

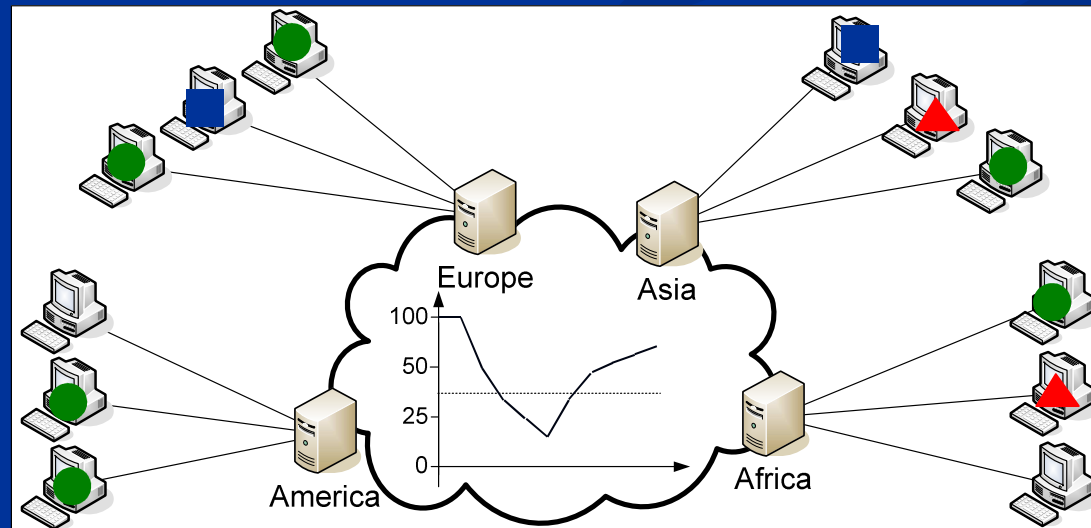
A scenic view of a river flowing through a dense forest in a mountain valley. The river is rocky and turbulent, surrounded by lush green trees and mountains in the background.

Monitoring Threshold Functions in systems comprised of Distributed Data Streams

Daniel Keren, Haifa U
Tsachi Sharfman, Technion
Assaf Schuster, Technion

Web Page Frequency Counts

- Mirrored web site
- Mirrors record the frequency of requests for pages
- Detect when the global frequency of requests for a page exceeds a predetermined threshold



Air Quality Monitoring

- Sensors monitoring the concentration of air pollutants.
- Each sensor holds a data vector comprising measured concentration of various pollutants (CO_2 , SO_2 , O_3 , etc.).
- A function on the *average* readings determines the Air Quality Index (AQI)
- Issue an alert in case the AQI exceeds a given threshold.

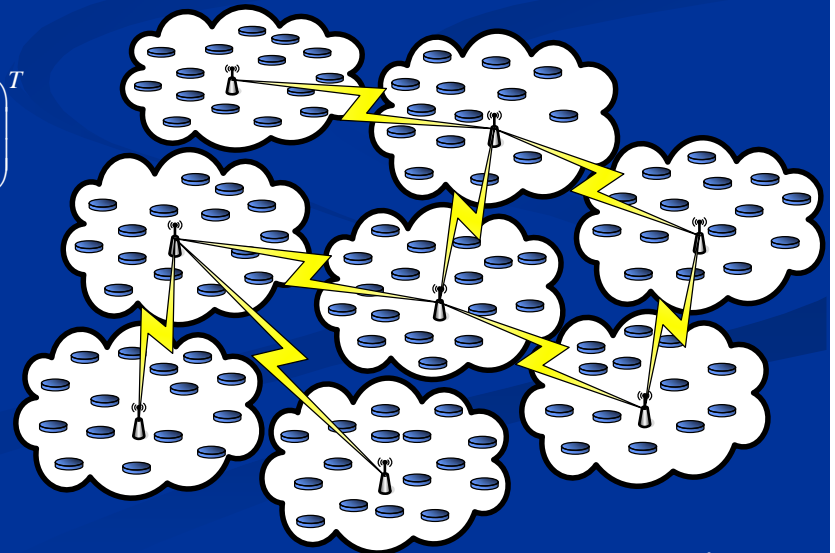


Sensor Networks

- Sensors monitoring the temperature in a server room (machine room, conference room, etc.)
 - Ensure uniform temp.: monitor variance of readings
 - Alert in case variance exceeds a threshold
- Temperature readings by n sensors x_1, \dots, x_n
- Each sensor holds a data vector $v_i = (x_i^2, x_i)^T$
- The *average* data vector is $v =$
- $Var(\text{all sensors}) =$

$$\frac{1}{n} \sum_{i=1}^n x_i^2 - \left(\frac{1}{n} \sum_{i=1}^n x_i \right)^2$$

$$\left(\frac{1}{n} \sum_{i=1}^n x_i^2 \quad \frac{1}{n} \sum_{i=1}^n x_i \right)^T$$



Search Engine



- Distributed datacenter/warehouse
 - 10Ks horizontal partitions
 - “Our logs are larger than any other data by orders of magnitude. They are our source of truth.” Sridhar Ramaswamy. **SIGMOD’08 keynote on “Extreme Data Mining”**
- Mining the logs: Compute pairs of keywords for which the correlation index is high
- Thousands simultaneous tasks
 - “Network bandwidth is a relatively scarce resource in our computing environment”. Dean and Ghemawat. **MapReduce paper, OSDI’04**

Cloud Computing

- Amazon's Elastic Compute Cloud – EC2
- Amazon's Simple Storage Service – S3



[Amazon Web Services](#) » [Service Health Dashboard](#)

Amazon S3 Availability Event: July 20, 2008

Amazon S3 Availability Event: July 20, 2008

"At 8:40am PDT, error rates in all Amazon S3 datacenters began to quickly climb and our alarms went off. By 8:50am PDT, error rates were significantly elevated and very few requests were completing successfully. By 8:55am PDT, we had multiple engineers engaged and investigating the issue. Our alarms pointed at problems processing customer requests in multiple places within the system and across multiple data centers. While we began investigating several possible causes, we tried to restore system health... At 9:41am PDT, we determined that servers within Amazon S3 were having problems... By 11:05am PDT, all server-to-server communication was stopped, request processing components shut down, and the system's state cleared..."

Quiescence

Ad-Hoc Mobile P2P Networks

Peer-to-peer network invites drivers to get connected

CarTorrent could smarten up our daily commute, reducing accidents and bringing multimedia journey data to our fingertips

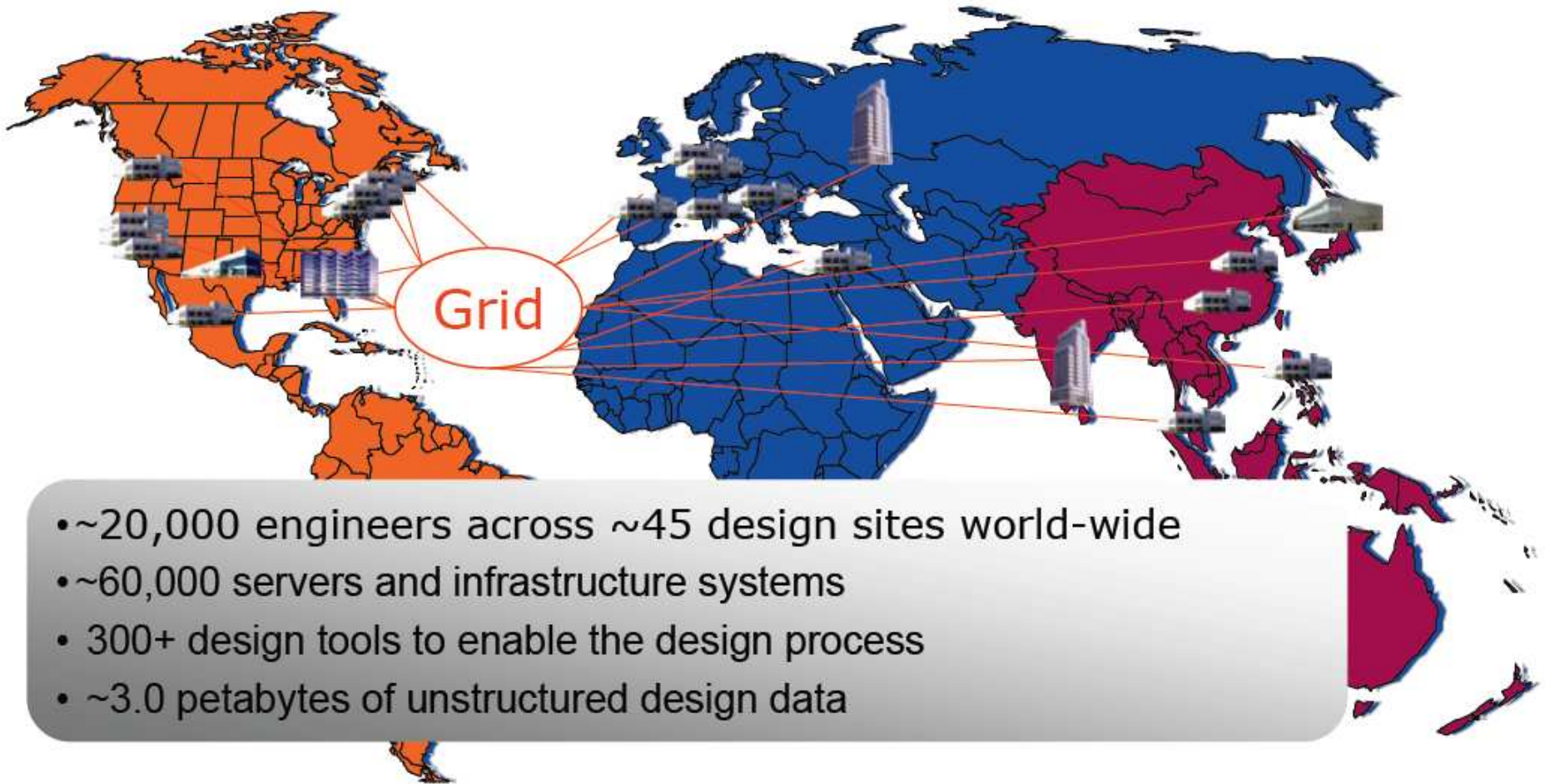
- Laura Parker
- [The Guardian](#),
- Thursday January 17 2008

"The name BitTorrent has become part of most people's day-to-day vernacular, synonymous with downloading every kind of content via the internet's peer-to-peer networks. But if a team of US researchers have their way, we may all be talking about CarTorrent in the not too distant future....."

Researchers from the University of California Los Angeles are working on a wireless communication network that will allow cars to talk to each other, simultaneously downloading information in the shape of road safety warnings, entertainment content and navigational tools...."



Intel Grid



- ~20,000 engineers across ~45 design sites world-wide
- ~60,000 servers and infrastructure systems
- 300+ design tools to enable the design process
- ~3.0 petabytes of unstructured design data

Social Networks

- Can you predict a global event?
- FB collecting 100TB/day...
- IBM System-S guys talk about millions continuous queries...

Centralized Algorithms

- All data is moved to a central location
- Communication overhead
 - Bandwidth
 - Power
 - Cycles
- Centralized resources
- Privacy issues

Distributed Algorithms

- Local filtering
- Local constraints
- Local data
- Local decisions
- Etc.
- But do not forget to maintain correctness!!!

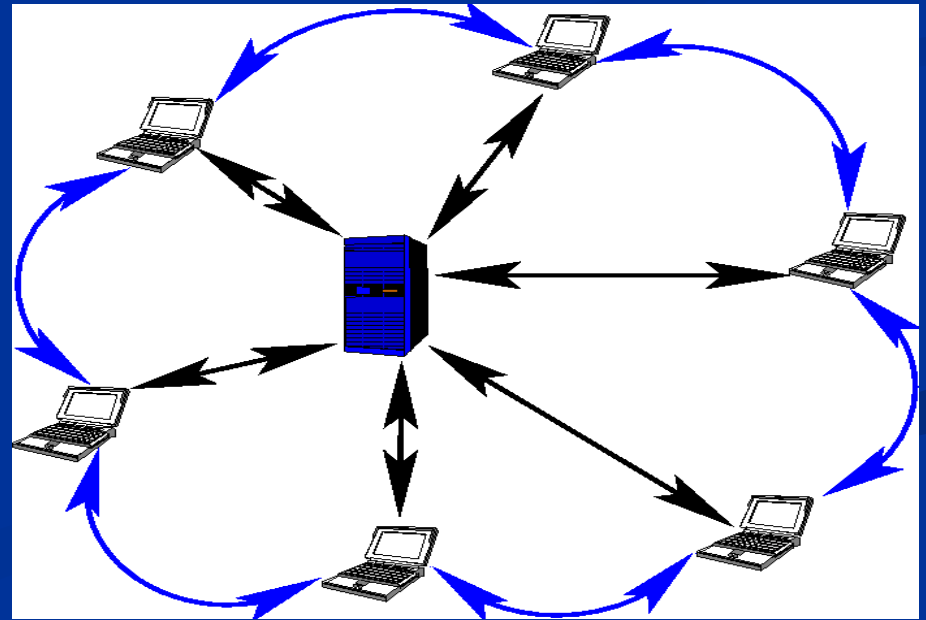
Mining HUGE Networks

- Peer-to-Peer
- Social
- Grid
- Internet-scale routers

Data-Centric view of Peer-to-Peer Networks

- Opportunity for peers: Current technology trend allows peers to collect huge amounts of data and share it
 - Once the territory of companies only
- Opportunity for companies: Mine the logs of P2P networks to improve system operation and customer experience
 - Skype, VOD, distribution networks, etc.

Customer data mining
Mirroring corporates'
Recommendations
(unbiased) - e-Mule
Product Lifetime Cost
(as opposed to CLV)



Data Mining HUGE Database

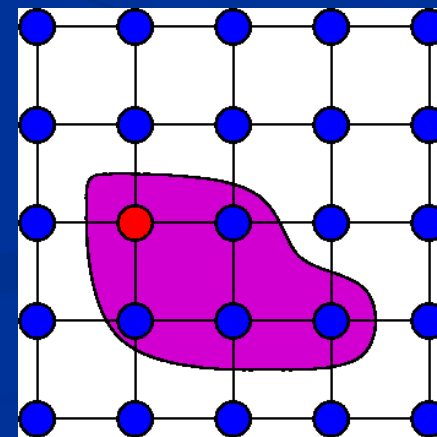
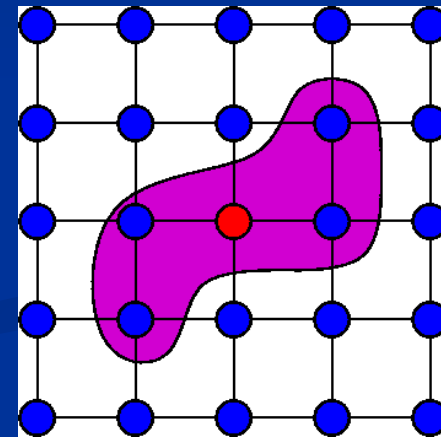
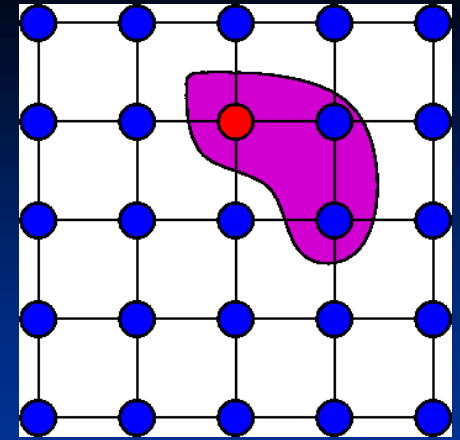
- Impossible to collect the data
- Internet scale
 - NO global operators
 - NO synchronizations
 - NO global communication
 - NO coordination
- Ever changing data and system
 - Failures, crashes, joins, departures
 - Data modified faster than propagated
 - Data streams

What you should not try

- Decomposable statistics (Avg, Var, cross-tables, etc.) can be calculated using (distributed) sum reduction
 - Synchronization
 - Bandwidth requirements
 - Failures
 - Consistency
- Gossip (aka sampling)
 - Rigid schemes
 - Hard to analyse
 - Slow to respond

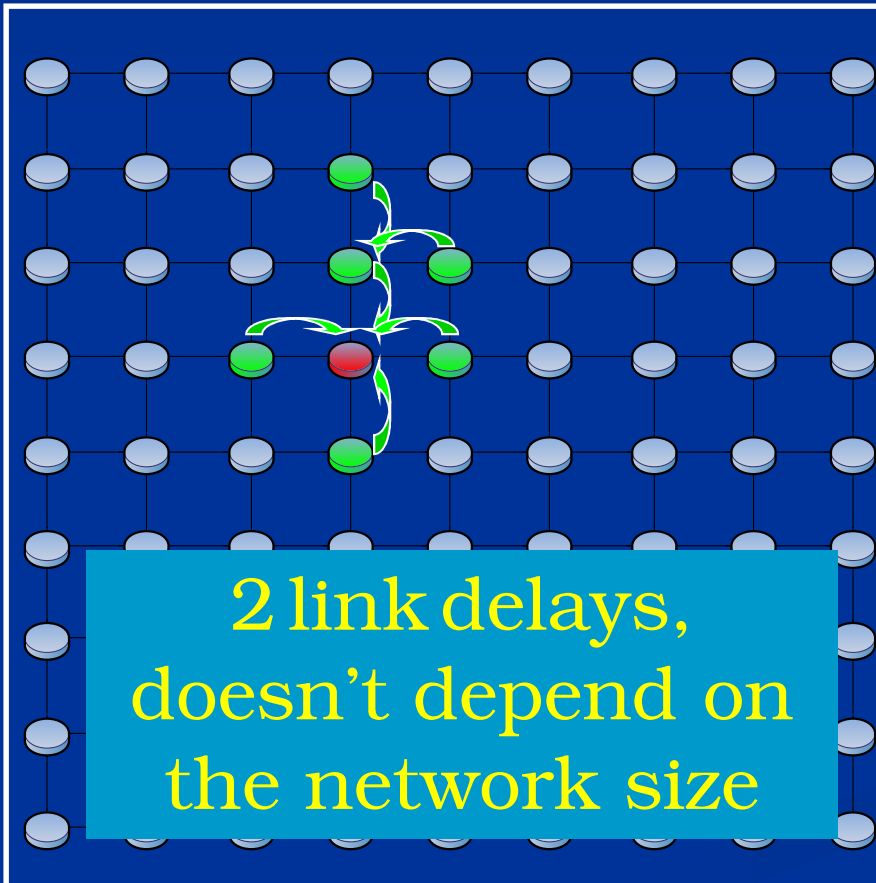
Local Algorithms

- Every peer's result depends on the data gathered from a (small) *environment* of peers
- **Size** of environment may depend on the problem/instance at hand
- Eventual **correctness** guaranteed (assuming stabilization)



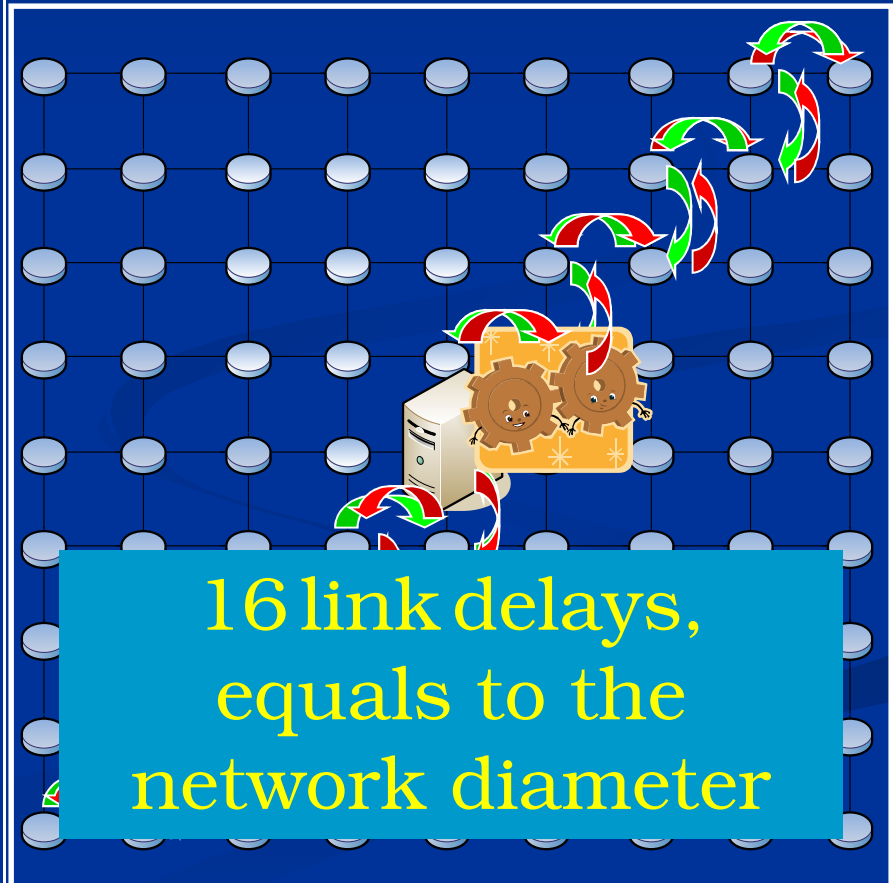
Local vs. Centralized

Local



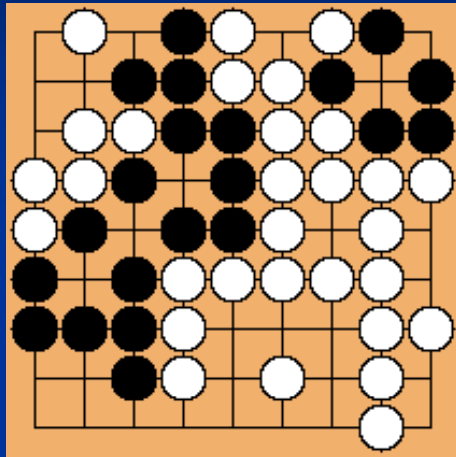
2 link delays,
doesn't depend on
the network size

Centralized

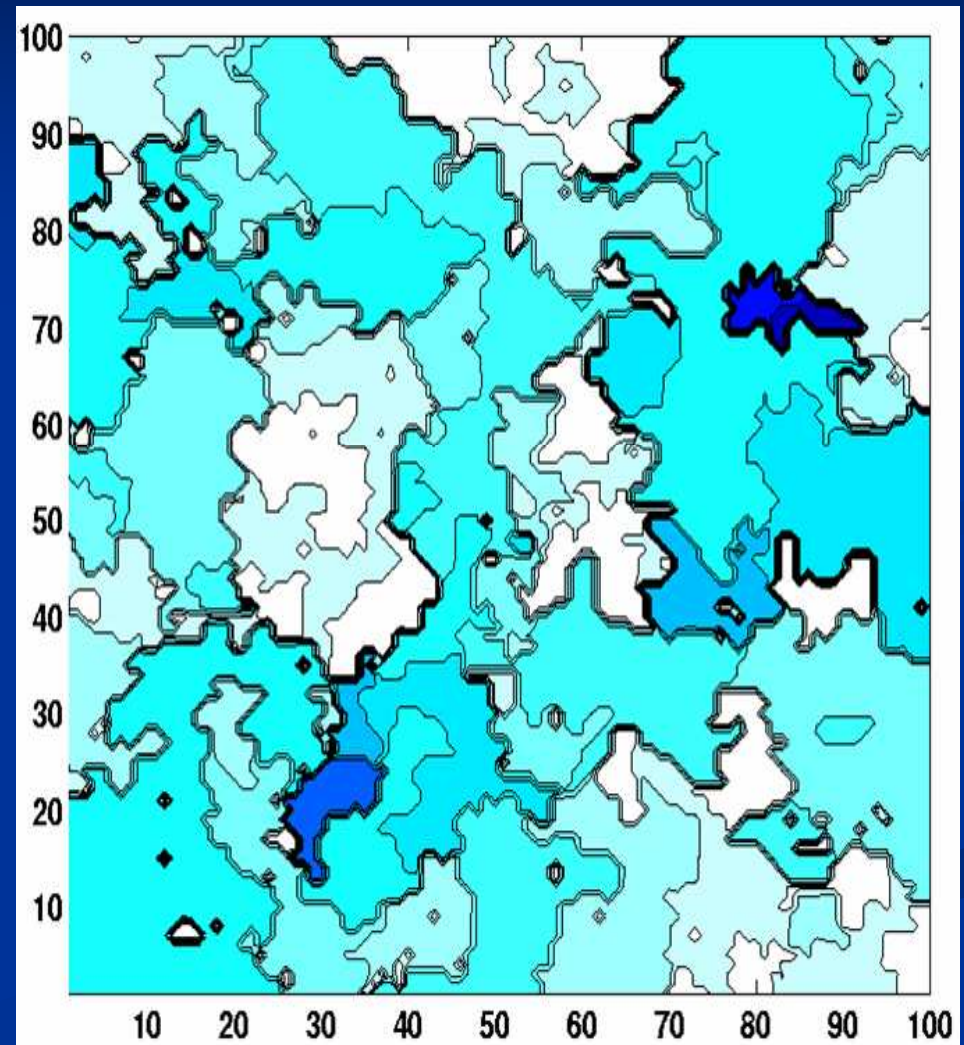


16 link delays,
equals to the
network diameter

Properties of Local Algorithms



- → Scalability
- → Asynchronosity
- → Incrementality
- → Robustness
- → Energy-efficient



Voting

Generalized (weighted) majority voting:

$$\sum_p 1s(p)/votes(p) > \lambda \quad (1 > \lambda > 0, p \in \text{peers})$$

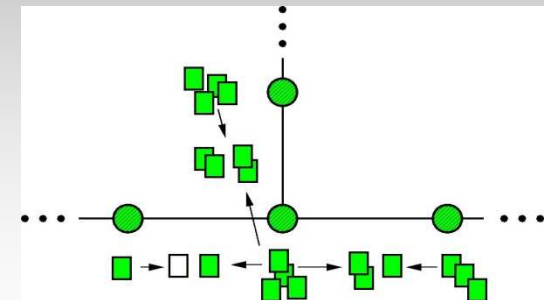
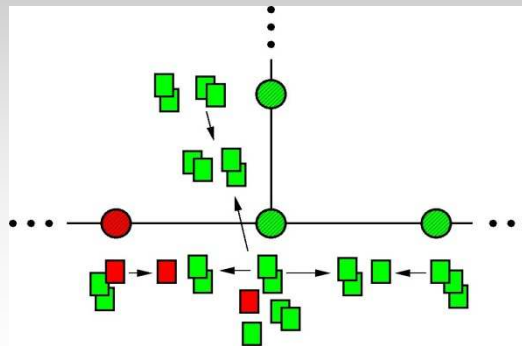
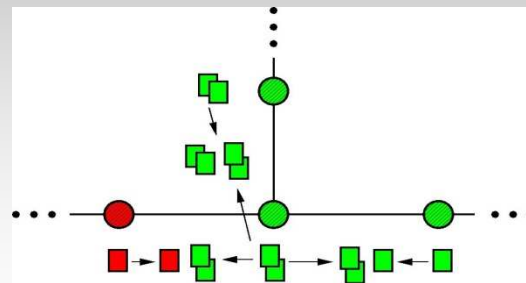
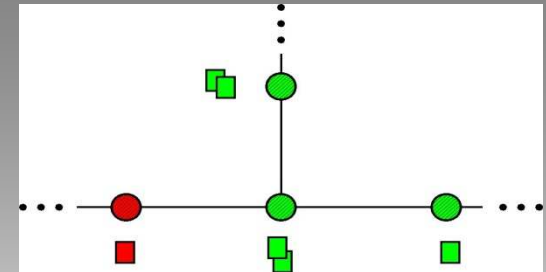
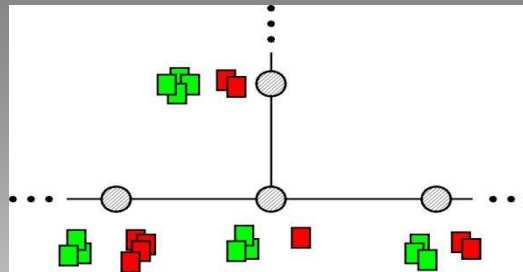
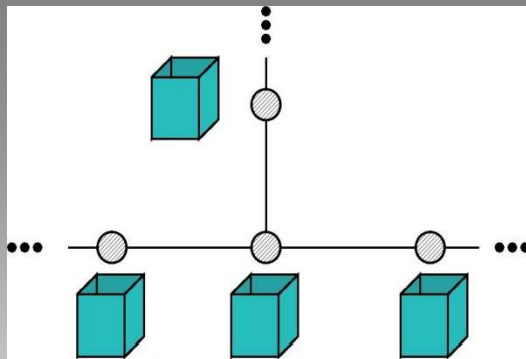
Depends on the significance of the vote:
in a tie all votes must be counted

For many applications a tie is not important:

Inconclusive voting

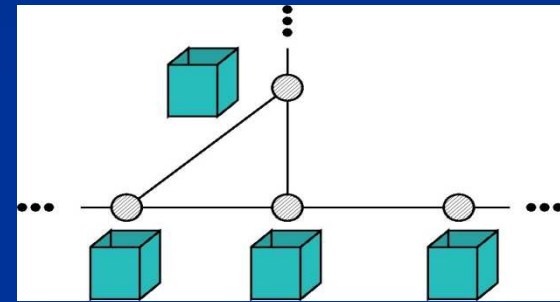


Local Majority Voting (spanning tree)

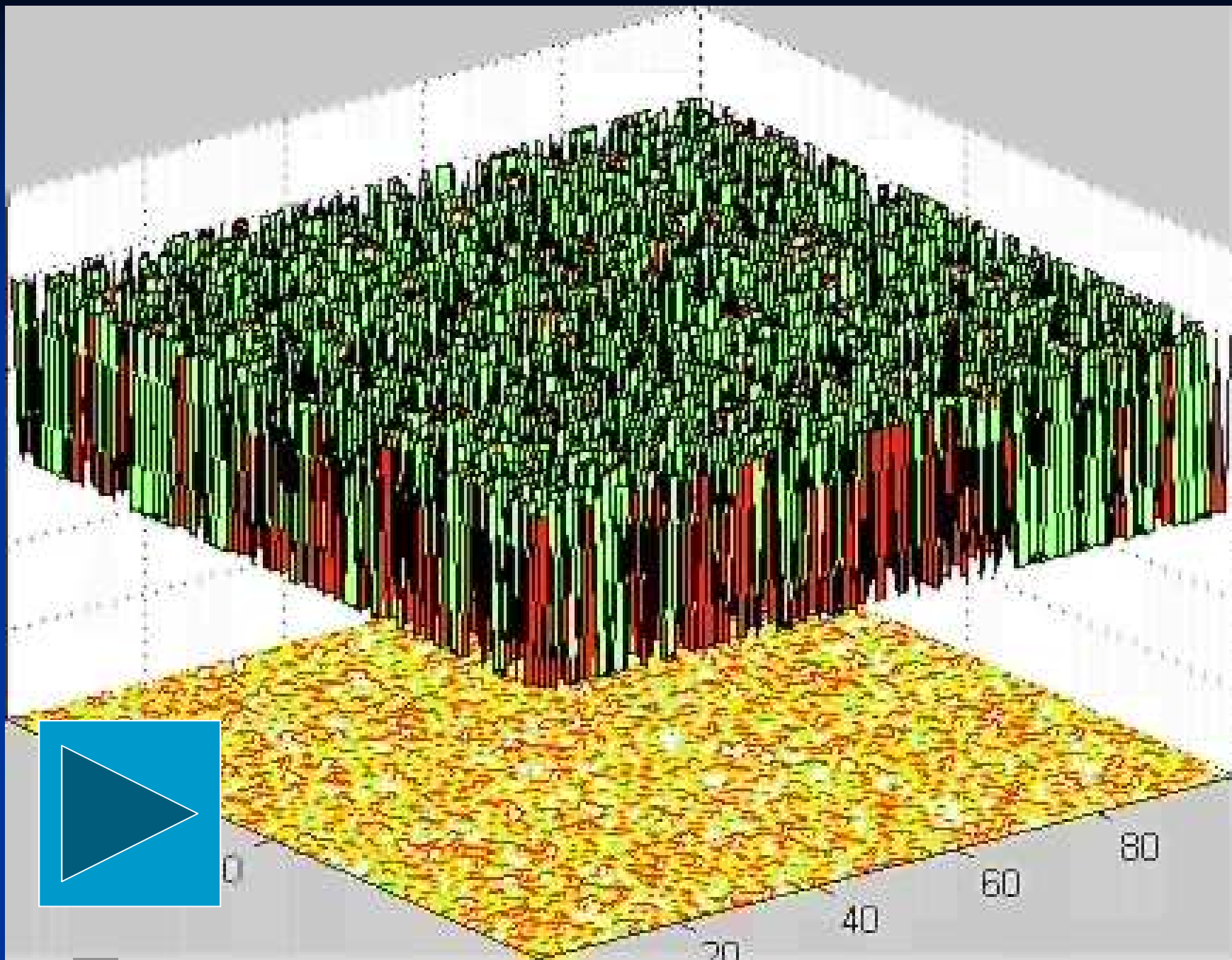


Correctness vs. Locality

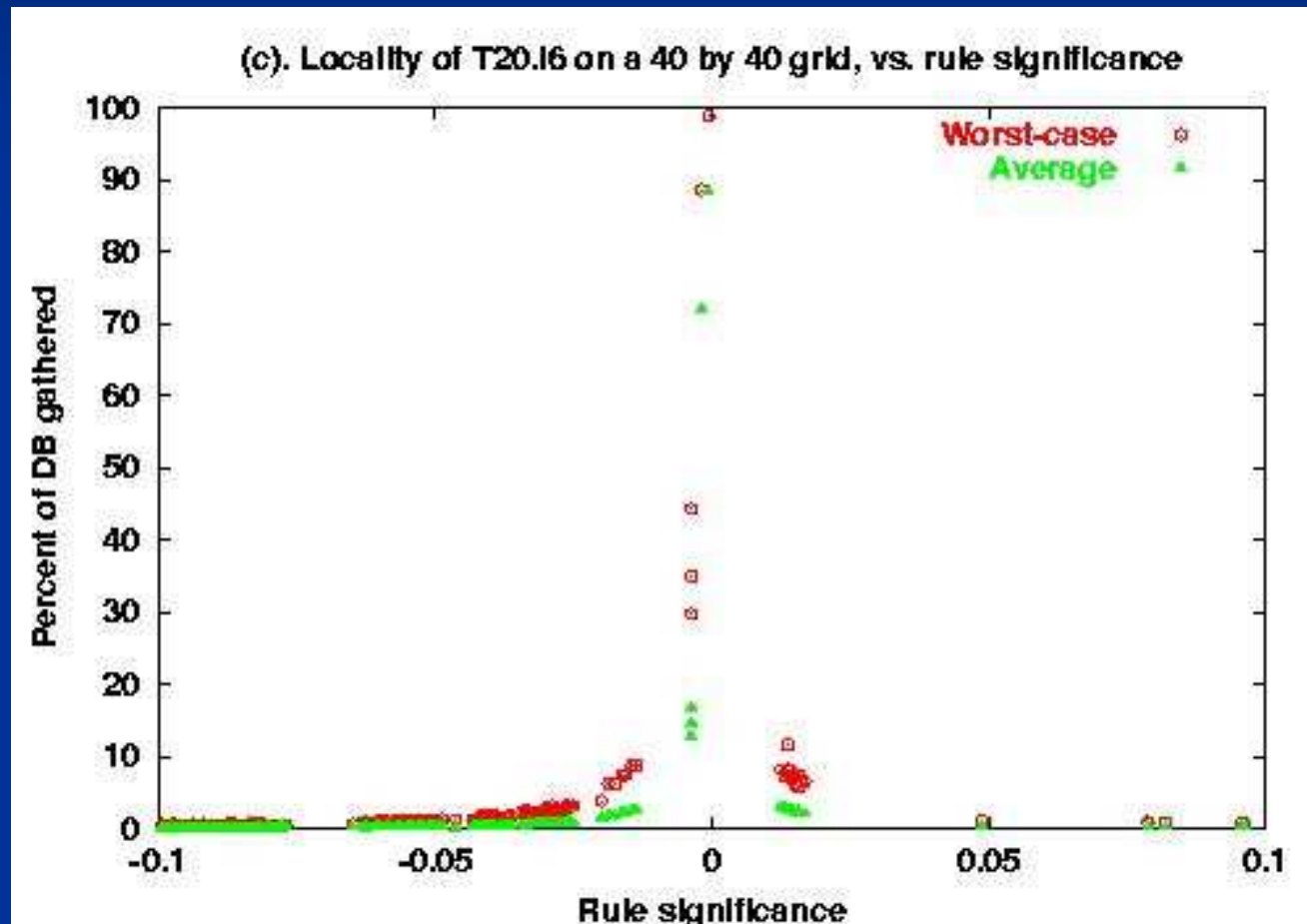
- Information propagation - *correctness*
- No propagation - *locality*



- Rule of thumb: propagate when either
 1. neighbor needs to be persuaded, or
 2. previous message to neighbor turns out to be potentially misleading.



Locality



1,600 nodes

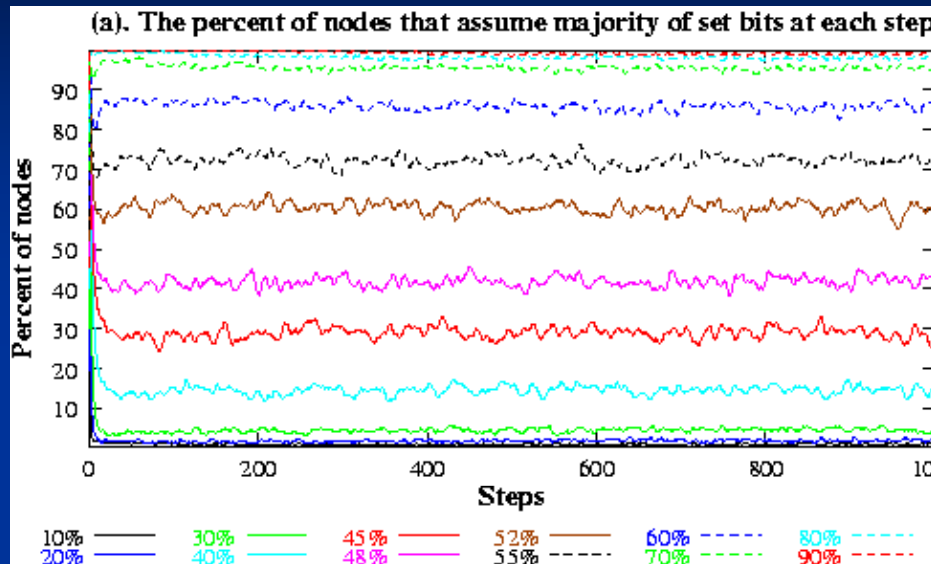
All initiated at once

Local DB of 10K transactions

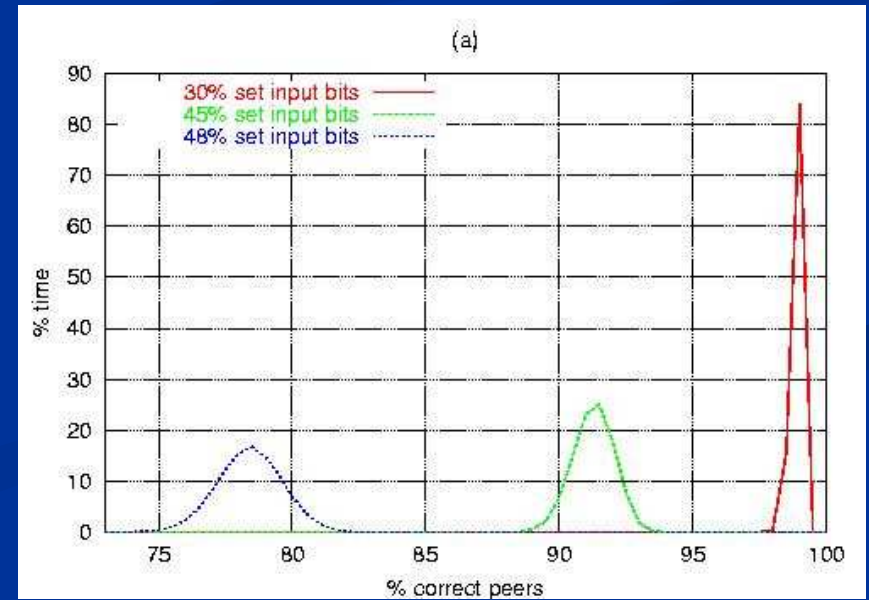
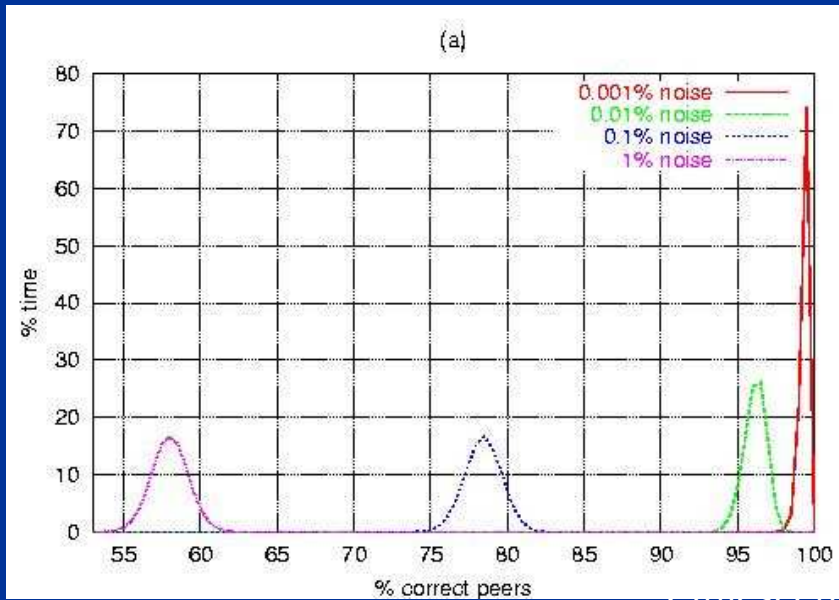
Locked step

Run until there are no further messages

Dynamic Behavior – 1M-peers

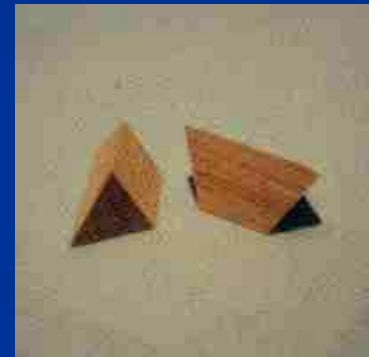


- 1% noise
- At every simulator step



A Decomposition Methodology

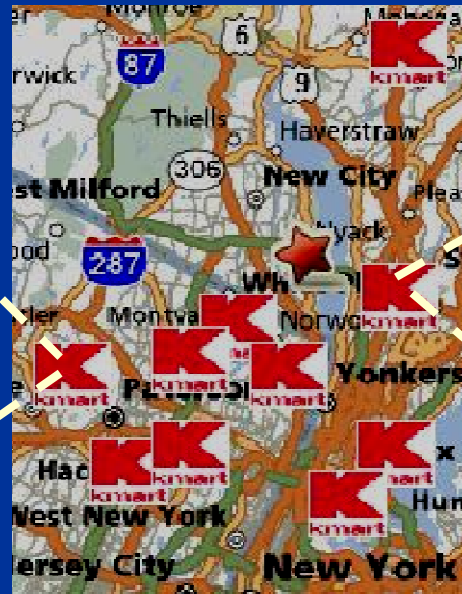
1. Decompose data mining process into primitives
 - Primitives are simpler
2. Find local distributed algorithms for primitives
 - Efficient in the "common case"
3. Recompose data mining process from primitives
 - Maintain correctness
 - Maintain locality
 - Maintain asynchronosity



Distributed Shopping Store

- Each store maintains statistics on last day performances
- Every day the region manager would like to know the *top-k* sold product pairs

<i>Product</i>	<i>Purchases</i>
Milk	100,000
Bread	90,000
Cheese	75,000
Beer	50,000
Diapers	40,000



<i>Product</i>	<i>Purchases</i>
Bread	40,000
Chocolate	35,000
Milk	30,000
Beer	20,000
Diapers	18,000

<i>Product1</i>	<i>Product2</i>	<i>Purchases</i>
Cheese	Milk	85,000
Beer	Diapers	30,000
Beer	Milk	10,000

<i>Product1</i>	<i>Product2</i>	<i>Purchases</i>
Chocolate	Milk	20000
Beer	Diapers	15,000
Beer	Milk	8,000

1/18/2012

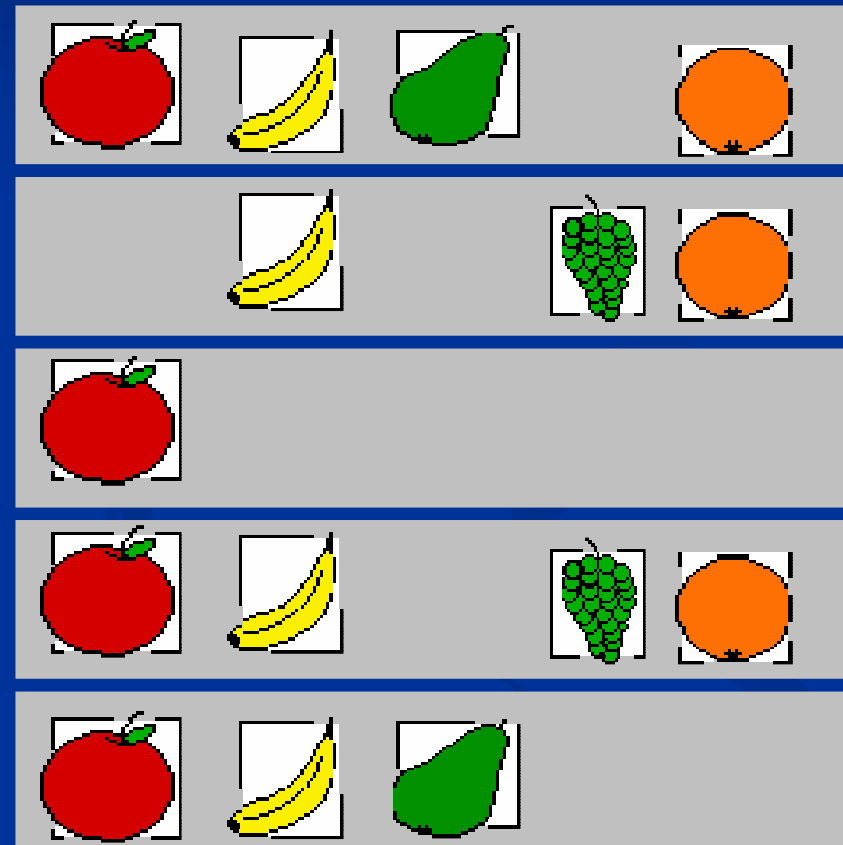
Dozens Kmart stores in NY state

26

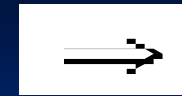
Associations



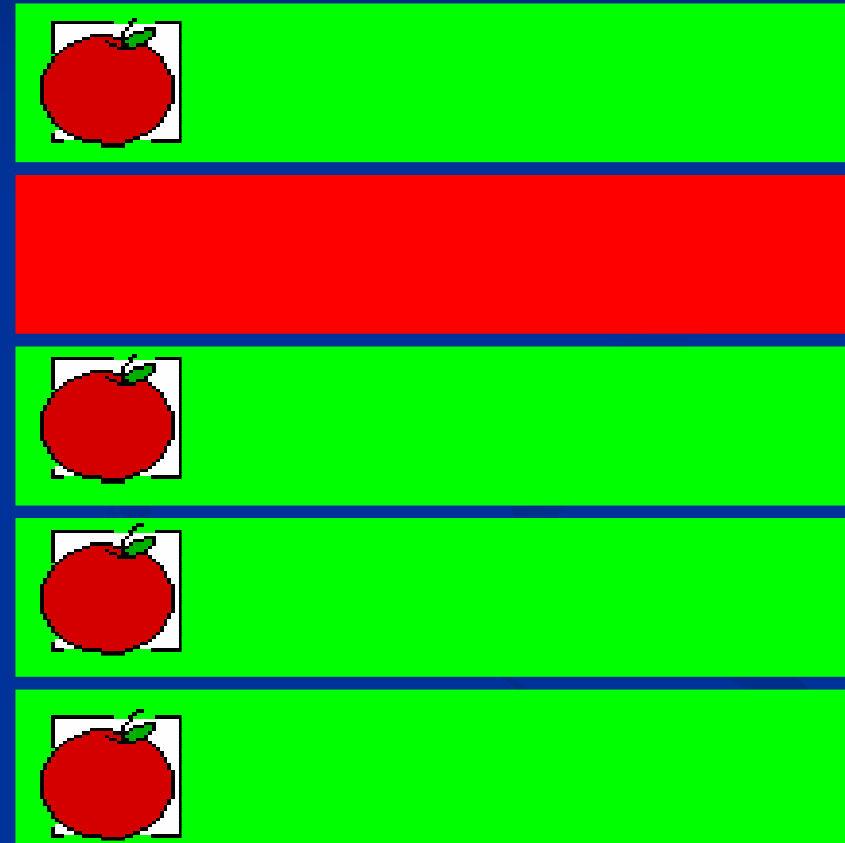
- $X \Rightarrow Y$ is *frequent* if more than **MinFreq** of the transactions contain X and Y
- $X \Rightarrow Y$ is *confident* if more than **MinConf** of the transactions that contain X also contain Y



Associations

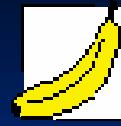


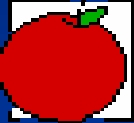
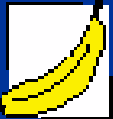
- Find that  is frequent

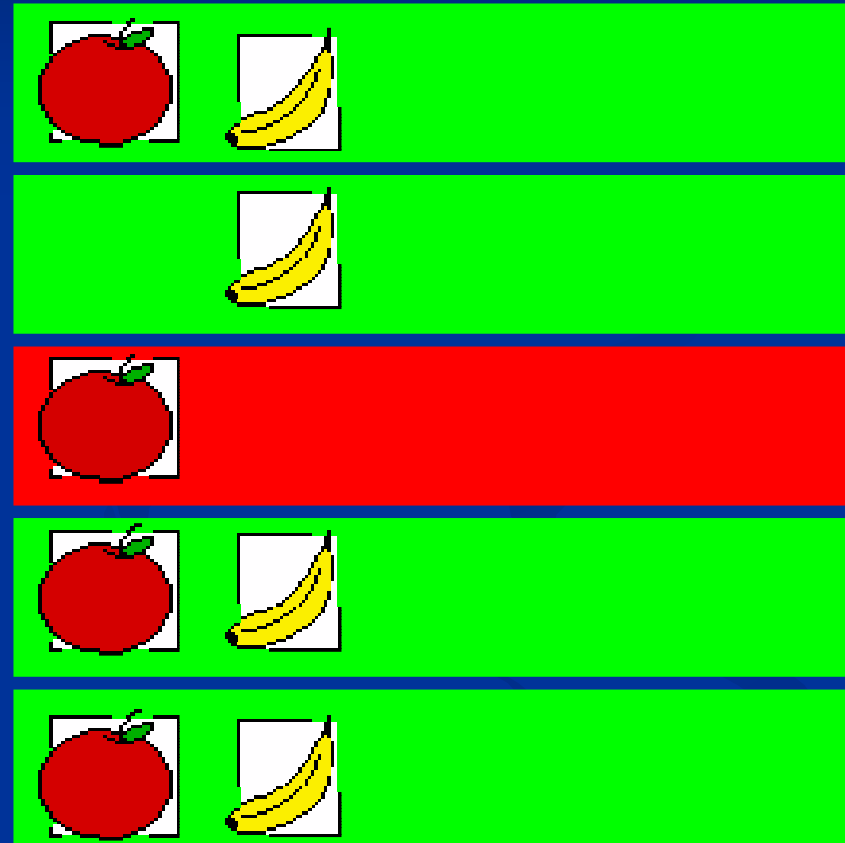


Apples 80%

Associations



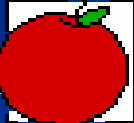
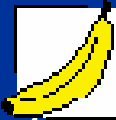
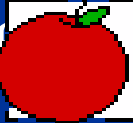
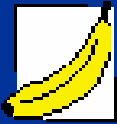
- Find that  is frequent
- And that  is frequent

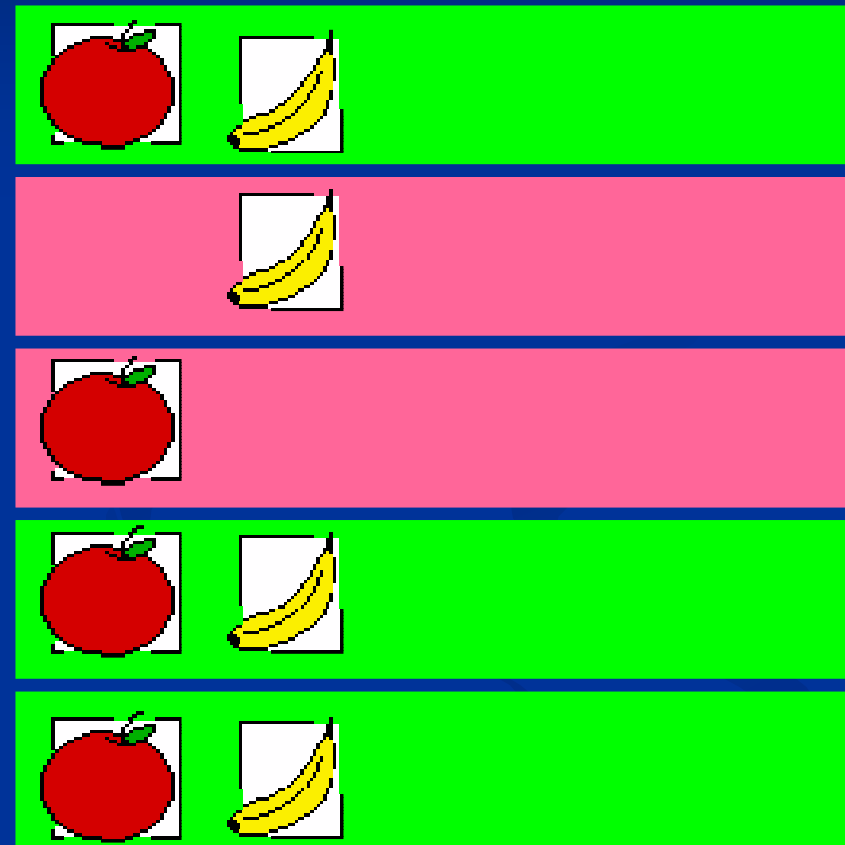


Bananas 80%

Associations

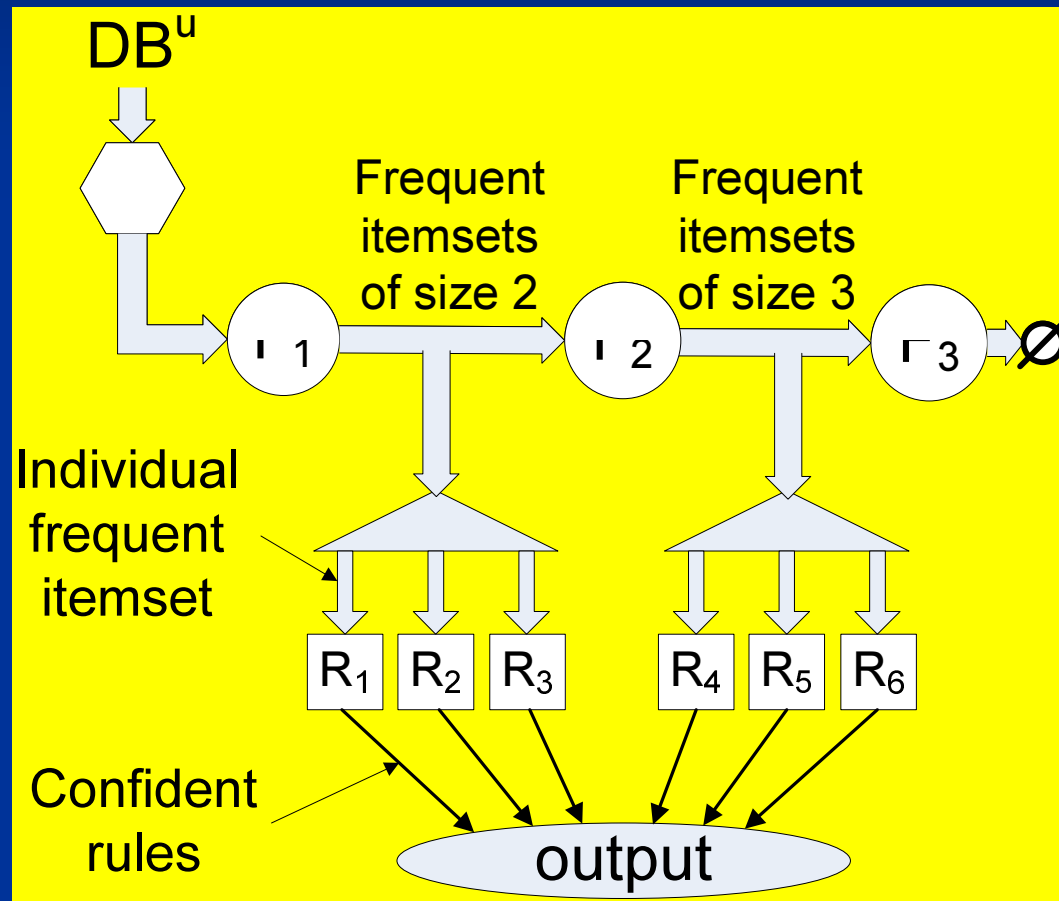


- Find that  is frequent
- And that  is frequent
- Then compare the frequencies of  and 

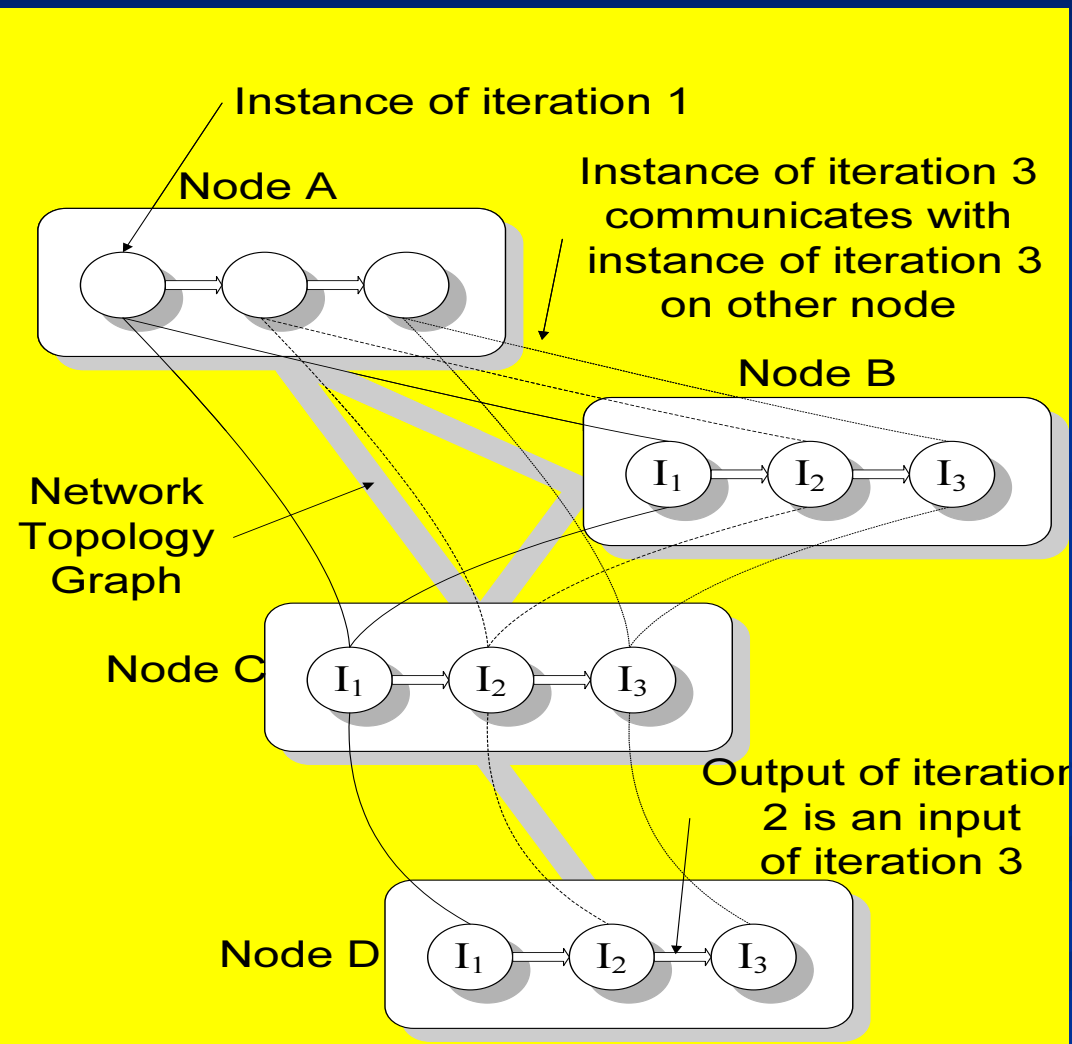


ples  Bananas 75%

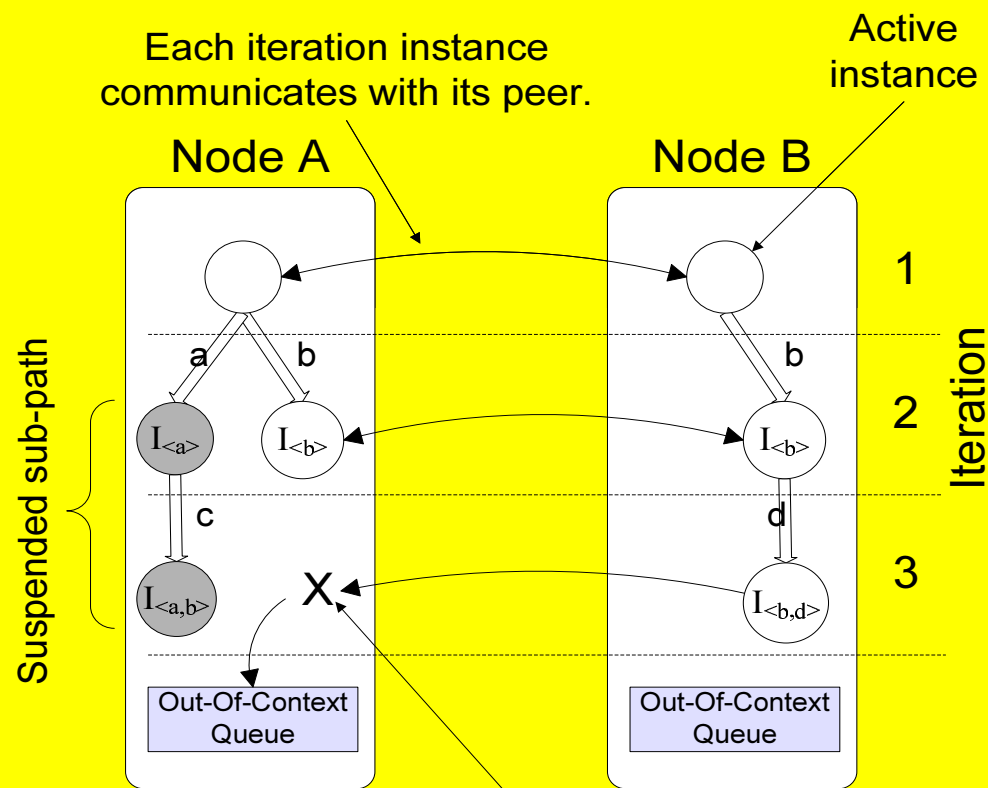
Activation Flow in ARM



Distributed Iterations



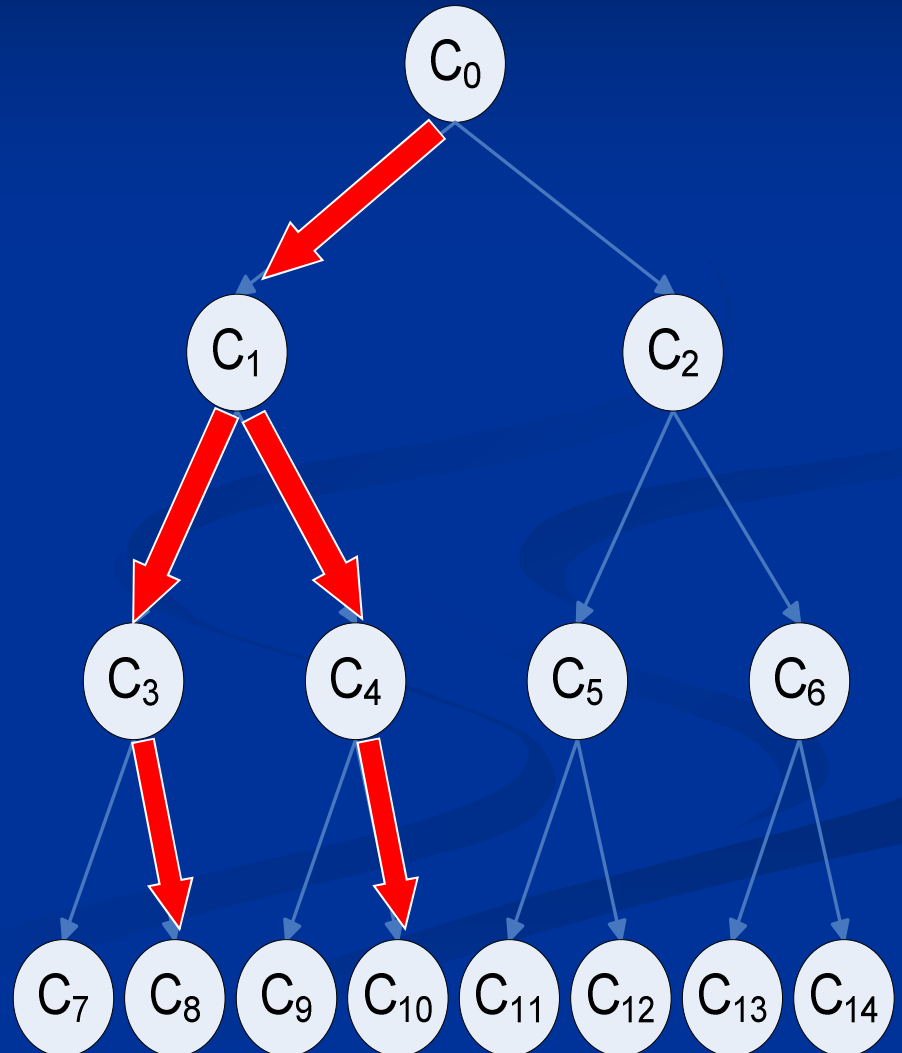
Avoiding Synchronization via Speculation



Instance of iteration 3 in node B does not have a peer in node A, thus its messages go to out-of-context queue.

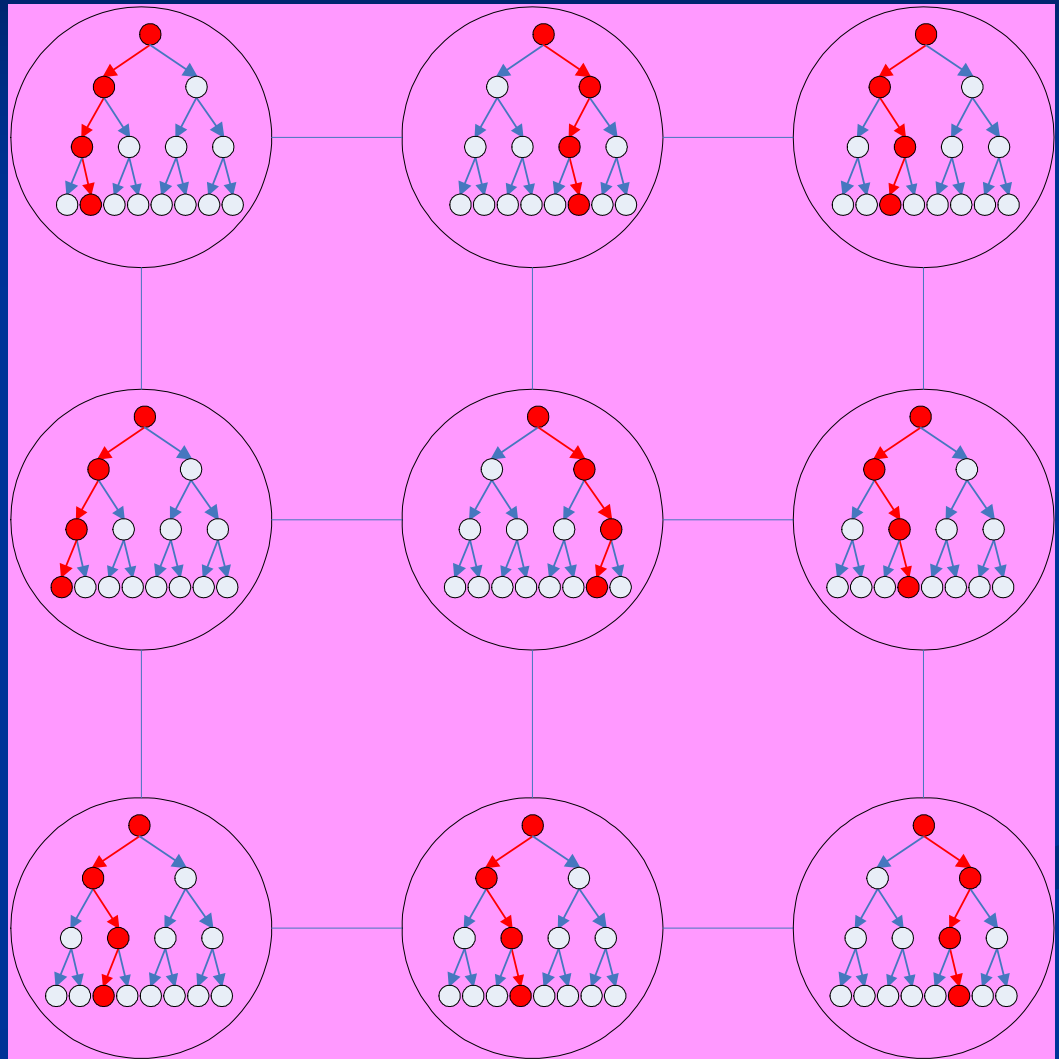
Speculative execution

- If we never finish to compute the first step, how can we start the second?
- We make a guess and base on it the next step.
- The guess is based on the current known data.
- Improves over time.
- If at a certain point the guess turns out wrong, we backtrack and recompute.



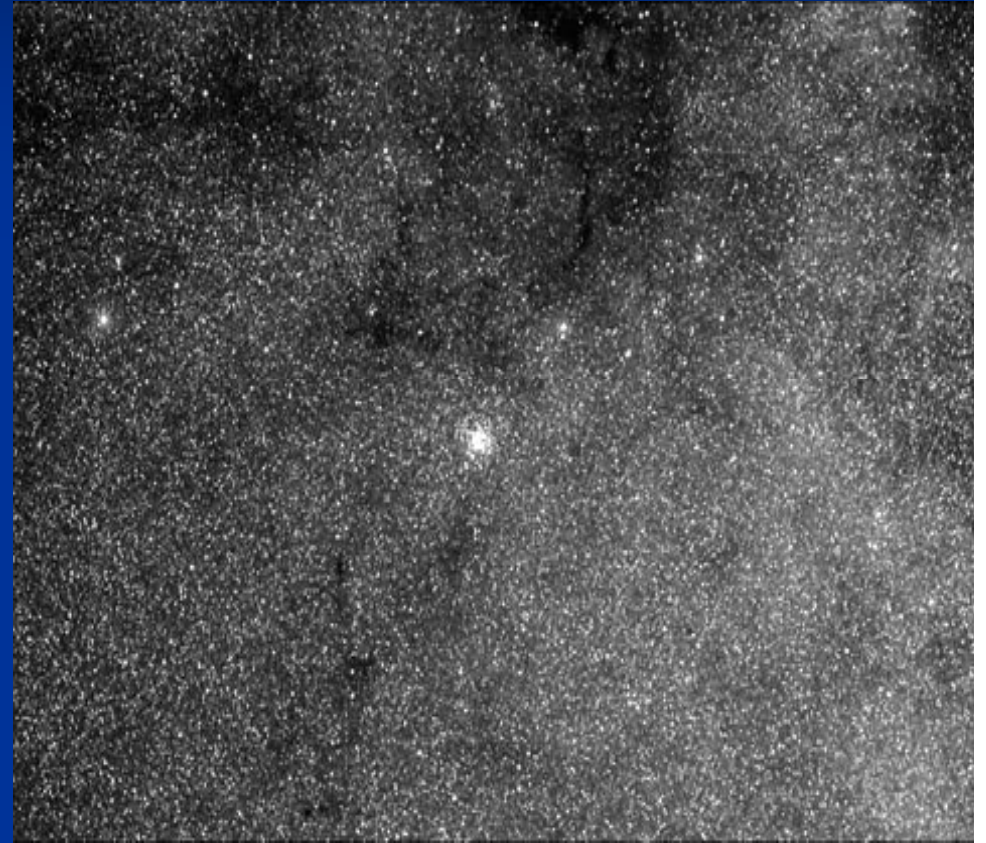
Every node speculates

1. Eventually, the first iteration will converge to the correct result, and will be the same in all nodes.
2. Then, the second iteration will converge, and so on until all iterations are correct.
3. When all iterations are correct, every node will output the correct solution.

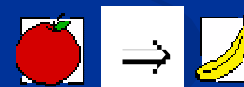
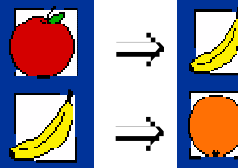
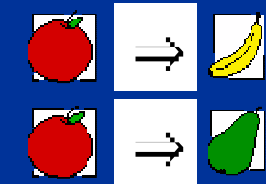
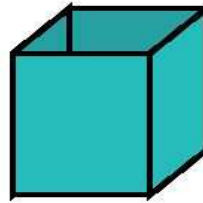
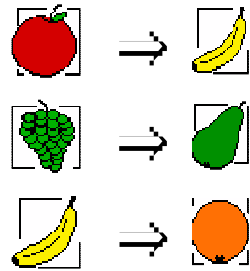
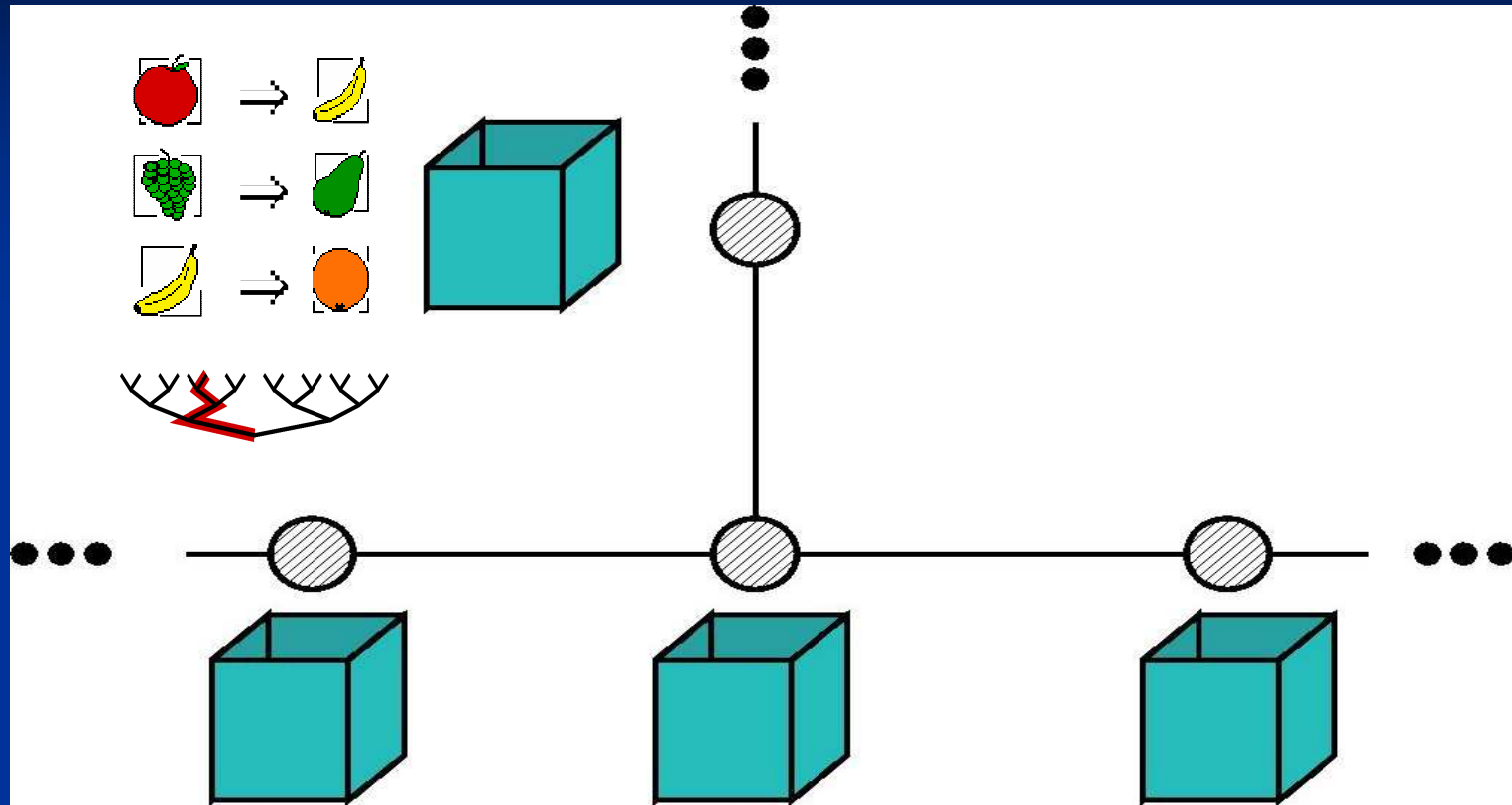


Properties of Speculative Algorithms

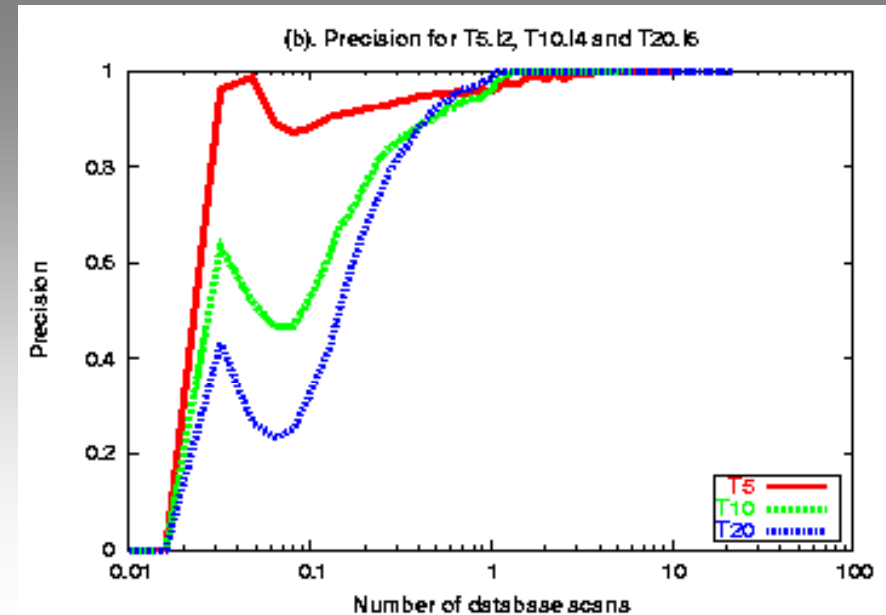
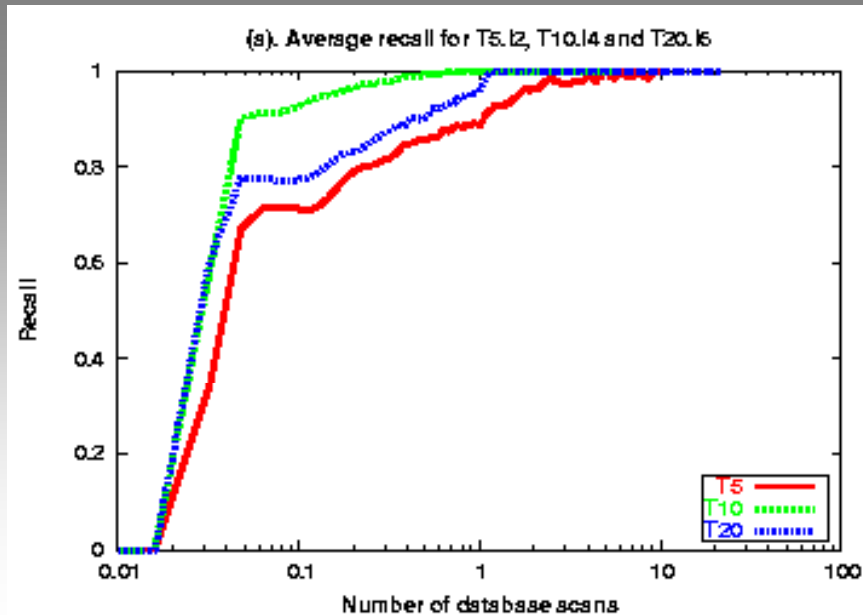
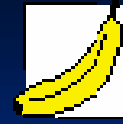
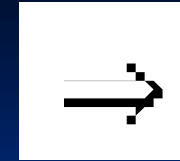
- → Anytime algorithms
- → Output stabilization
- → Incremental ad-hoc view of data
- → Dynamic changes



The Big Picture



Associations Performance

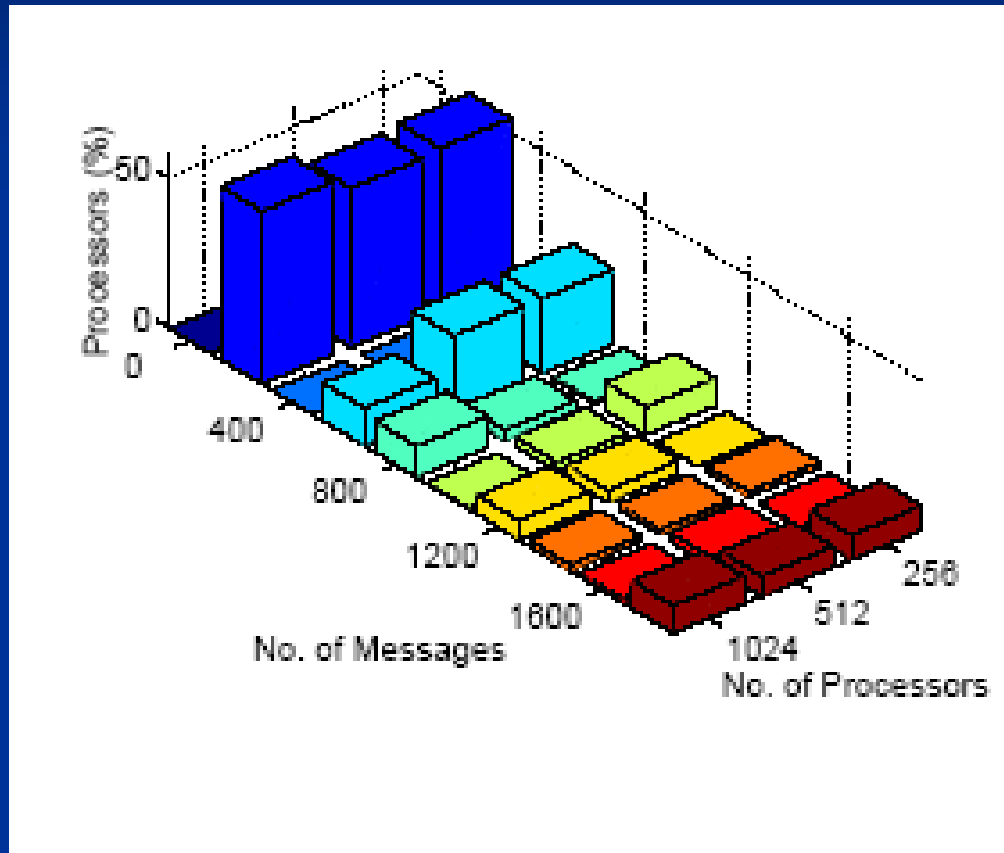


4K peers, Internet-like (Brite), 10K trans./peer, 150 trans read/step.

By the time the database is scanned once, in parallel:

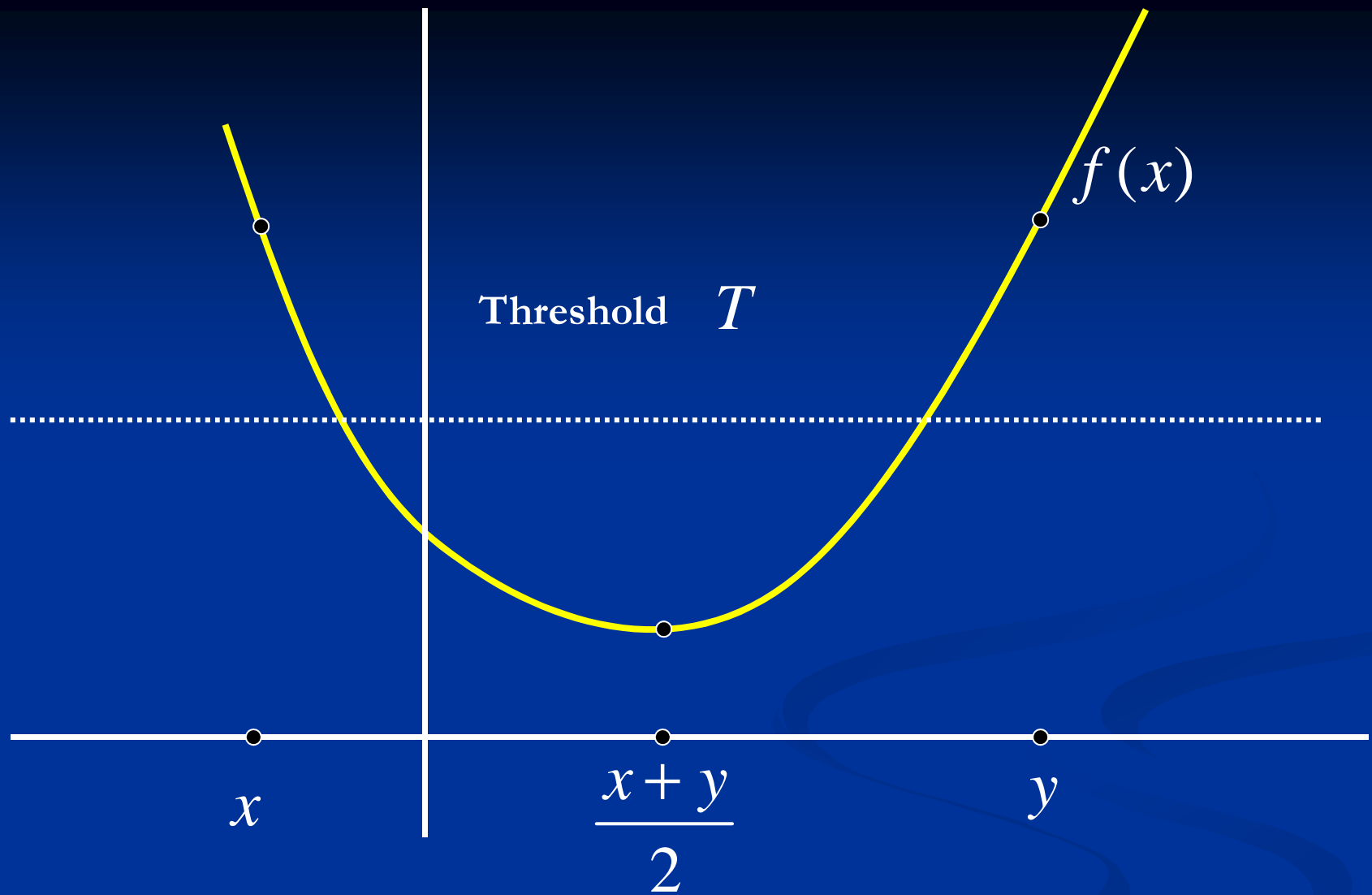
- the average peer has discovered 95% of the rules;
- has less than 10% false rules.

Association Overhead



The Geometric Method

Central coordination



$$f(x) > T, f(y) > T, f\left(\frac{x+y}{2}\right) < T$$

More interesting example: two nodes hold (local) contingency tables, and the global contingency table is the average of the local ones. Given a table π , the mutual information is defined as

$$MI(\pi) = \pi_{11} \log \left[\frac{\pi_{11}}{(\pi_{11} + \pi_{12})(\pi_{11} + \pi_{21})} \right] + \pi_{12} \log \left[\frac{\pi_{12}}{(\pi_{11} + \pi_{12})(\pi_{12} + \pi_{22})} \right] + \\ \pi_{21} \log \left[\frac{\pi_{21}}{(\pi_{21} + \pi_{22})(\pi_{11} + \pi_{21})} \right] + \pi_{22} \log \left[\frac{\pi_{22}}{(\pi_{21} + \pi_{22})(\pi_{12} + \pi_{22})} \right]$$

$$\pi_1 = \begin{pmatrix} 0.9 & 0.03 \\ 0.02 & 0.05 \end{pmatrix}, \quad \pi_2 = \begin{pmatrix} 0.04 & 0.02 \\ 0.03 & 0.91 \end{pmatrix}$$

$$MI(\pi_1) = 0.104, \quad MI(\pi_2) = 0.082, \quad MI\left(\frac{\pi_1 + \pi_2}{2}\right) = 0.494$$

The mutual information of the global table is much larger than the local values. As in the parabola case, there's no way to infer about the global MI given the local ones.

Mining Non-Linear Functions

“ ...

The link function is, of course, **nonlinear**. So we agonize over trading off optimization performance with ability to use the massive infrastructure.

... ”

Sridhar Ramaswamy.

SIGMOD'08 Keynote talk on “Extreme Data Mining”

Slide title: “*10 top reasons why googlers do not sleep at night*”
(Coffee is reason #5)

Summary

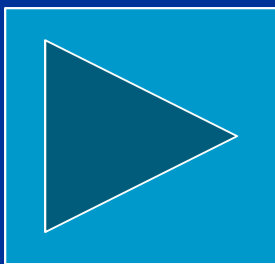
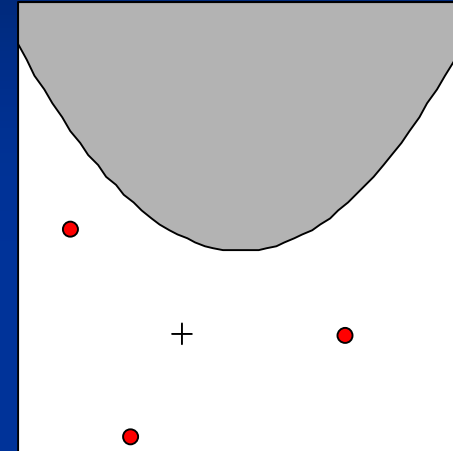
- As in Srikanta's talk... Monitoring is:
- Business as usual: input streams + coordinator
- A global function over the average of the inputs
- A continuous query: does the function cross a given threshold???
- Goal: minimize communication.

Discussion

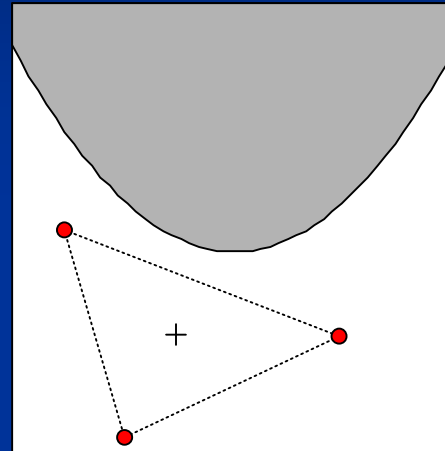
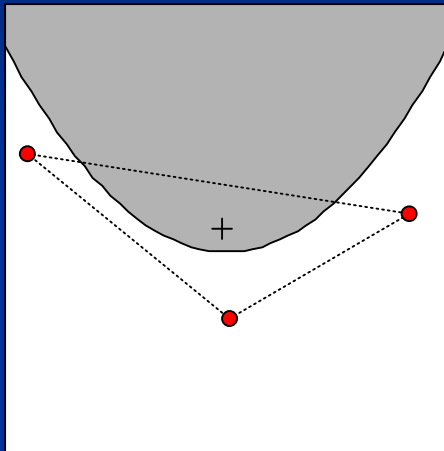
- Looking at the *domain* of the global function may be easier than looking at its value
- Local inputs commonly stationary; rarely: "phase change"
- → Compile LOCAL constraints as indicators to function output getting too close to threshold
 - Not time-based, because of the tradeoff ☹️
 - Not count base, because of real-time ☹️

Geometric Approach

- Geometric Interpretation:
 - Each node holds a statistics vector
 - Coloring the vector space
 - Grey:: $\text{function} > \text{threshold}$
 - White:: $\text{function} \leq \text{threshold}$
- Goal: determine color of global data vector (average).



Bounding the Convex Hull



- Observation: average is in the convex hull
- If convex hull monochromatic then average too
- Problem - convex hull may become large



Drift Vectors

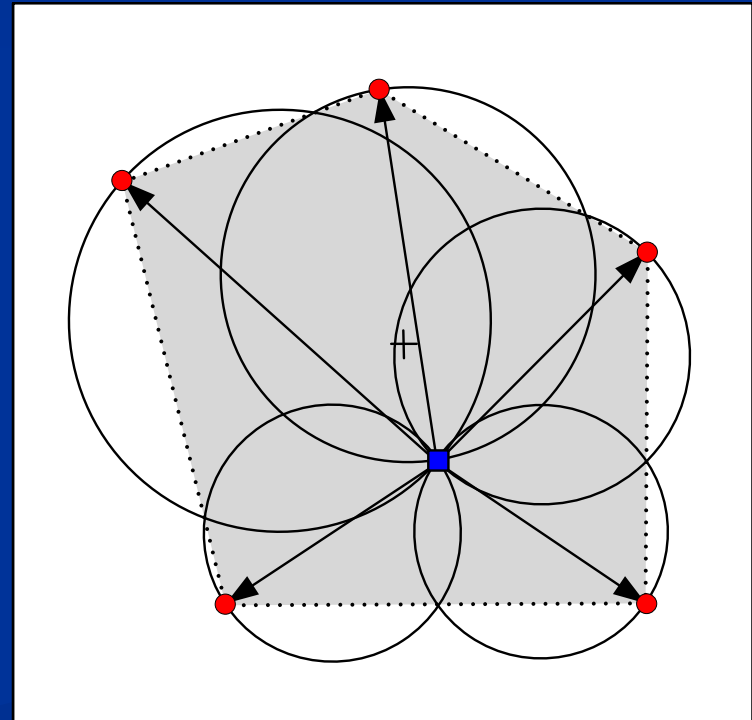
$$\begin{aligned} \text{Avg}(v_i) &= \frac{\sum_{i=1}^n \vec{v}_i}{n} = \frac{\sum_{i=1}^n \vec{v}_i^{\text{known}}}{n} + \frac{\sum_{i=1}^n \Delta \vec{v}_i}{n} = \\ &= \vec{e} + \frac{\sum_{i=1}^n \Delta \vec{v}_i}{n} = \frac{\sum_{i=1}^n (\vec{e} + \Delta \vec{v}_i)}{n} \end{aligned}$$

- Periodically calculate an *estimate vector* - the current global
- Each node maintains a *drift vector* - the change in the local statistics vector since the last time the estimate vector was calculated
- Global average statistics vector is also the average of the drift vectors



The Bounding Theorem

- A reference point is known to all nodes
- Each vertex constructs a sphere
- Theorem: convex hull is bounded by the union of spheres
- → Local constraints!



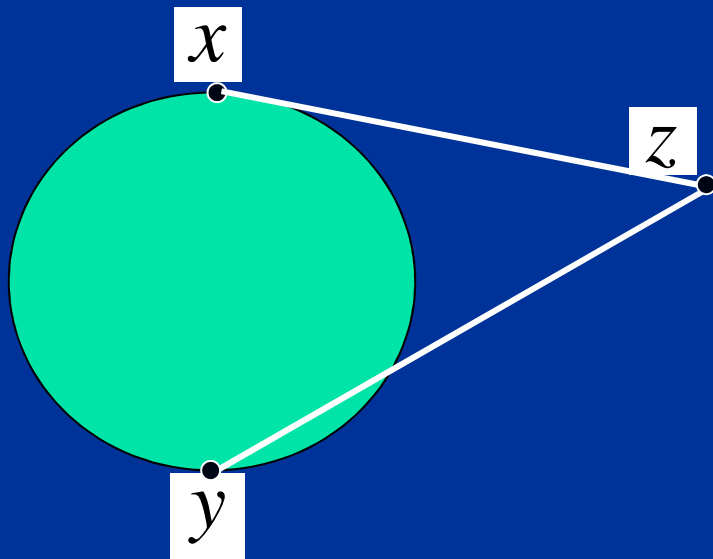
Proofs of the bounding theorem:

SIGMOD06 - induction on the dimension.

Micha Sharir - induction on number of points.

Yuri Rabinovich - uses the following observation:

z is not in the sphere supported by x, y iff $(z-x, z-y) > 0$.



$$z \in \text{conv}(x, y_i) \Rightarrow \lambda x + \sum \lambda_i y_i = z$$

$$\lambda(x-z) + \sum \lambda_i (y_i - z) = 0$$

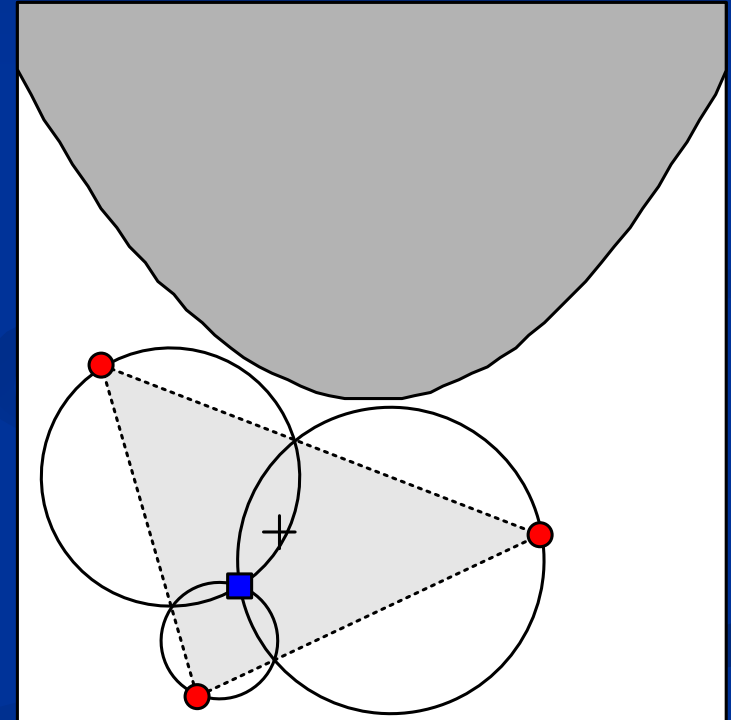
$$\text{if } (z \notin \cup B(x, y_i), \forall i (x-z, y_i - z) > 0$$

$$\text{Hence } (\lambda(x-z), \sum \lambda_i (y_i - z)) > 0$$

$$\text{But } \lambda(x-z) = - \sum \lambda_i (y_i - z), \text{ a contradiction.}$$

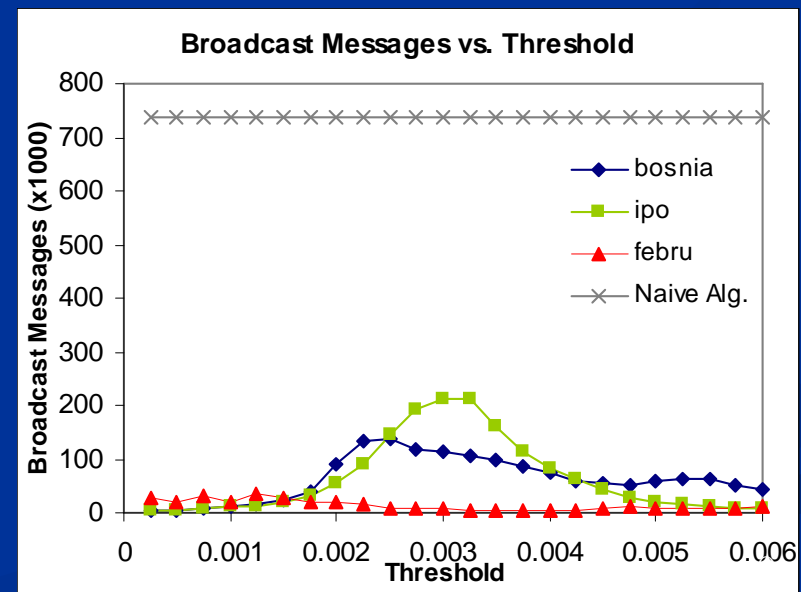
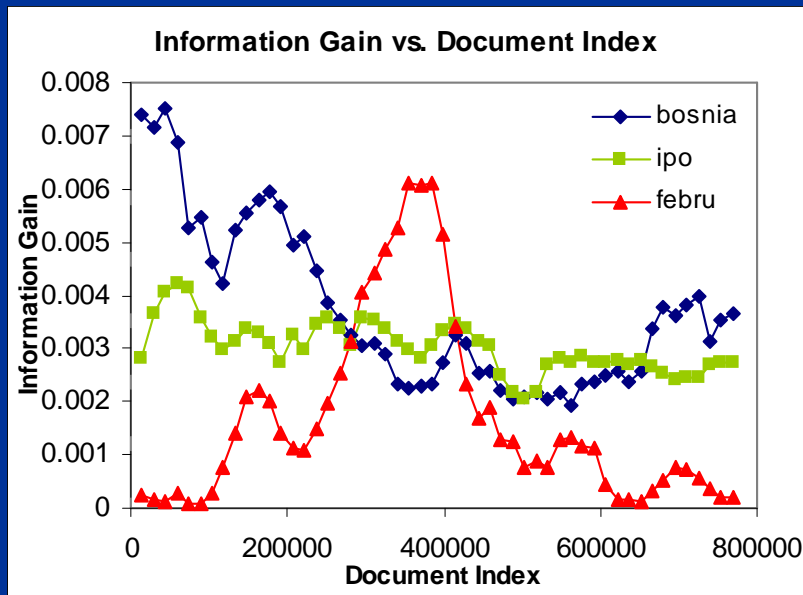
Basic Algorithm

- An initial estimate vector is calculated
- Nodes check color of drift spheres
 - Drift vector is the diameter of the drift sphere
- If any sphere non monochromatic: node triggers re-calculation of estimate vector



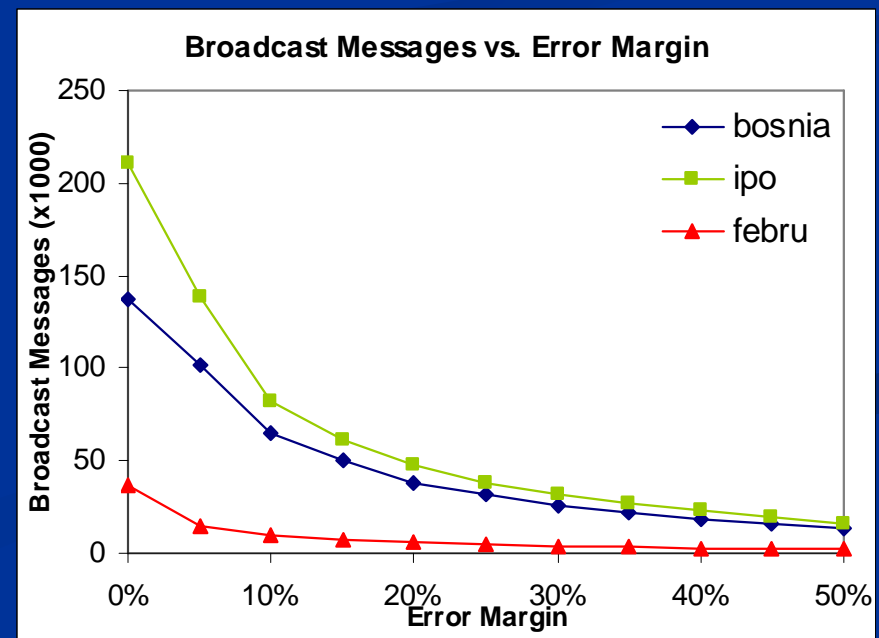
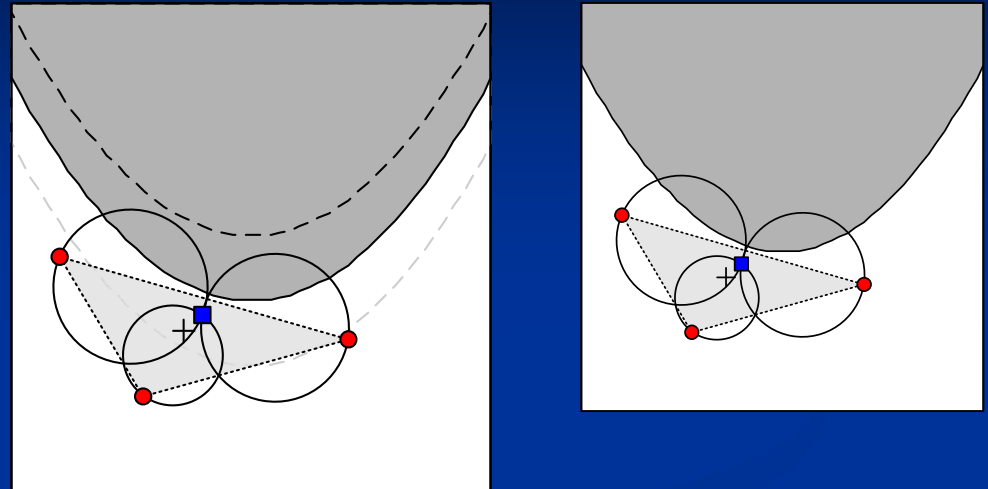
Reuters Corpus (RCV1-v2)

- 800,000+ news stories
- Aug 20 1996 -- Aug 19 1997
- Corporate/Industrial tagging simulates spam

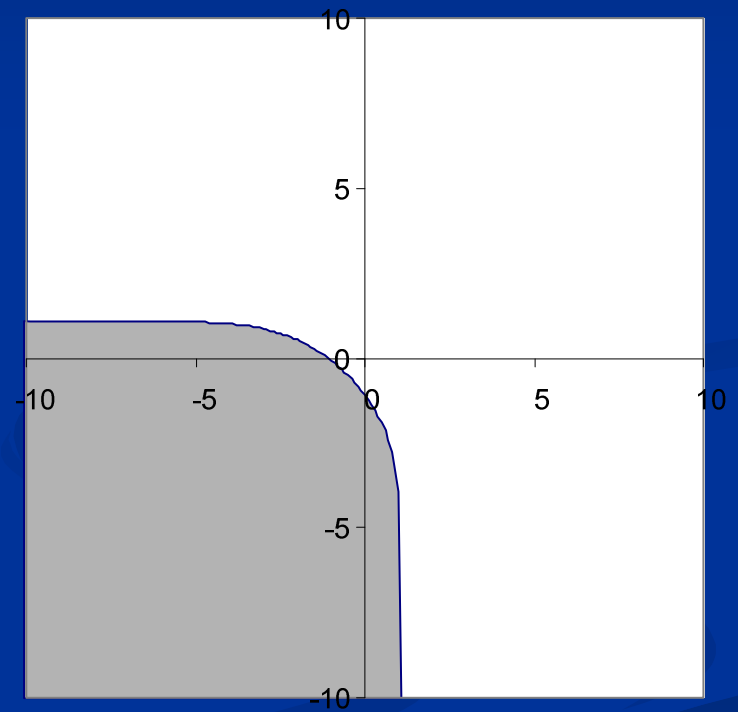
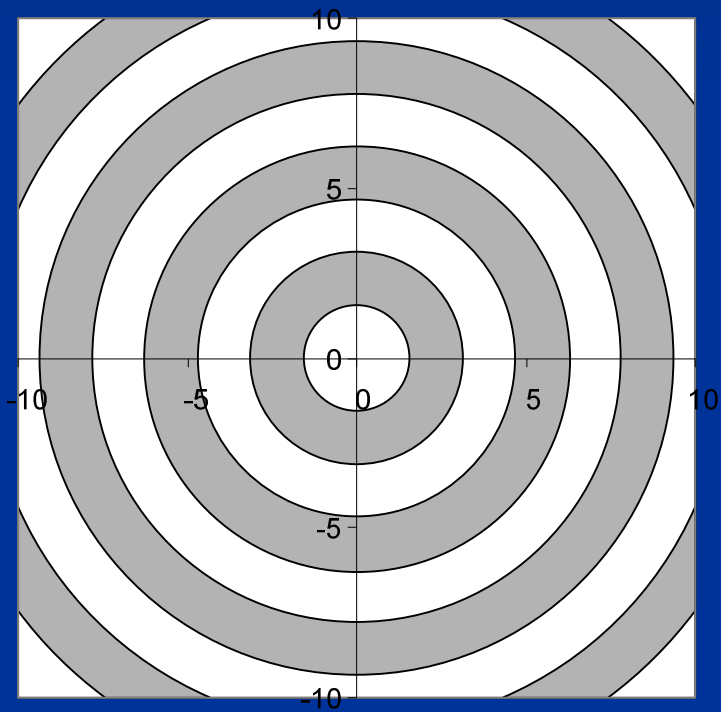


Trade-off: Accuracy vs. Performance

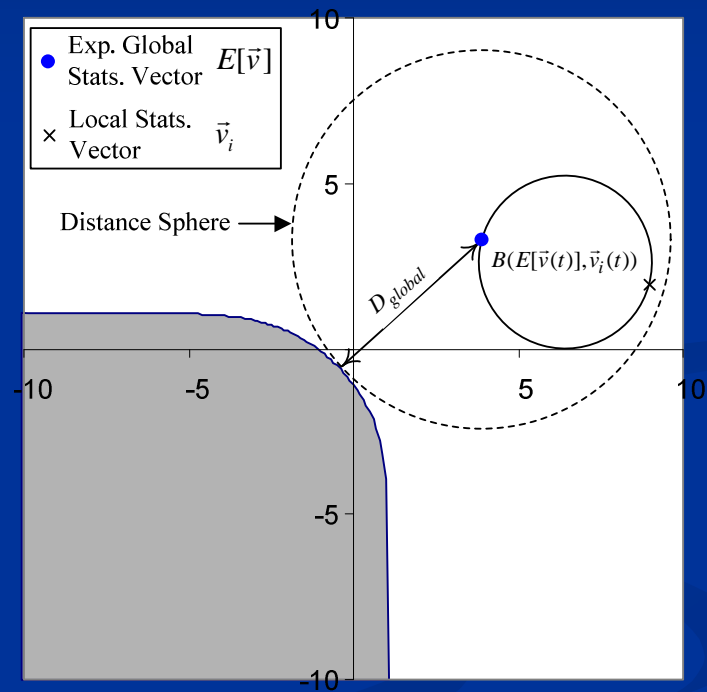
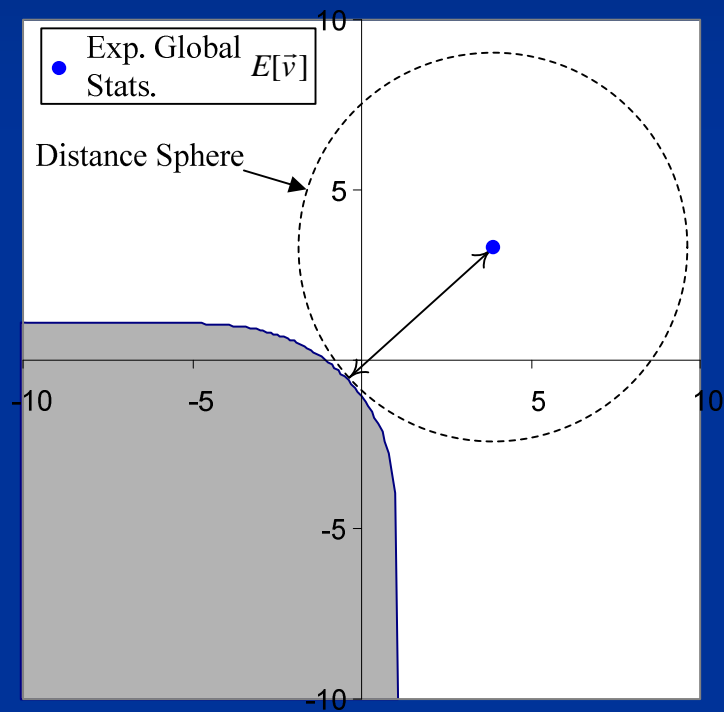
- Inefficiency: value of function on average is close to the threshold
- Performance can be enhanced at the cost of less accurate result:
- Set error margin around the threshold value



Performance Analysis



Performance Analysis (cntd.)



Balancing

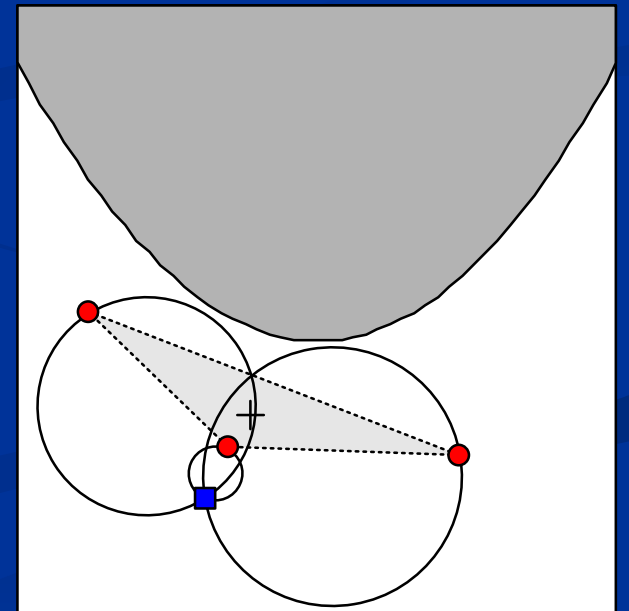
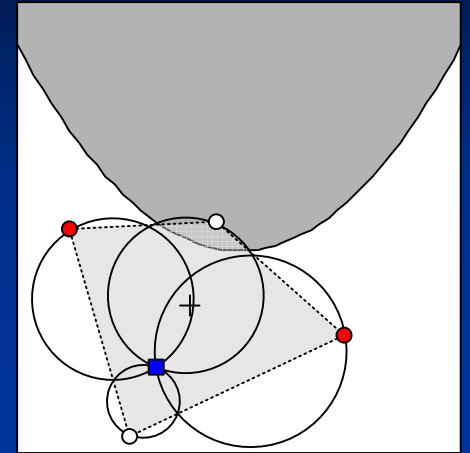
- Globally calculating average is costly
- Often possible to average only *some* of the data vectors.

$$\text{Avg}(v_i) = \frac{\sum_{i=1}^n (\vec{e} + \Delta\vec{v}_i)}{n}$$

$$\sum_1^n \vec{\delta}_i = 0$$

\Leftrightarrow

$$\text{Avg}(v_i) = \frac{\sum_{i=1}^n (\vec{e} + \Delta\vec{v}_i + \vec{\delta}_i)}{n}$$

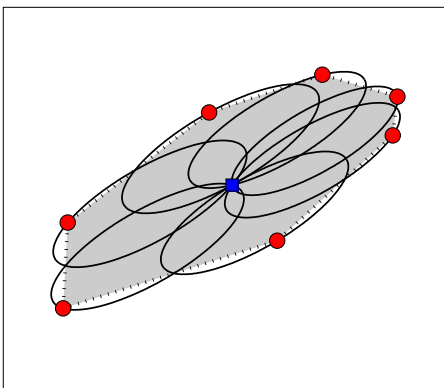
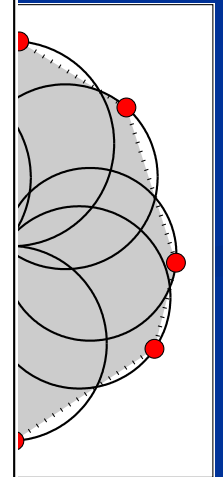
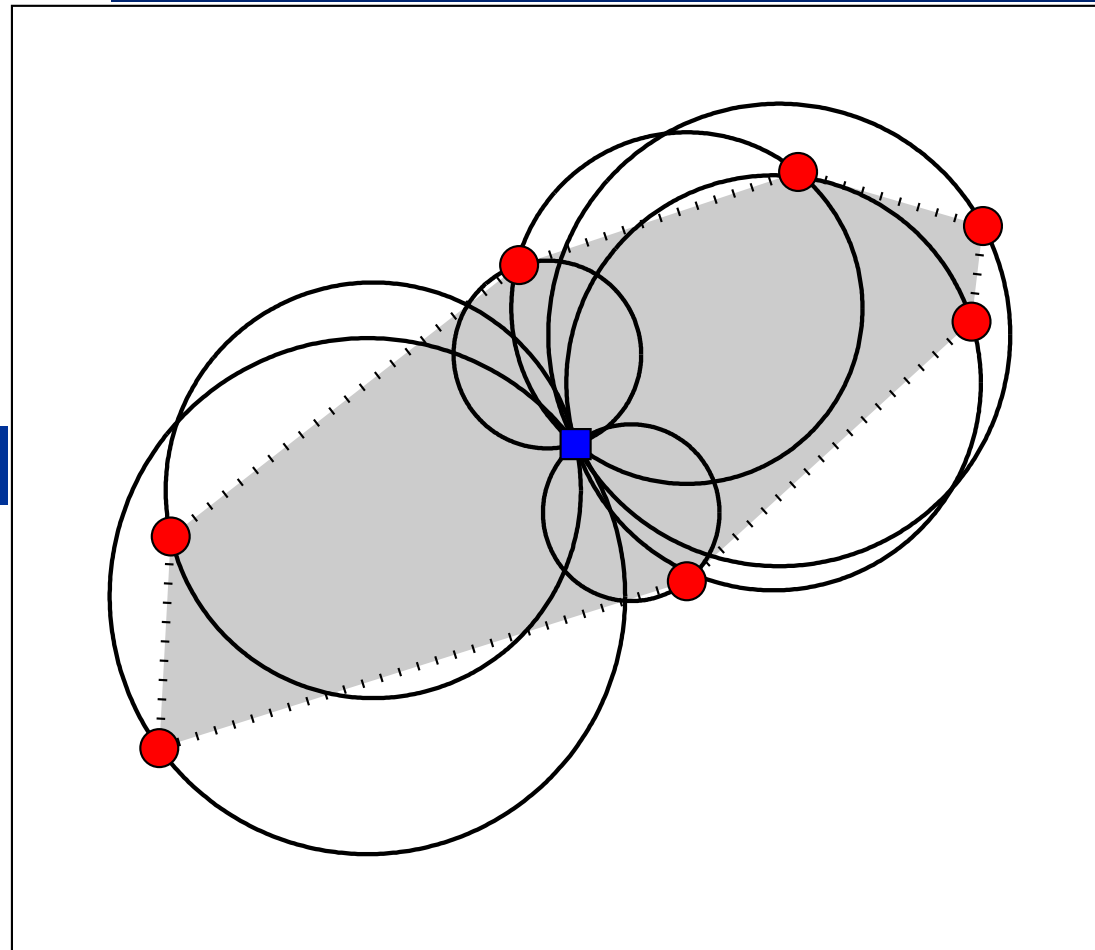
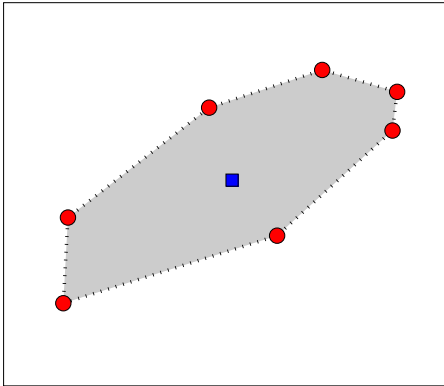


Shape Sensitivity

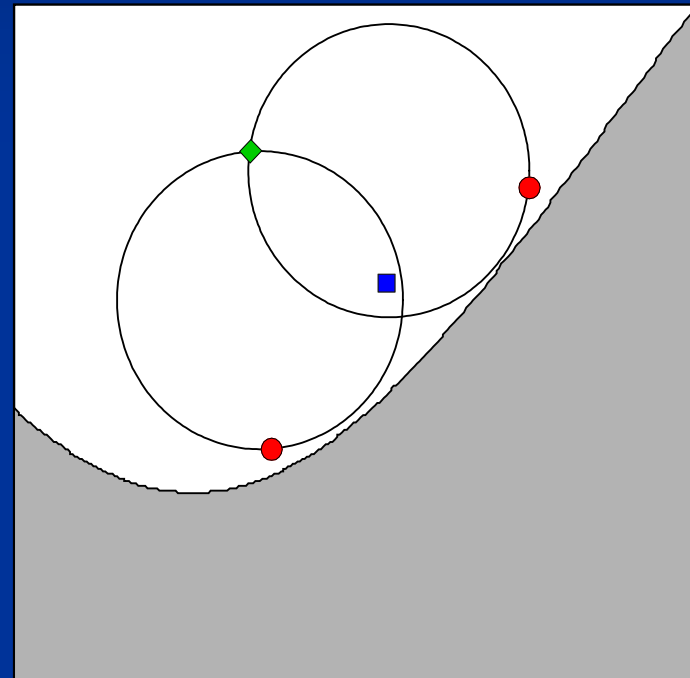
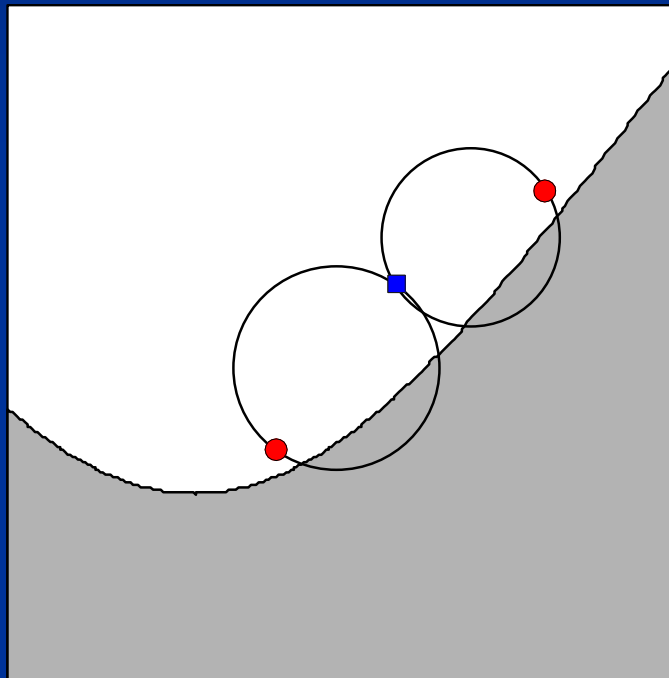
- Fitting cover to Data
- Fitting cover to threshold surface
- Specific function classes

Fitting Cover to Data

(using the covariance matrix)

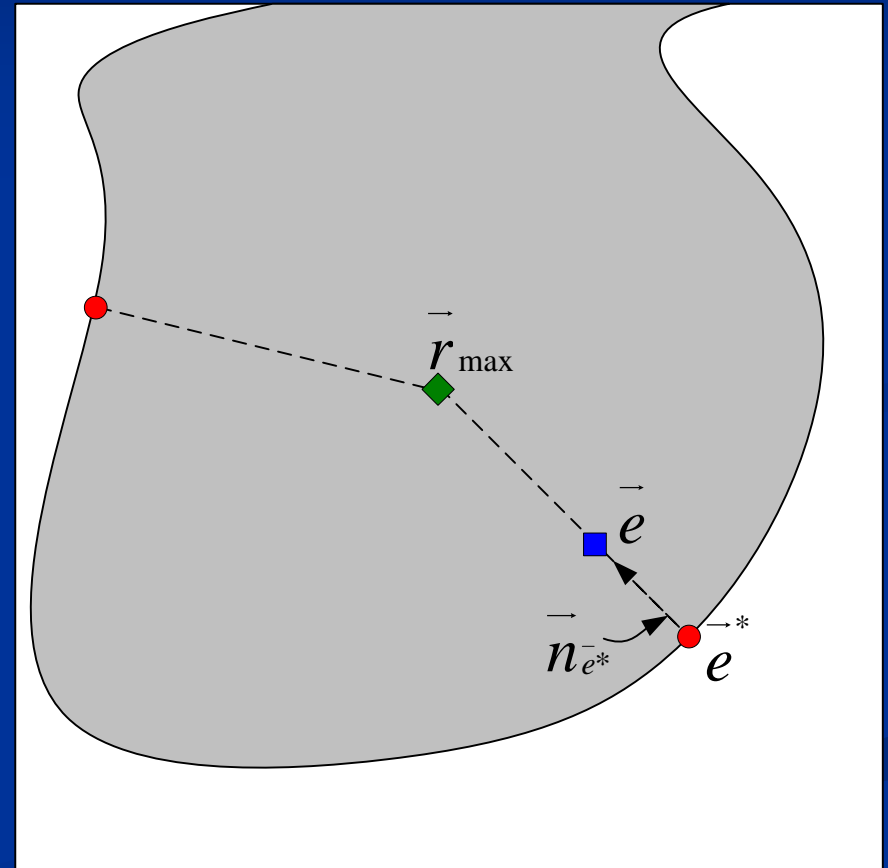
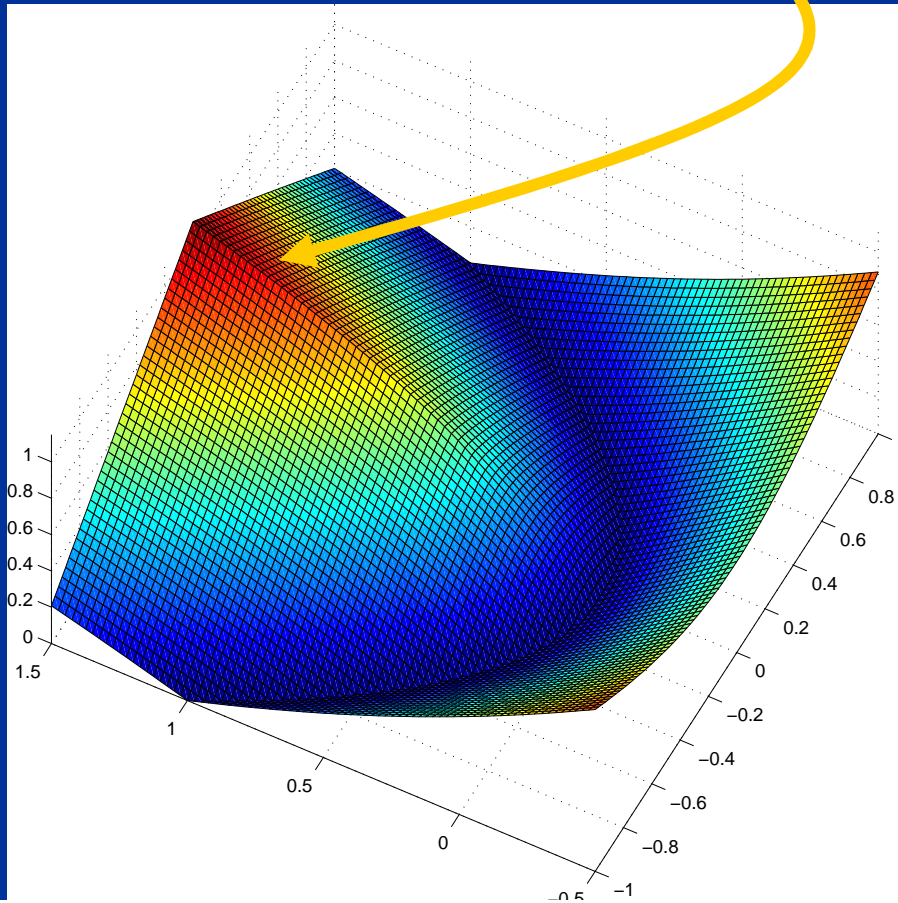


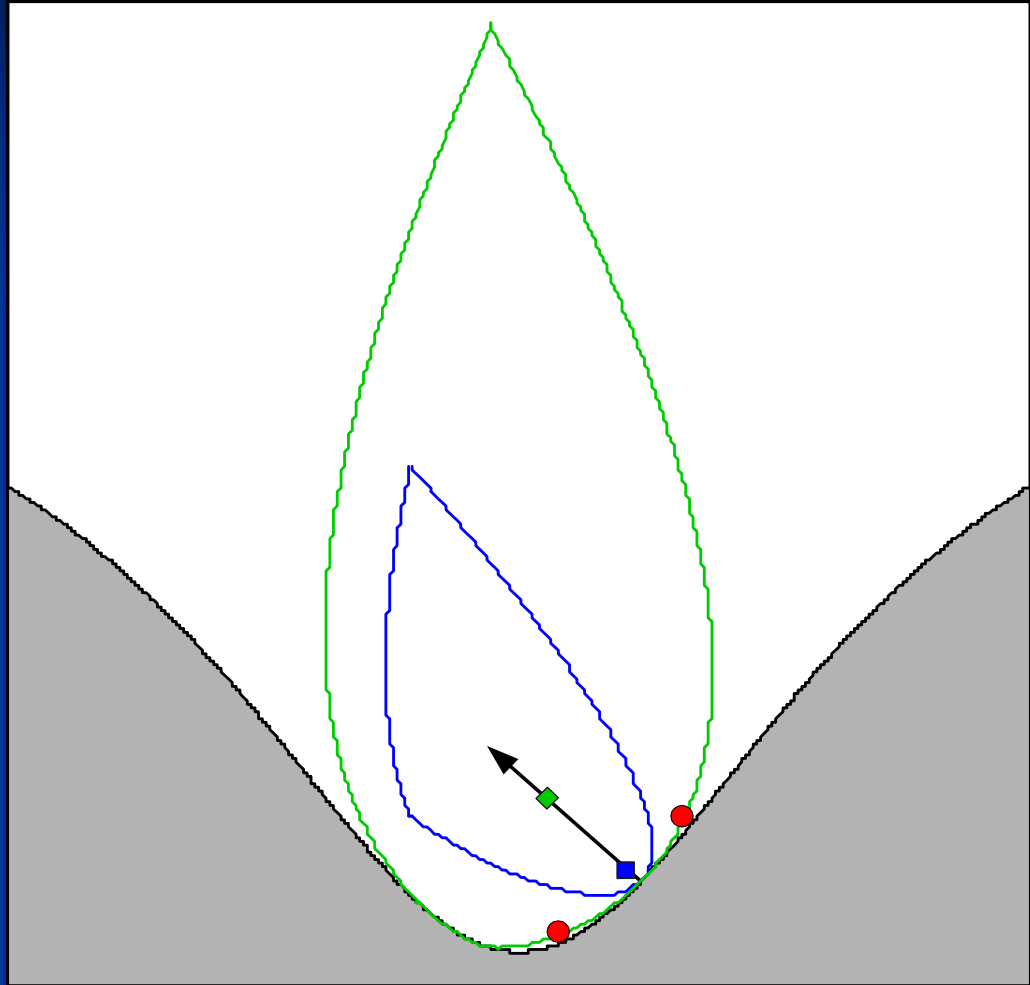
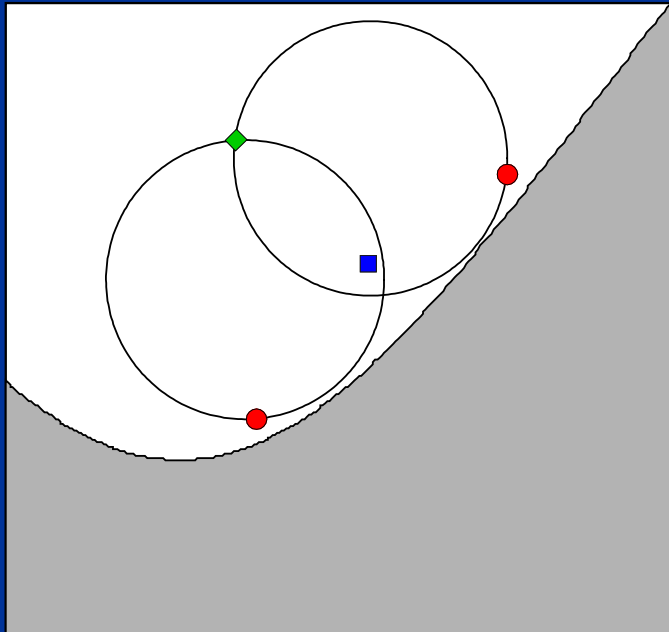
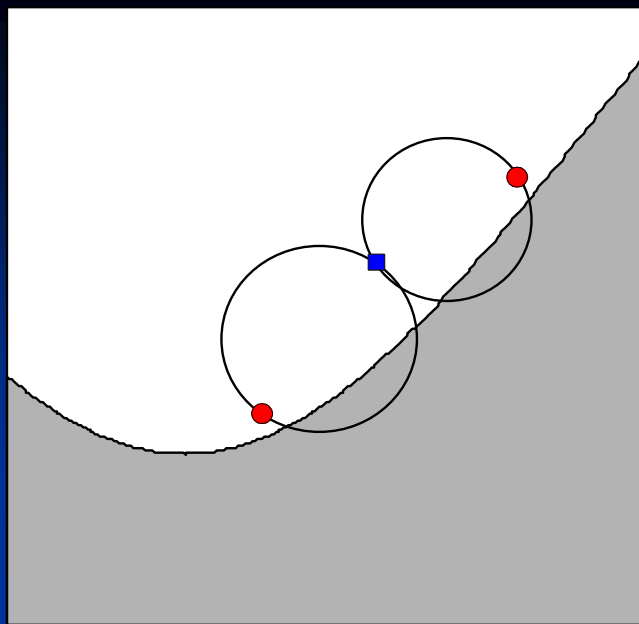
Fitting Cover to Threshold Surface -- Reference Vector Selection



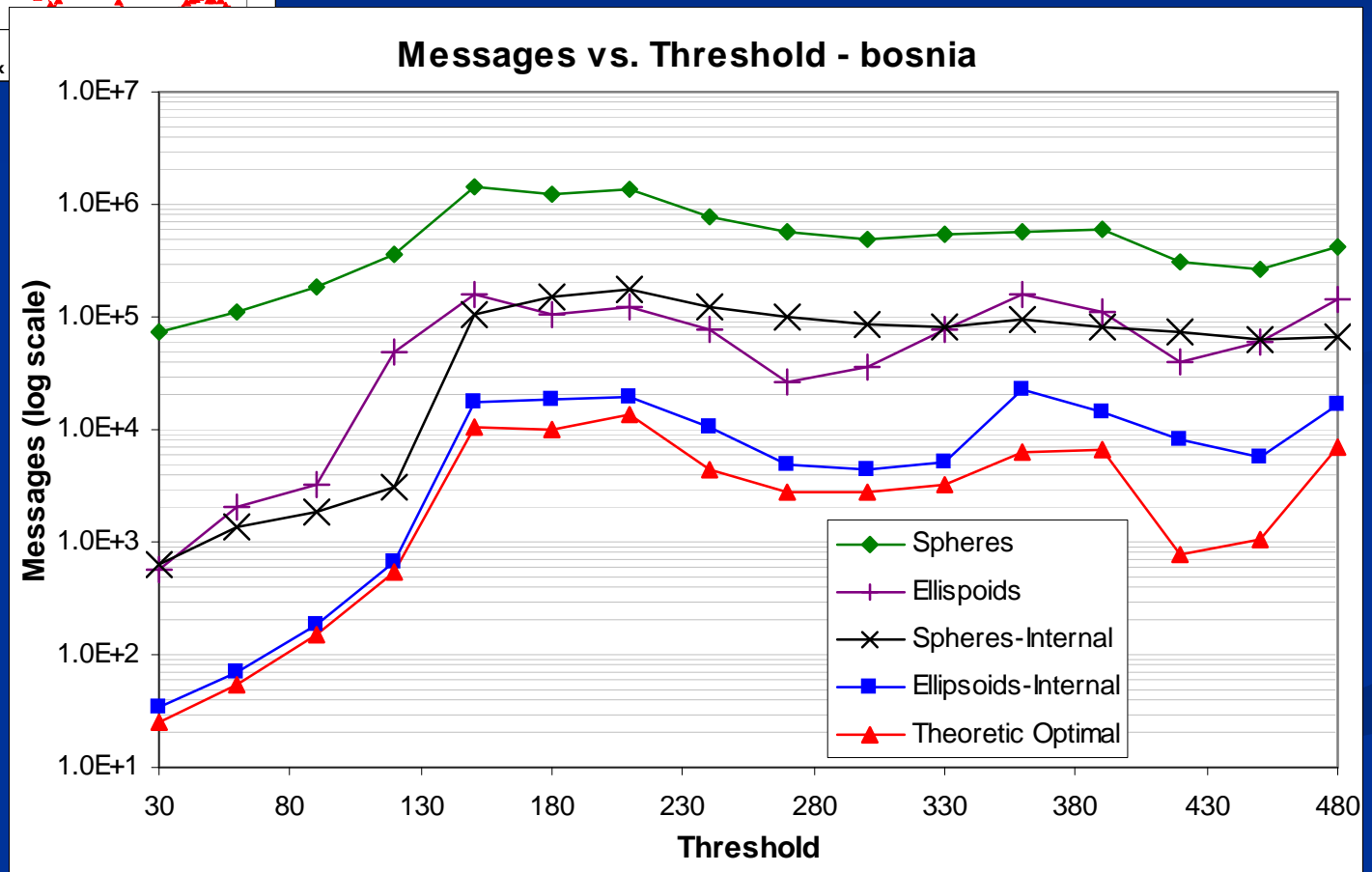
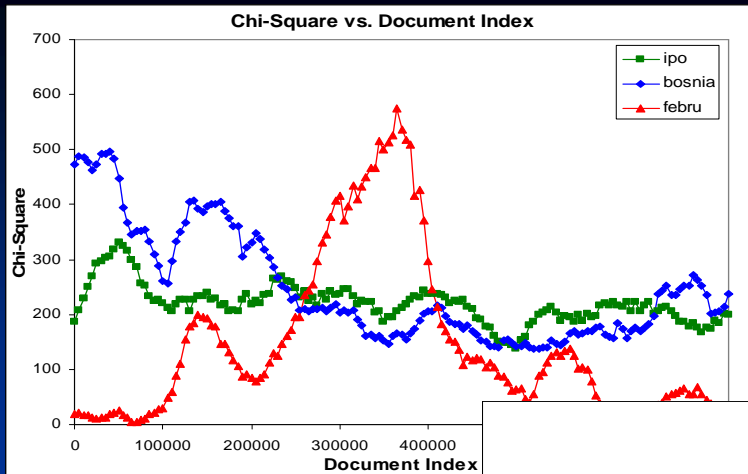
Distance Fields

Skeleton, Medial Axis





Results – Shape Sensitivity



Distributed Top-K

Distributed Search Engine Example

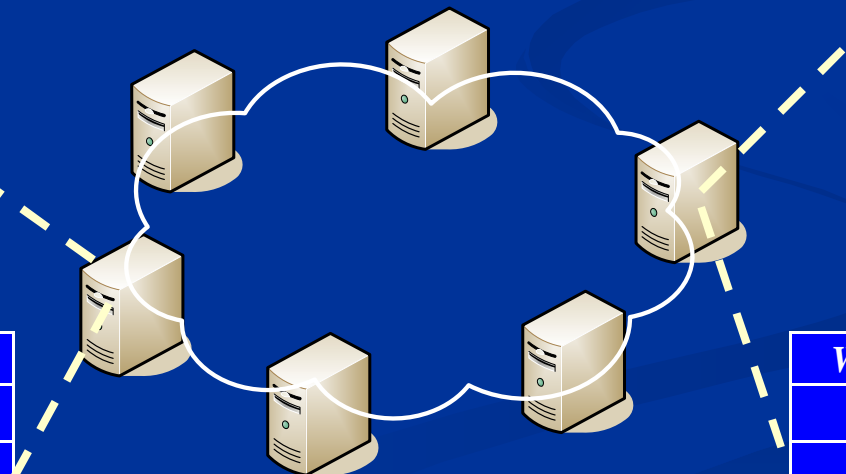
- Each node maintains local statistics on last day queries
- Every day the search engine manager would like to mine the logs for the top-k appearing keyword pairs

<i>Word</i>	<i>Count</i>
Total	1000
DVD	850
Sell	800
Insurance	620
Price	500

<i>Word</i>	<i>Count</i>
Total	5000
Sell	3500
Price	2000
Day	1750
Election	1200

<i>Word1</i>	<i>Word2</i>	<i>Count</i>
Total	Sell	640
DVD	Price	500
Insurance	Price	450

<i>Word1</i>	<i>Word2</i>	<i>Count</i>
Total	Sell	2900
Day	Election	1000
Price	Total	900



Correlation Coefficient Function

$$\rho_{AB} = \frac{f_{AB} - f_A f_B}{\sqrt{(f_A - f_A^2)(f_B - f_B^2)}}$$

- Value in the range [-1,1]
- >0 - indicates that the terms tend to appear in the same queries
- =0 - indicates that there is no correlation between terms
- <0 - indicates that the terms tend to exclude each other
- Restrict to $f_{AB} \leq \min(f_A, f_B)$
- Monotonically decrease with f_A, f_B
- Monotonically increase with f_{AB}

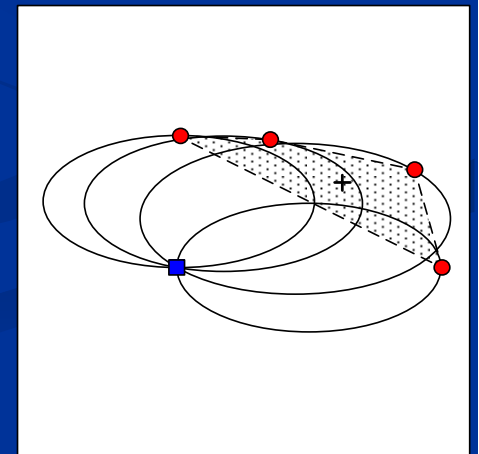
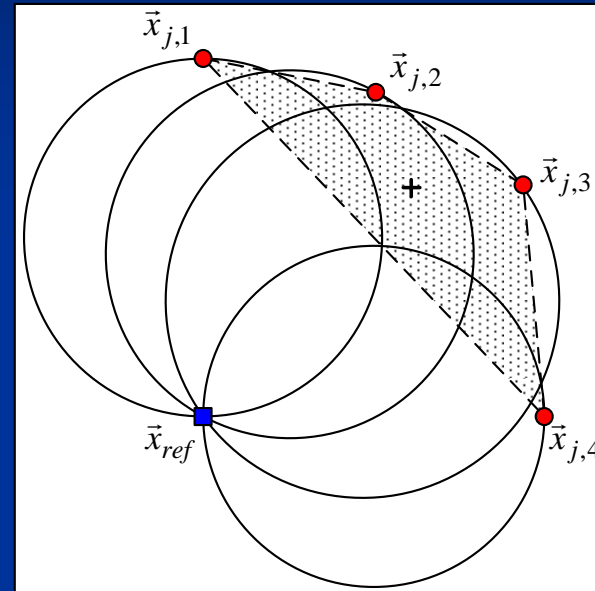
	Queries	A	B	A & B	$f_{A,i}$	$f_{B,i}$	$f_{AB,i}$	$\rho_{AB,i}$
Node1	1000	100	100	19	0.1	0.1	0.019	0.1
Node2	1000	400	400	184	0.4	0.4	0.184	0.1
					f_A	f_B	f_{AB}	ρ_{AB}
Global 1/18/2012	2000	500	500	203	0.25	0.25	0.1015	0.208

A Four-Phase Approach

- Phase I - Determine a lower bound on the score of the top- k objects
 - By collecting local contenders
- Phase II - Improve the lower bound
 - By collecting all local contenders
- Phase III - Determine candidates by locally prune out objects according to lower bound
 - Use geometry to derive local constraints and local upper bounds
 - Use domination relations to avoid access whole DB
- Phase IV - Determine top- k objects among candidates

Determining Local Upper Bounds

- The local upper bound $u_{j,i}$ at the node p_i for the object o_j is determined as follows:
 - The local statistics vector for each object creates a sphere whose diameter is distance to the origin
 - $u_{j,i}$ is the maximum score received within the sphere



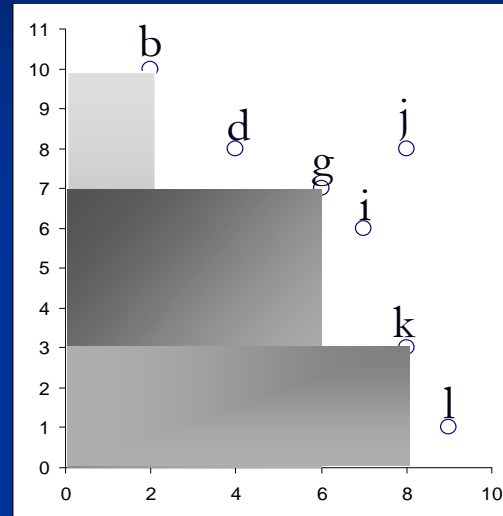
Reducing I/O Costs - Domination Relations

- If \vec{x} dominates \vec{y} , then for any monotone function f

$$f(\vec{x}) \geq f(\vec{y})$$

- Theorem: Let $\vec{x}_{a,i}$ dominate $\vec{x}_{b,i}$. Then

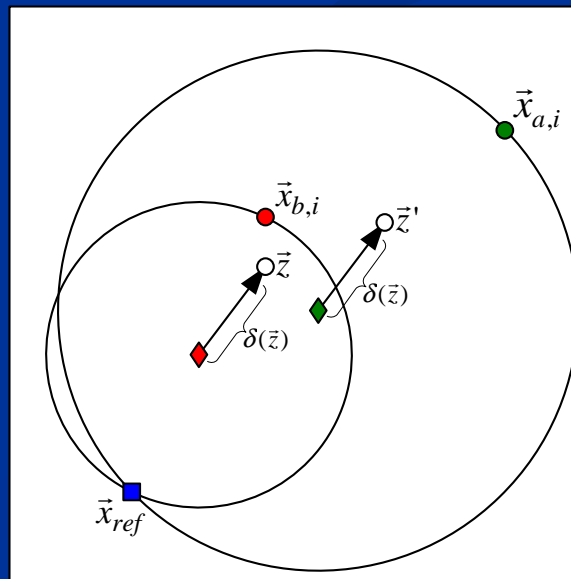
$$u_{a,i} \geq u_{b,i}.$$



b dom a

g dom c, e, f, h

k dom c, h



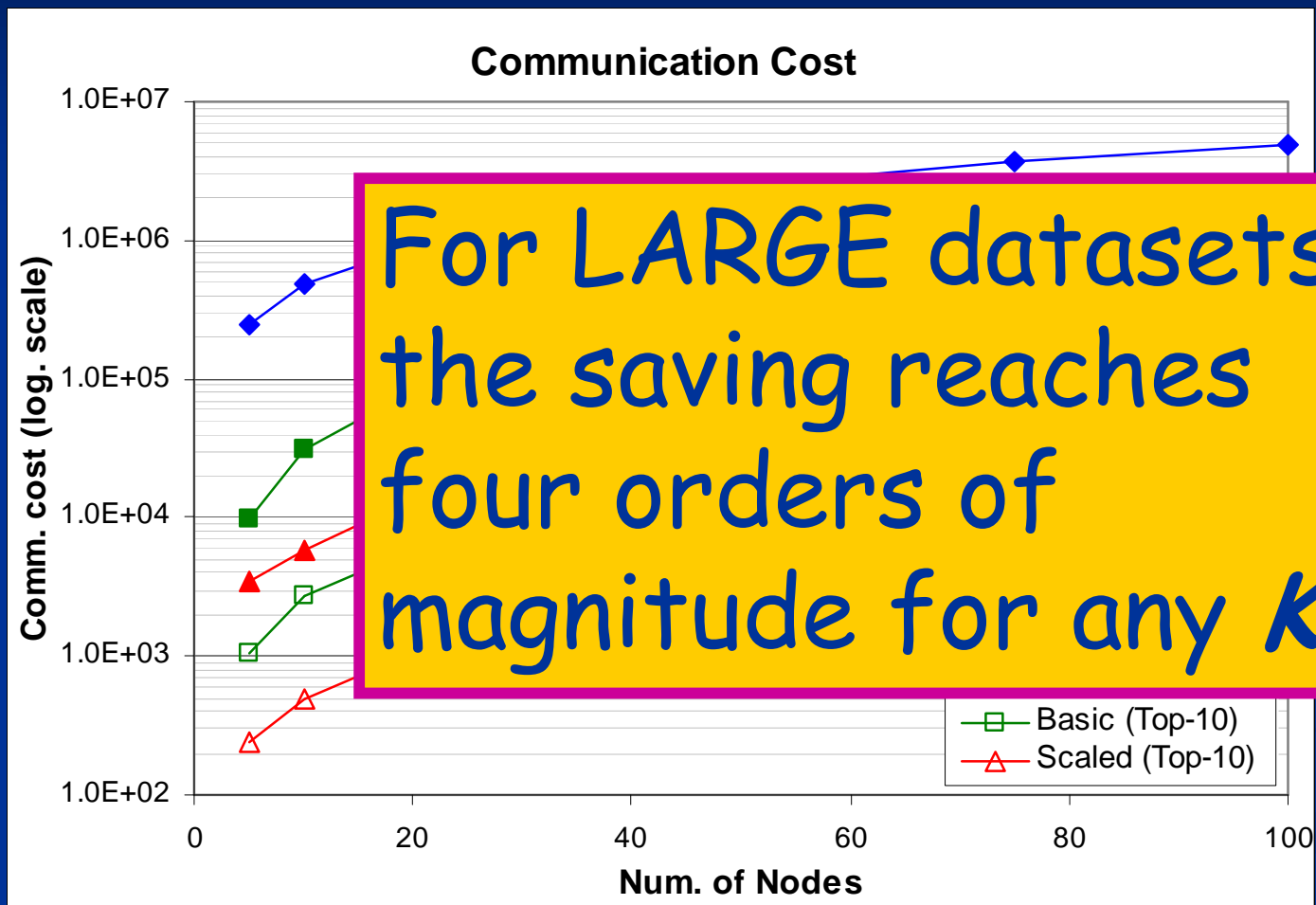
Results

- Queries collected from a search engine over a period of 3 months.

(<http://gregsadetsky.com/aol-data/>)

<i>Keywords</i>	<i>Score</i>
donnie, mcclurkin	0.92328
duff, hilary	0.85893
las, vegas	0.85488
estate, real	0.83995

Communication



Conclusions

- Local filtering as a first choice in large-scale distributed data stream systems
- Not necessary to sacrifice precision
- **Saving is unlimited**
 - Bounded only by the size of the data over system lifetime
- Facilitates high scalability, Autonomous operation, and Privacy
- Threshold is a building block for many data mining (machine learning) algorithms

Issues

- Heuristic, not a theory
- Computation complexity in nodes
- Convex Hull a lower bound to efficiency, is it really necessary?

A scenic landscape photograph showing a river with white-water rapids flowing through a dense forest of evergreen trees. The river is surrounded by large, smooth boulders. In the background, there are mountains under a cloudy sky. The word "Questions?" is overlaid in white text in the center of the image.

Questions?