

# Supporting Online Analytics with User-Defined Estimation and Early Termination in a MapReduce-Like Framework

Yi Wang, Linchuan Chen, Gagan Agrawal  
The Ohio State University

# Outline

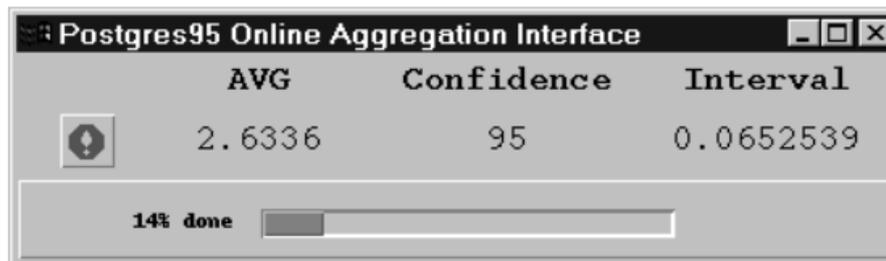
- Introduction
- Challenges
- System Design
- Experimental Results
- Conclusion

# Two Trends of Big Data Analytics

- More Interactive
  - Online processing is favored
  - Approximate algs are great
    - 100% accuracy is often not necessary
    - Trade accuracy for performance and responsiveness
- More Cloud-Based
  - Pay as you go
  - Can save more with early termination

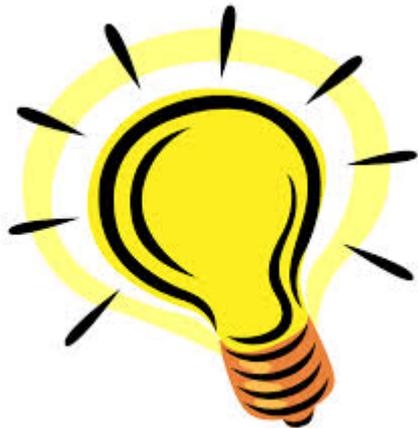
# Online Analytics

- Processes Data Incrementally
  - Processes a (small) portion at a time
  - Provides online estimate
- Early Termination
  - Stops when the estimate is good enough
  - Saves time and resources
- Example: Online Aggregation



# MapReduce

- Major Advantage
  - Hides all the parallelization complexities by simplified API
- Limitation
  - Mainly designed for batch processing



Online  
Analytics



# Challenges

- Online Mode + Batch Mode
  - Online mode is similar to iterative processing
    - Each iteration processes a portion and estimates
  - Backward compatibility
    - Old batch mode code should still work
    - Should not break the original API
- Runtime Estimation + Synced Termination
  - No more communication among workers after shuffling
  - How to obtain a global estimate at each iteration?
  - How to terminate each node synchronously?

# Challenges Cont'd

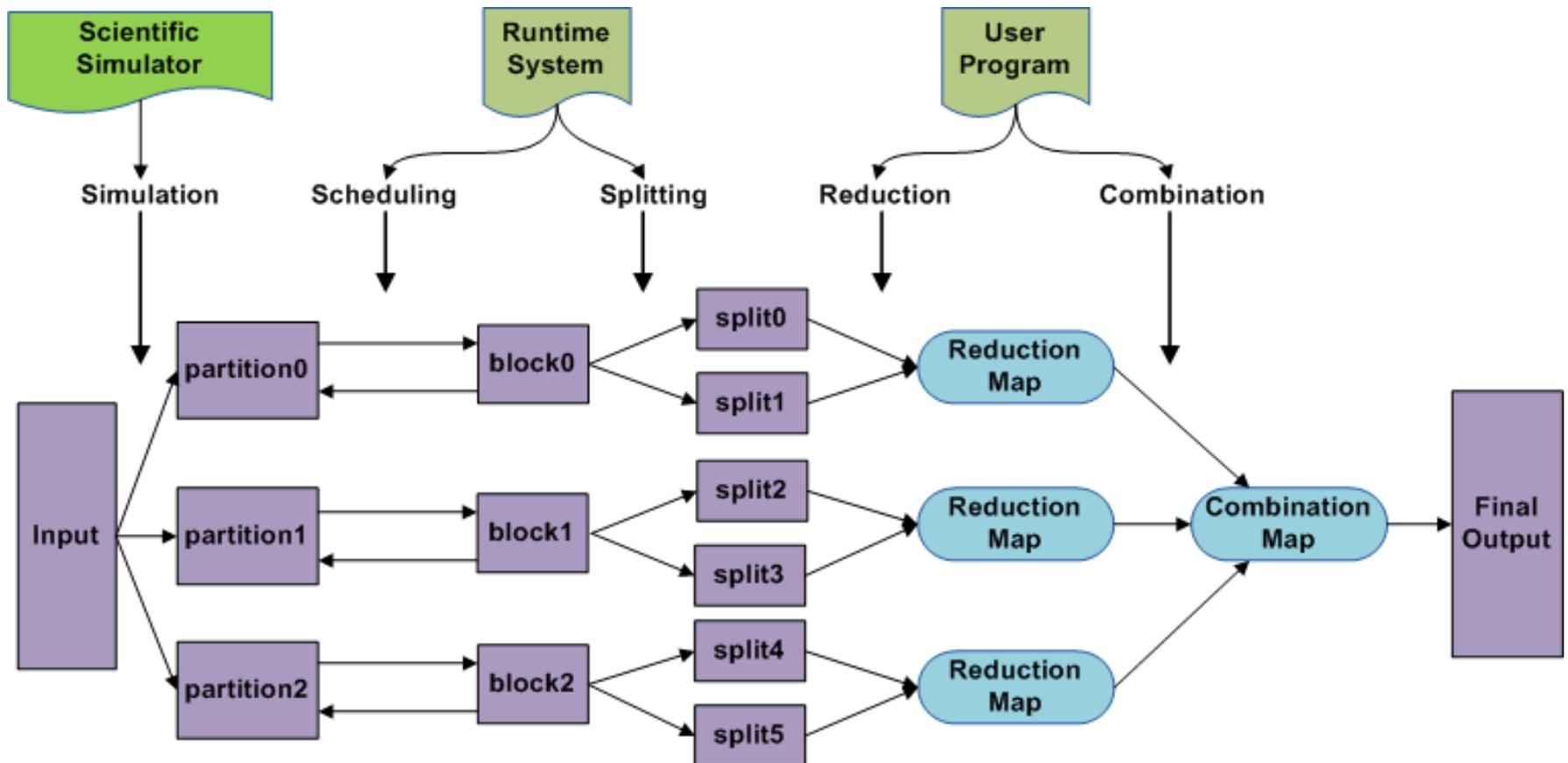
- Associates Termination Condition with the App Semantics
  - Accuracy
    - Online aggregation
  - Threshold (not necessarily accuracy)
    - Top-k
  - Convergence or delta of a series of estimates
    - Clustering and regression

# Bridging the Gap

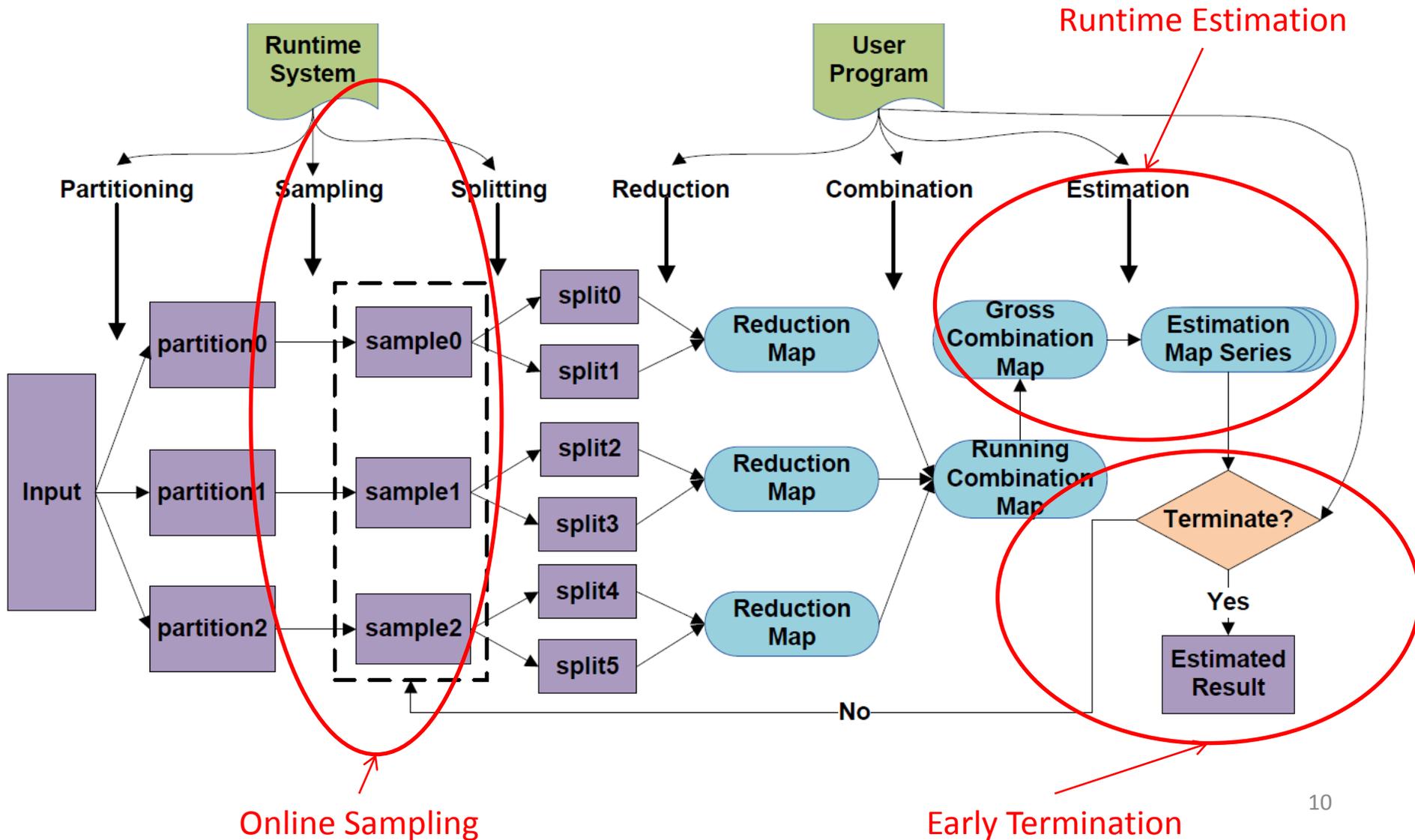
- Addresses All the Challenges
  - Applies online sampling and adds 2 optional functions
    - Customizable runtime estimation and early termination
    - Falls back to batch mode by default
  - Master node combines local estimates and controls termination
    - Unlike ETL, the output of online analytics is usually small
  - Evaluates termination condition based on the estimates of the past few sampling iterations

# Precursor System: Smart

- Smart is Our MR-like Framework (SC'15)



# System Overview



# Online Sampling

- Stratified Sampling without Replacement
  - Stratified
    - Good accuracy
  - Without Replacement
    - Sampled data continues to increase strictly until early termination or full scan
    - Sample size is constant -> no tricky dynamic sample size adjustment

# Runtime Estimation and Early Termination

- Runtime Estimation
  - Further reduces sampling data in the time dimension
  - Reuses the same merge (reduce) func. to reduce all the data processed so far
  - Estimates based on the total input size and the running reduction results
- Early Termination
  - Evaluates the most recent snapshot(s) of estimates

# Extended System APIs

- Running Estimation Func.
  - Gives total input size + stats of the data processed so far
  - Estimates based on the partial input
  - E.g., totally **10 GB** data, after processing **1 GB** data, sum = **1**  estimated sum = **10**
  - By default, returns the results of the data processed so far (i.e., no magnification or adjustment)

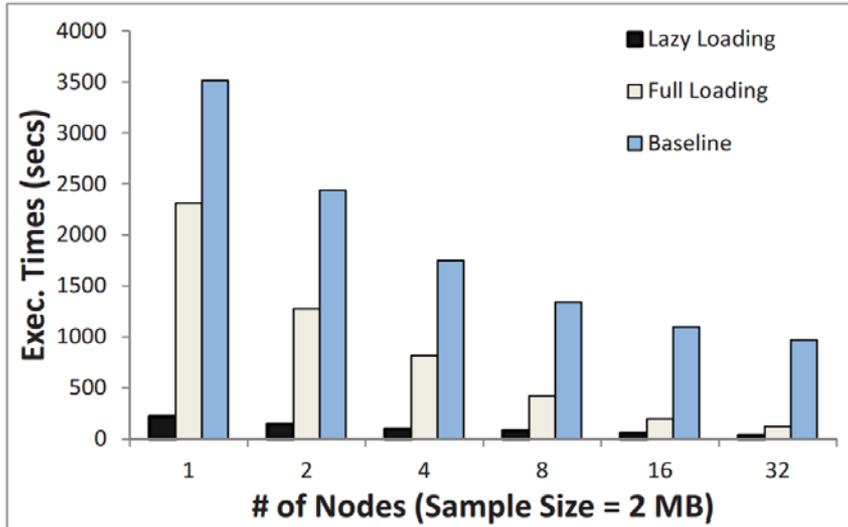
# Extended System APIs (Cont'd)

- Early Termination Func.
  - Evaluates accuracy or threshold -> retrieves the most recent estimate
  - Evaluates convenience or delta -> retrieves the most recent 2 or more estimates
  - By default, returns false to disable early termination and fall back to the batch mode

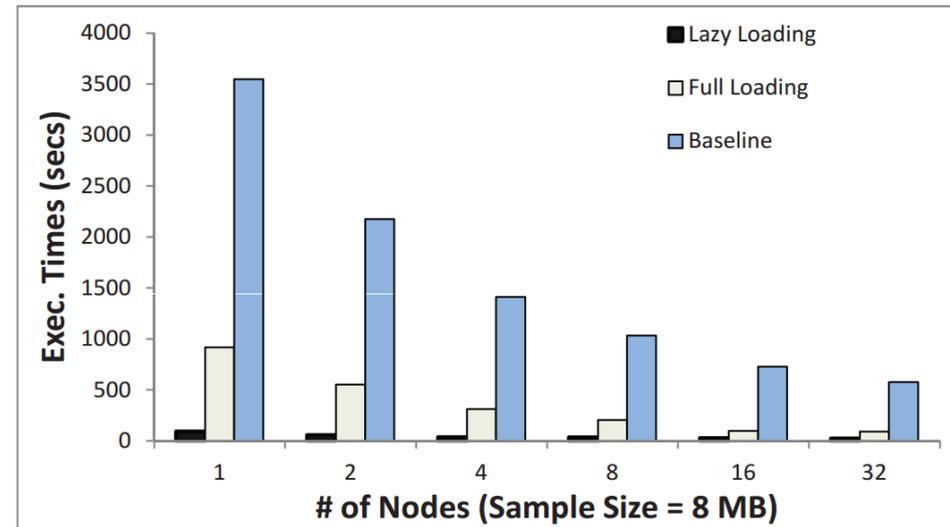
# Experiments

- Setup
  - 32 nodes \* 8 cores, 12 GB memory
  - 112 GB Ocean Dataset
- Apps:
  - Online Aggregation: **accuracy** based termination
  - Top-k: **threshold** based termination
  - K-means: **convergence** based termination

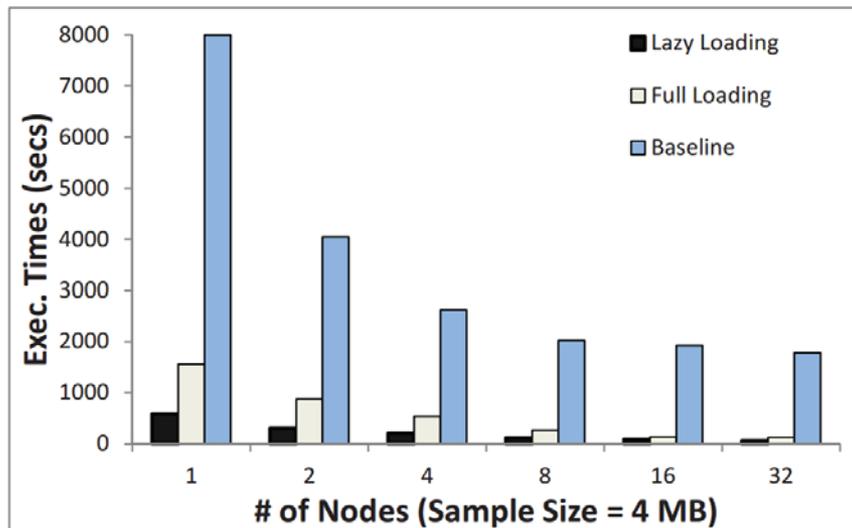
# Performance Evaluation



(a) Online Aggregation



(b) Top-K

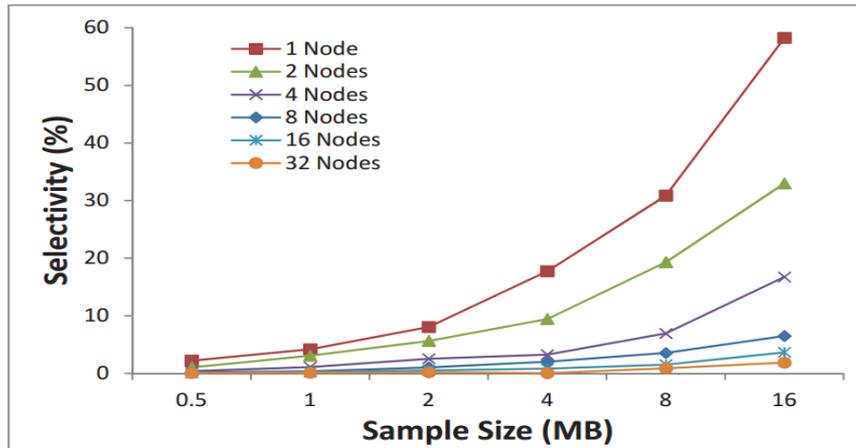


(c) K-Means

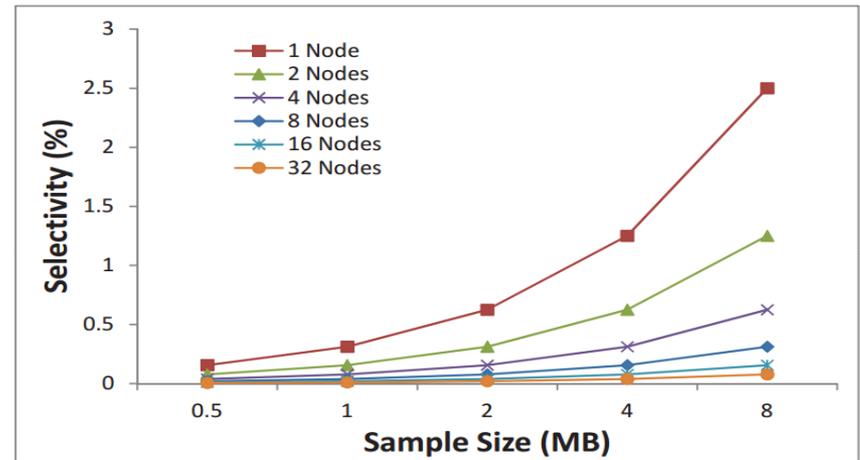
- **Lazy Loading:** partial I/O + partial computation
- **Full loading:** full I/O + partial computation
- **Baseline:** batch processing
- Lazy loading is up to **30x** faster than baseline, and up to **11x** faster than full loading

# Which Factors Delay Termination?

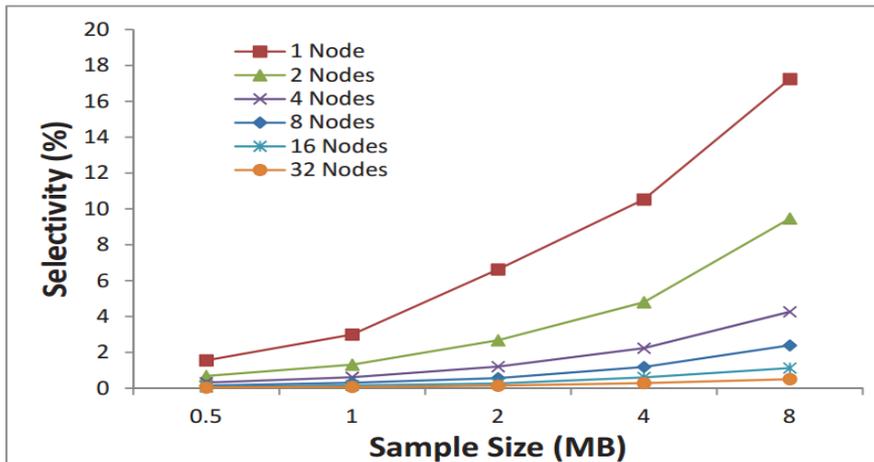
Selectivity = sampled data size / total input size



(a) Online Aggregation



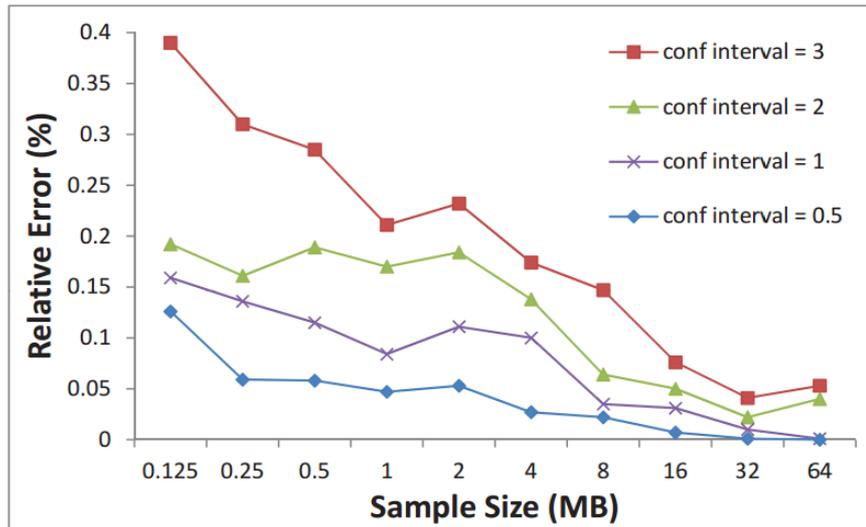
(b) Top-K



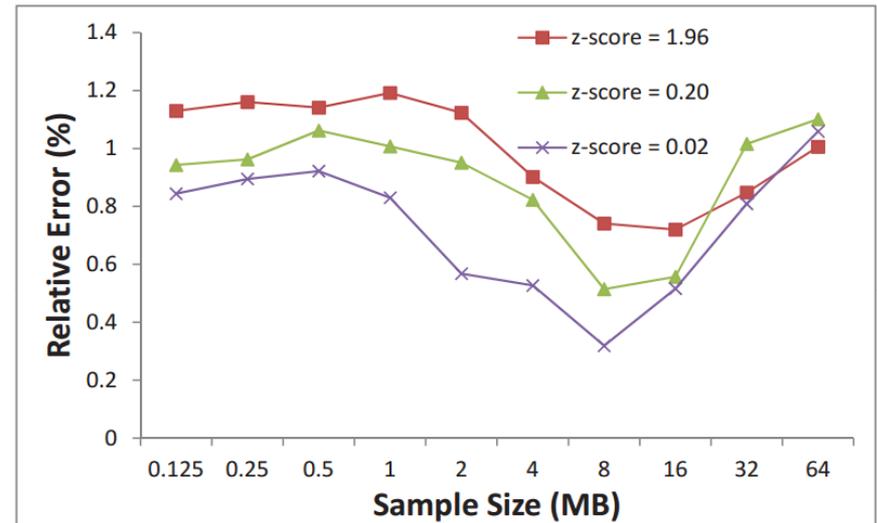
(c) K-Means

- Selectivity  $\propto$  **sample size**
- Selectivity  $\propto$  **# of nodes**

# Which Factors Affect Accuracy?



(a) Online Aggregation



(b) K-Means

- No noticeable diff on varying # of nodes
- More strict **termination condition** => higher accuracy
- **Sample size** for each sampling iteration also somehow affects accuracy

# Conclusion

- Online Analytics + MR
  - Big data analytics can be done incrementally and stopped early to bring timely insight
  - Need for approximate algs on MR
- Critical Extensions
  - Online sampling
  - Extended APIs
    - Flexible runtime estimation
    - Early termination condition w.r.t. app semantics