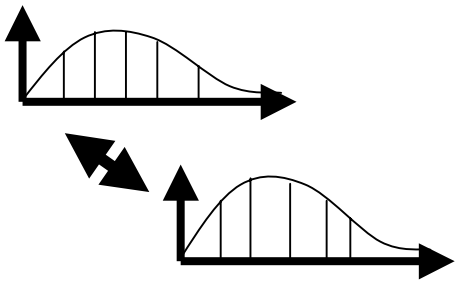
Holistic Aggregates in a Networked World: Distributed Tracking of Approximate Quantiles



Graham Cormode
cormode@bell-labs.com

Minos Garofalakis
minos@acm.org

S. Muthukrishnan
muthu@cs.rutgers.edu

Rajeev Rastogi
rastogi@bell-labs.com

Continuous Distributed Queries

Traditional data management supports *one shot* queries

- May be look-ups or sophisticated data management tasks, but tend to be on-demand
- New large scale data monitoring tasks pose novel data management challenges

Continuous, Distributed, High Speed, High Volume…

Networking Application

Network Operations Center (NOC) of a major ISP:

Monitoring 100s of routers, 1000s of links and interfaces, millions of events / second.

Monitor all layers in network hierarchy: from physical properties of fiber, to packet forwarding at routers, to VPN tunnels, etc.

Also applies to data centers/web caching (eg Akamai, Google): monitor 1000s of nodes, carry out sophisticated load balancing

- both for performance and for failure resiliance

Other Monitoring Applications

Sensor networks

- Monitor habitat and environmental parameters
- Track many objects, intrusions, trend analysis…

Utility Companies

- Monitor power grid, customer usage patterns etc.
- Alerts and rapid response in case of problems

Common Aspects / Challenges

Monitoring is Continuous…

- Need real time tracking, not one-shot query/response

…Distributed…

- Many remote sites, connected over a network but with communication constraints

…Streaming…

- Each site sees a high speed stream of data, and may be resource (CPU/Memory) constrained.

…Holistic…

- Queries over whole distribution (eg. median)

Problem

Need to monitor complete distribution of data

- Eg, counting IP traffic from one address is easy;
- summarizing whole traffic distribution is challenge

Hardwired solutions/measurements not sufficient

But… Exact answers are not needed

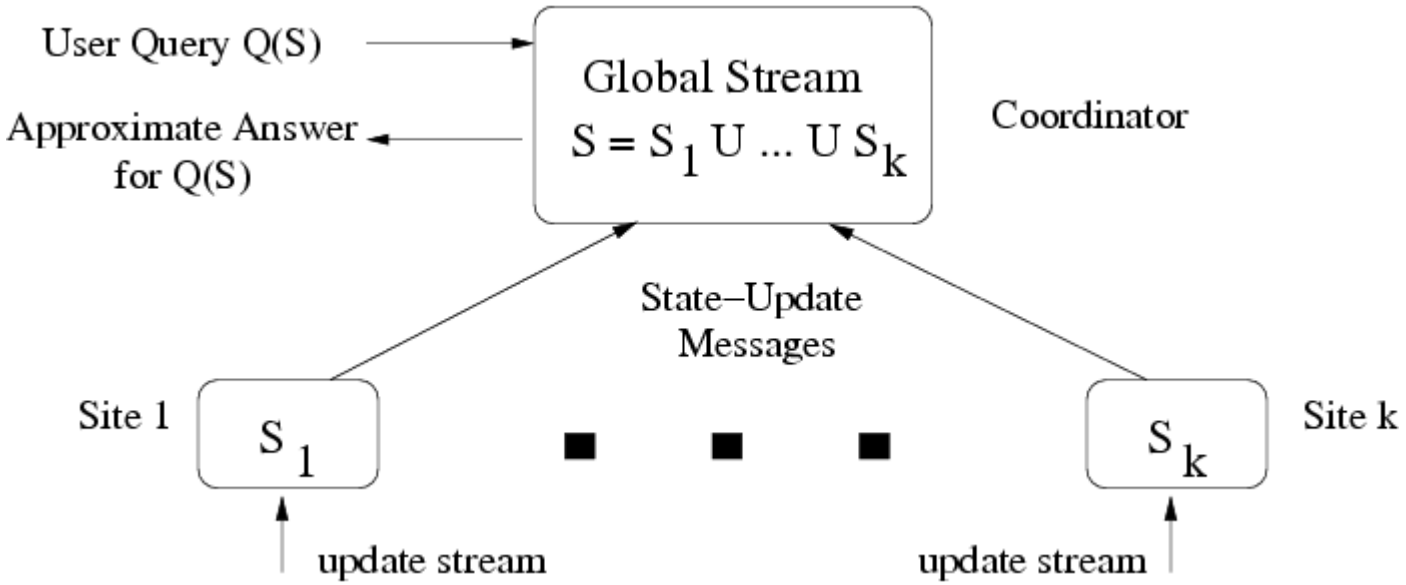
- Approximations with accuracy guarantees suffice
- Allows a tradeoff between accuracy and communication/processing cost

Prior Work

	continuous	distributed	streaming	holistic	
Distributed top-k & quantiles	X	✓	✓	✓	GK04, MSDO05
Streaming top-k & quantiles	✓	X	✓	✓	GK01, MM02
Distributed filters	✓	✓	X	✓	OJW03
Distributed top-k	✓	✓	✓	X	BO03

We aim for all four properties!

Architecture



Streams at each site add to (or subtract from) multisets S_j

(More generally, can have hierarchical structure)

Quantile Queries

Quantiles summarize data distribution concisely.

Focus on *rank queries* — given value v, estimate

$$\text{rank}(v) = \text{number of items} < v \text{ in } \cup_j S_j$$

Allow approximation: $\text{rank}(v) \pm \varepsilon N$

- N = total number of items = $|S|$
- Small space solutions for centralized stream [GK01]

Can use rank queries to answer arbitrary quantile queries, ie, search for v so that $\text{rank}(v) \approx \phi N$

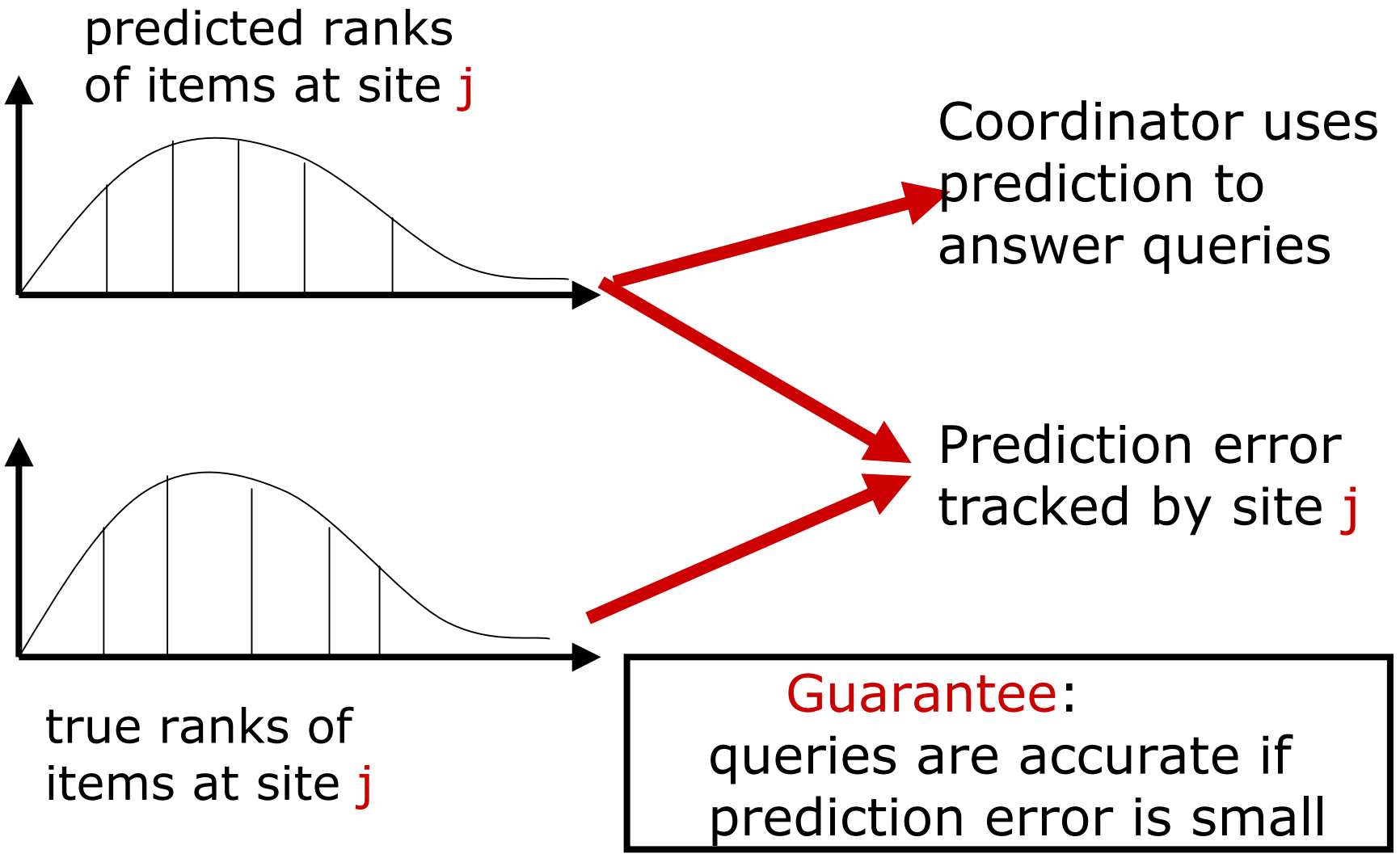
Goal: Minimize communication overhead, reach stability (zero communication) if possible.

Overview of Scheme

Remote sites monitor local stream, compare ranks of certain items to *predicted ranks*

- Use summaries to communicate…
  Much smaller cost than sending exact values
- No/little global information
  Sites only use local information, avoid broadcasts
- Stability through prediction
  If behavior is as predicted, no communication

Prediction


predicted ranks
of items at site j
Coordinator uses
prediction to
answer queries
Prediction error
tracked by site j
true ranks of
items at site j
Guarantee:
queries are accurate if
prediction error is small

Tracking Scheme

Summary used is local quantiles at site j, $\{v_{i,j}\}$
  $i\phi$ for i = 1 to $1/\phi$ eg 5%, 10% … 95% quantiles

Use a simple model (specified later) to predict current rank of each $v_{i,j}$:

$$\text{Predicted rank of } v_{i,j} = r_j^p(v_{i,j})$$

Local site shares model, communicates only if

$$| r_j^p(v_{i,j}) - r(v_{i,j})| > \theta N_j$$

θ = "lag" between remote site and coordinator

Communication tradeoff is between ϕ and θ

Query Answering

For query v coordinator finds i' for each site j so

$$v_{i',j} < v < v_{i'+1,j}$$

and estimates

$$\text{rank}(v) = \tfrac{1}{2} \Sigma_j \, (r_j^p(v_{i',j}) + r_j^p(v_{i'+1,j}))$$

Claim: Provided $(r_j^p(v_{i+1,j}) - r_j^p(v_{i,j})) \leq 2\phi \, N_j$ then error in this approximation is at most $(\phi + \theta)N$

Proof outline: rank(v) = sum of ranks at each site. Error is difference in $\text{rank}(v_{i',j})$ and $\text{rank}(v_{i'+1,j})$. Applying prediction bounds gives result.

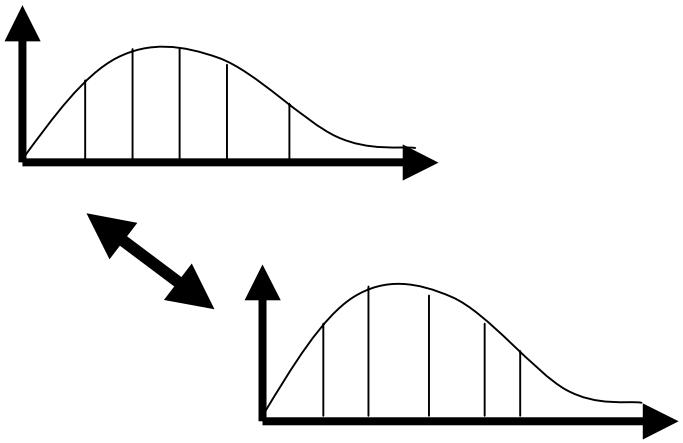
Prediction Models

Zero Information:

Predict $r_j^p(v_{i,j}) = i\phi\ N_j$ (old rank)

(assumes no new items ever arrive)

Will be proved wrong eventually, but gives a baseline communication cost to compare against



Communication Bounds

With Zero Information model:

- Can show number of communications is $1/\theta \ln N_j$
- Each message is $1/\phi$ quantile values
- Total cost is $1/(\theta\phi) \ln N_j$
- To minimize cost and guarantee error $\varepsilon = \phi + \theta$, set $\phi = \theta = \varepsilon/2$
- Total cost $= O(1/\varepsilon^2 \ln N_j)$

Prediction Models 2

Rate based model

Assume that the quantile values stay same, ranks grow with constant rate δ_j at site j. So:

$$r_j^p(v_{i,j}) = i\phi\ (N_j + \delta_j t_j)$$

If number of new updates = $\delta_j t_j$ and distribution is roughly the same, will be a better prediction.

How to find δ_j? We used a recent history, or average over all time…

Many other models possible, not main focus here

Approximate Local Summaries

So far, we assumed each site tracks local quantiles exactly.

In general, need solutions to work in small space.

Can use an approximate stream alg for tracking quantiles, eg [GK01]

Reapply the analysis from before, but now sites have approximate ranks instead of exact ranks.

If summary error is α, total error is $\varepsilon = \alpha + \phi + \theta$

Hierarchical Networks

Have each level run the protocol with its parent as coordinator, using θ_l and ϕ_l

Using previous result, error guarantee is

$$\alpha_{l-1} = \alpha_l + \theta_l + \phi_l$$

Error at root (level 0) is $\Sigma_{l=1}^{h} \theta_l + \phi_l$

Using simplifying assumptions, find optimal settings of θ_l and ϕ_l

Guarantee overall error ε while minimizing total communication, or minimizing maximum communication by any node

Hierarchical Results

To minimize maximum transmission cost:

$$\theta_l = \phi_l = \frac{\epsilon \sqrt{\ln \frac{N}{k_l}}}{2 \sum_j \sqrt{\ln \frac{N}{k_j}}}.$$

To minimize total communication cost:

$$\theta_l = \phi_l = \frac{\epsilon (k_l \ln \frac{N}{k_l})^{1/3}}{2 \sum_j (k_j \ln \frac{N}{k_j})^{1/3}}.$$

Experimental Study

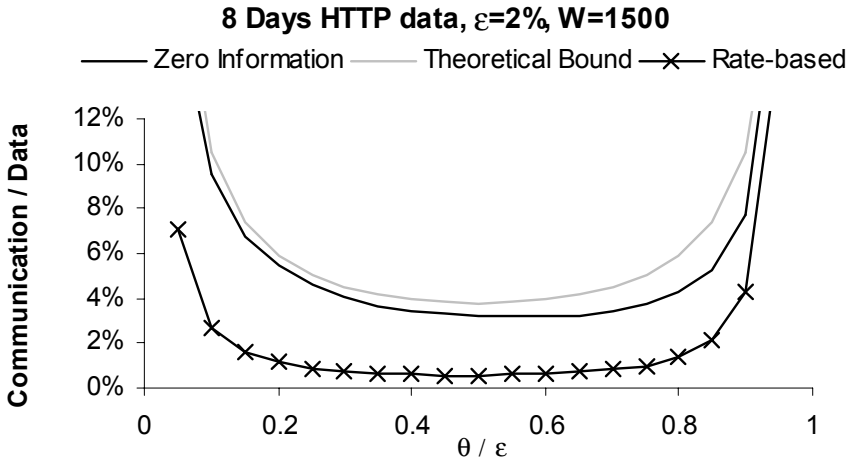
Implemented a simulator for continuous distributed tracking in C

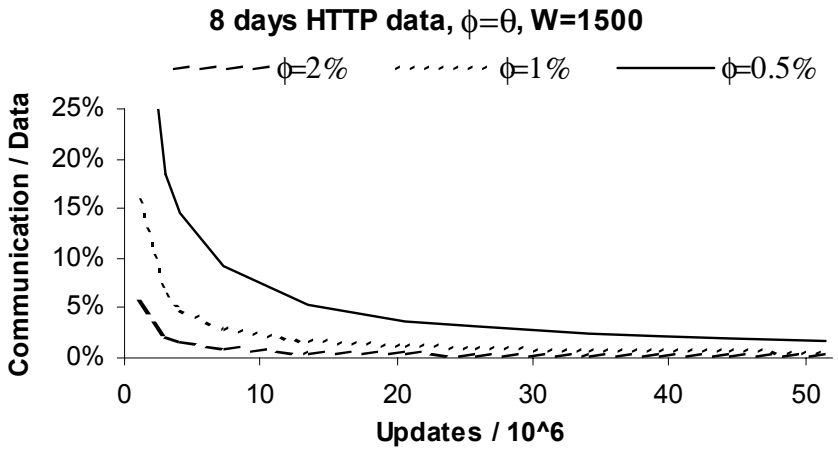
Measured communication cost compared to cost of sending all updates

Ran on:

- World cup 1998 HTTP request data (23 sites)
- Dartmouth wireless SNMP traces (200+ sites)
- Synthetic data – Zipfian distribution, Gaussian Delays, randomly changing parameters (1 site)

Experimental Results





Close to predicted $1/\varepsilon^2$ cost

Rate based considerably better than zero-information, itself much better than sending all updates.

Conclusions

Local information is sufficient, initial attempts using global information exchanges were much too costly

Quantiles encompass heavy hitters / frequent items, so can apply to those problems.

Recent work extends this approach to general aggregates by tracking sketches (in VLDB05)

Extensions

Using only local information seems to work, but surely giving something up by not using correlations between sites?

Other aggregates may be of interest, but many already captured by quantiles and sketches.

Sliding window version also fits in our model, but need to test how practical compared to sending all updates… perhaps new approaches needed?