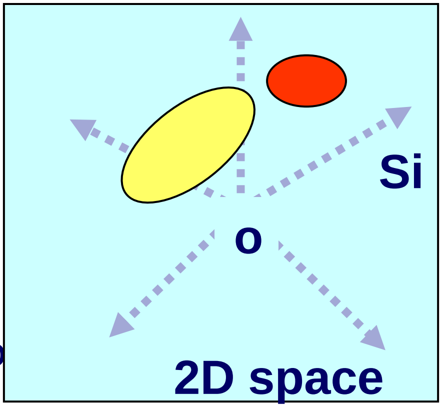
$$A(x_1, \ldots, x_k, \alpha_1, \ldots, \alpha_k) = (c/k) \sum_{i=1}^{k} \alpha_i x_i \vec{s}_i - \vec{o}$$

Normalized kDimensional point
$x_i \in [-1, 1]$

Dimensional adjustable alpha parameters
$\alpha_i \in [-1, 1]$

2D unit vector for dimension i.

Visualization center



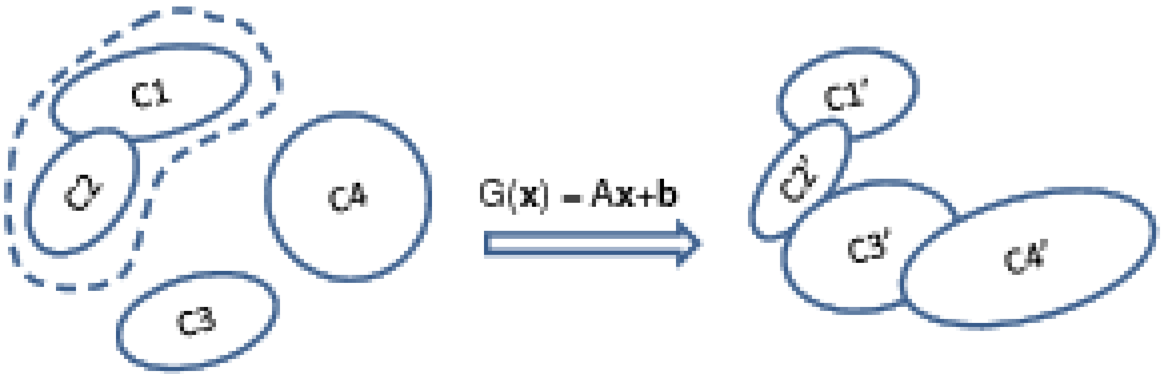
Important property: (1) simple adjustable linear mapping →separated clusters are indeed separated in the original space; visually overlapped clusters may not overlap in the original space ;**(2)** parallelizable

Challenge: to find all visually separated clusters

Method: adjust alpha parameters to generate cluster "animation"; may need several visual frames to describe complicated distributions

the theory behind

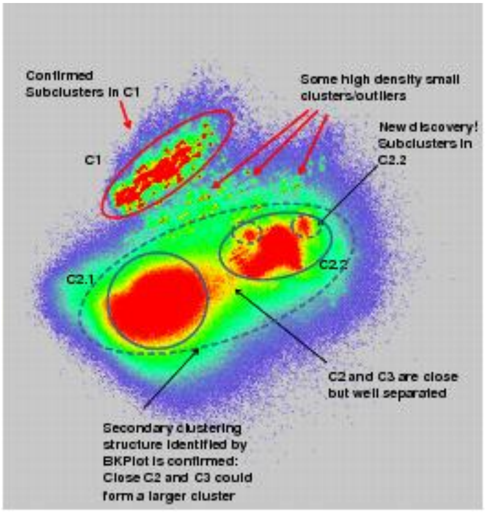
- Consider X as a mixture of Gaussian clusters
  - affine transformation moves centers and changes scales of Gaussian clusters



Visual Frame

- A set of VISTA parameters determines a visual frame
- A visual frame is a 2D density map

  -Each cell records the number of points mapped to the cell.
  -Sparse representation: (x, y, density)
  -**The size is limited, much smaller than the dataset**
  - the size depends on the **resolution**



- Density maps need to be generated remotely and in parallel – it is a data/compute intensive process
  – Vista mapping is naturally parallel (point wise)
  – Easy to implement with MapReduce

RandGen algorithm

- Infeasible to generate one frame *remotely* upon each user's parameter tuning
- Goal: generate a batch of **related** visual frames **useful** for finding clustering structure

B: uniform random variable generating -1/1
t: step length – the amount of alpha change
phi: # of frame

$$\delta_i = t \times B,$$

$$\alpha_i^{\phi+1} = \begin{cases} 1 & \text{if } \alpha_i^{\phi} + \delta_i > 1 \\ \alpha_i^{\phi} + \delta_i & \text{if } \alpha_i^{\phi} + \delta_i \in [-1, 1] \\ -1 & \text{if } \alpha_i^{\phi} + \delta_i < -1, \end{cases}$$

RandGen algorithm

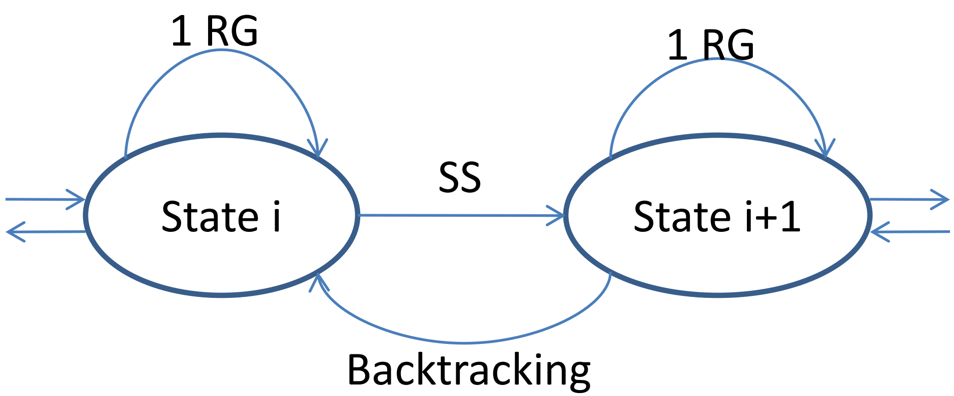
- We prove that

  "Distant points are more likely mapped to distant visual points in the random process"

  – Separated clusters in the original space will be seen separated in visualization with high probability, given a sufficient number of frames generated by RandGen

How many cloud operations

- Formalize the interaction model



RG: RandGen
SS: subset selection

Total # of cloud operations: **~ length of the chain * # of interesting areas in the top level**

Length of the chain : $\lfloor \log_\lambda \frac{\mu}{r N_0} \rfloor + 1$ **➔ this number is often small!**

N0: size of the entire data set
r: acceptable sample rate

mu: client capacity
lambda: selected area/total area

Experimental Results

- Hadoop cluster (1master + 15 worker nodes)
- Data
  - 25 Millions of census data records (68 dimensions, 5.3 GB)
  - 40 Millions of intrusion detection records (41 dimensions, 13.5GB)

Sample visualization comparison

10K sample records

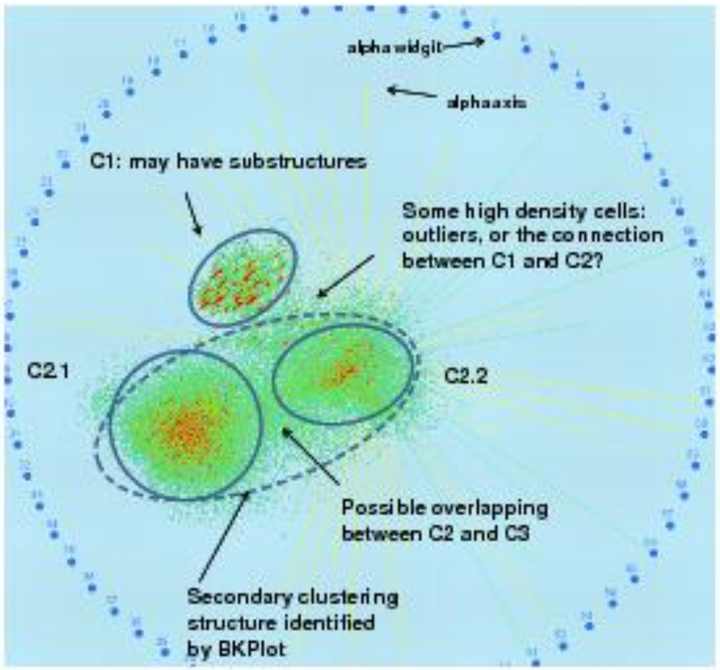


Fig. 6. Visualization and Analysis of Census data with the VISTA system.

25 million records

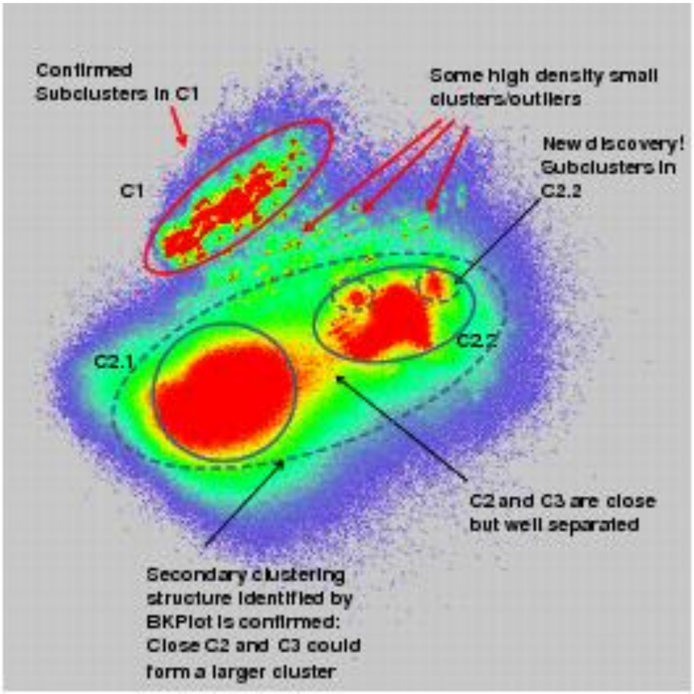


Fig. 7. Visualization and Analysis of 25 Million Census records (in 1000x1000 resolution).

Demos

- http://www.youtube.com/watch?v=SPMKn4dlfQk
- http://www.youtube.com/watch?v=EIaWR1gRc24

Performance of MapReduce

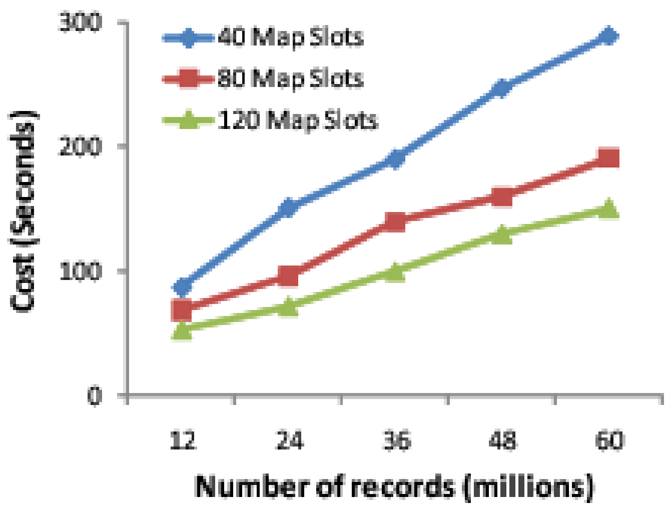


Fig. 8. Running time vs data size for Census$_{ext}$ data (RandGen for 100 frames).

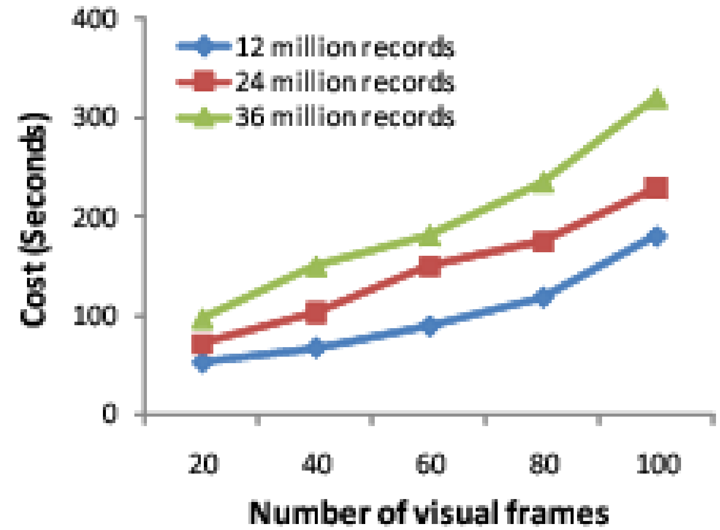


Fig. 9. Running time vs the number of frames for Census$_{ext}$ data.

Fixed # of map/reduce slots

•We have systematically studied the cost model of MapReduce programs - the IEEE CLOUD paper: **"Towards Optimal Resource Provisioning for Running MapReduce Programs in Public Clouds"**

Effect of Resolution

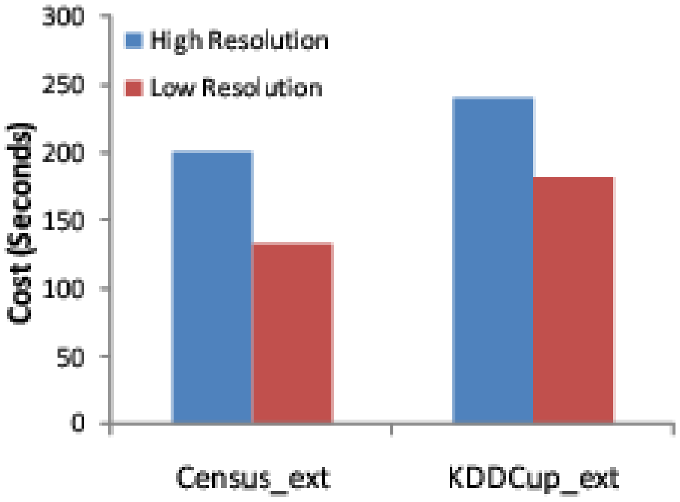


Fig. 10. Cloud processing time vs resolutions for RandGen (100 frames, Census_ext: 25 Million records, KDDCup_ext: 40 Million records).

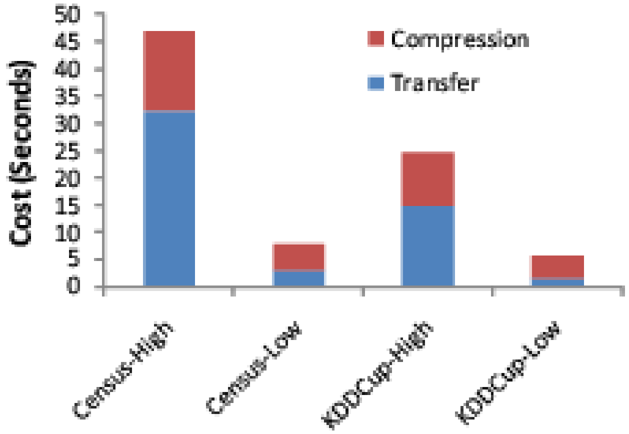


Fig. 11. Cost breakdown (data transfer + compression) in app server processing (100 frames, Census-*: 25 Million records, KDDCup-*: 40 Million records, *-high: 1000x1000 resolution, *-low: 250x250 resolution).

Cost of generating 100 frames

High: 1000x1000
Low: 250x250

Average # of filled aggregation buckets

MapReduce+
Transfer to AppServer+
Compression

	resolution	frame size	compressed frames	total time(sec)
$Census_{ext}$	High	320K	100MB	247
	Low	25K	9.7MB	141
KDD_{ext}	High	143K	45MB	265
	Low	12K	4.6MB	188

Table 1. Summary of the RandGen experiment.

* MapReduce can compress the output, which may reduce the total time, but we haven't tested yet.

Cost of subset selection

of buckets with the
Resolution : 1000x1000

Direct: selected data very small
Sampling: medium size
SS-RG: too large, need RandGen

Average time of
MapReduce
processing

	Size of Selected Area	# of Cloud Operations			D&S Time(sec)
		Direct	Sampling	SS-RG	
Census_{ext}	13896 ± 17282	4	34	22	36
KDD_{ext}	6375±9646	9	33	18	43

Table 2. Summary of the subset selection experiment.

- The numbers are on the top level exploration
-For low level exploration
the number of "direct" cases will significantly increase,
the number of Subset RandGen will decrease and disappear

Conclusion

- A framework for visually exploring remote large data with the cloud infrastructure
  - Based on VISTA visualization model, easy to parallelize
  - A batch-interactive exploration model
  - RandGen algorithm and Subset Selection algorithm (implemented with MapReduce)
  - Analysis of the number of cloud operations in the hierarchical exploration model
- Experimental study on performance and scalability
- Future work
  - Larger scale experiments
  - Effectively use feedback information