NexTech 2011 - AP2PS 2011
The Third International Conference on Advances in P2P Systems, November 20-25, 2011, Lisbon, Portugal

Keynote Presentation:

Epidemic Protocols in Peer-to-Peer Computing

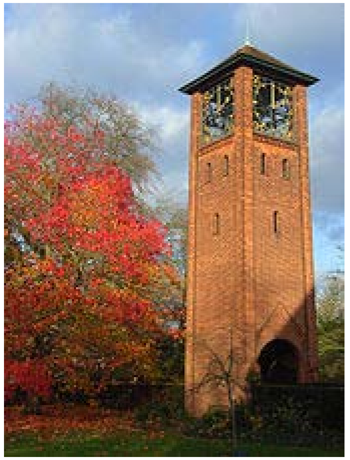
Dr. Giuseppe Di Fatta

G.DiFatta@reading.ac.uk

University of Reading

Monday, November 21, 2011

The University of Reading



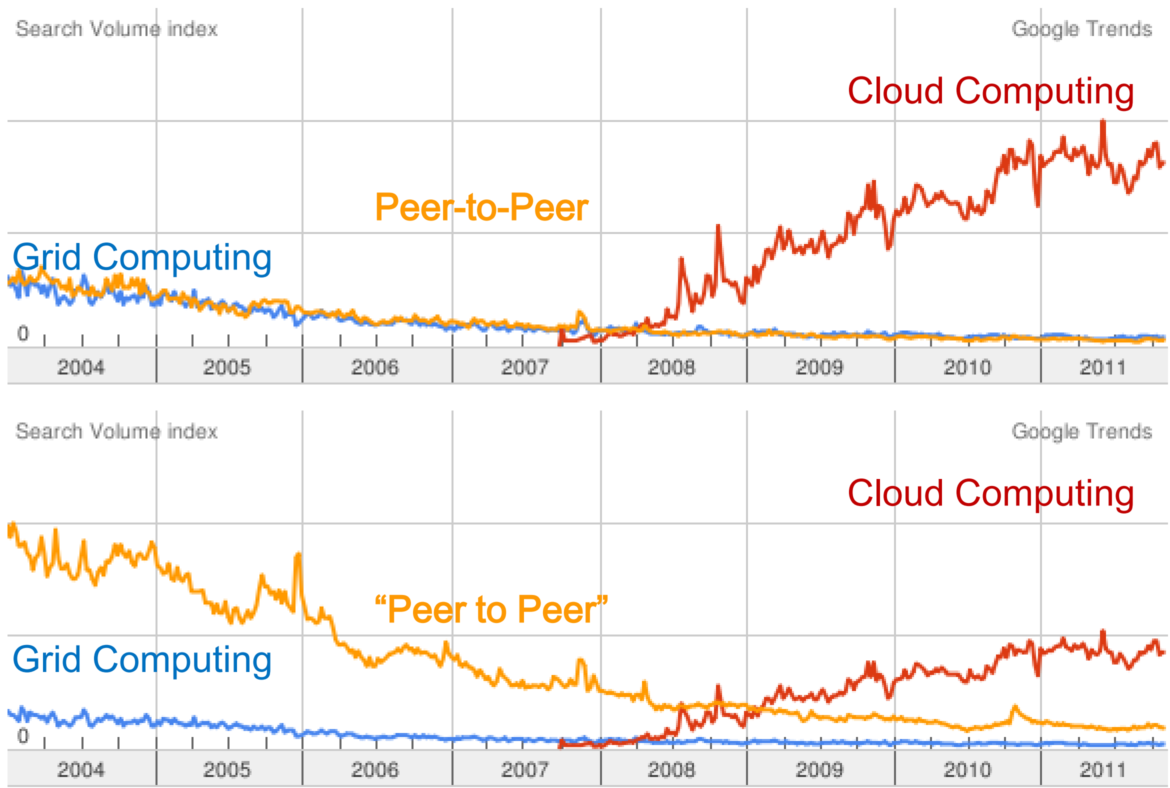
- Established in 1892 as an extension of the Christ Church College of the University of Oxford.
- Received its Royal Charter in 1926.
- Awarded the Queen's Anniversary Prize for Higher and Further Education in 1998, 2005 and 2009.
- One of the ten most research intensive universities in the UK.
- Campus voted as one of best green spaces in the UK in 2011.

Outline

- Introduction
- Gossip or Epidemic protocols
  - robustness and efficiency
  - push vs. pull schemes
  - convergence speed and accuracy
- Applications in large-scale systems
  - information dissemination vs. global knowledge
  - the data aggregation problem
- Future applications in/of P2P systems
- Open issues, research directions and conclusions

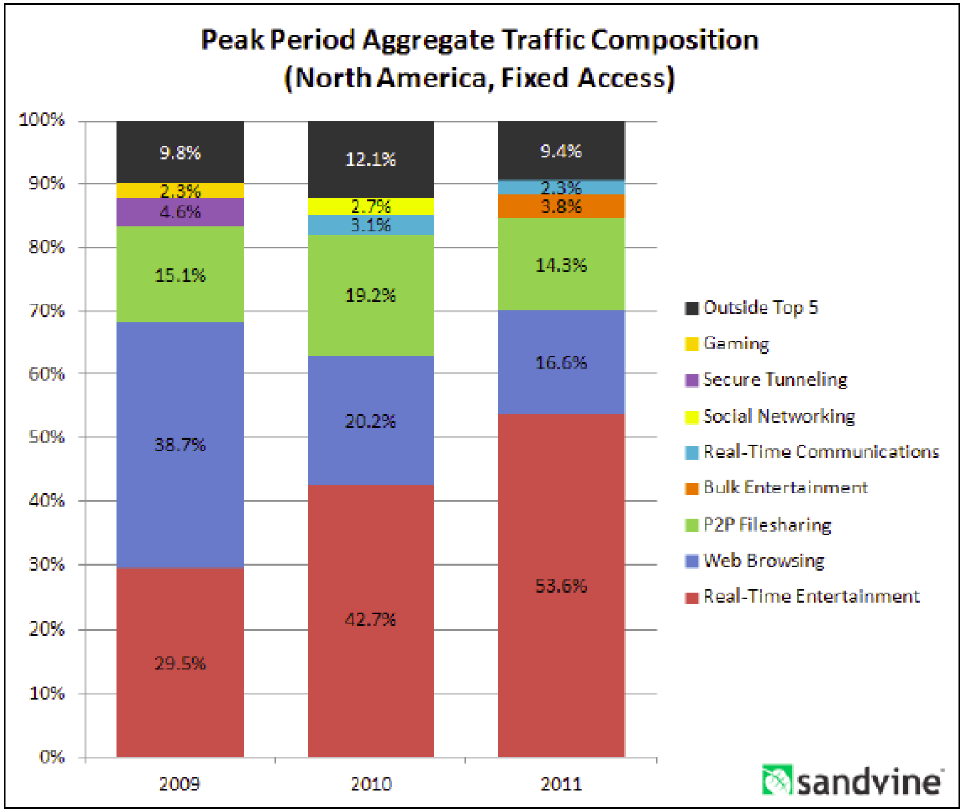
Is Peer-to-Peer in Decline?

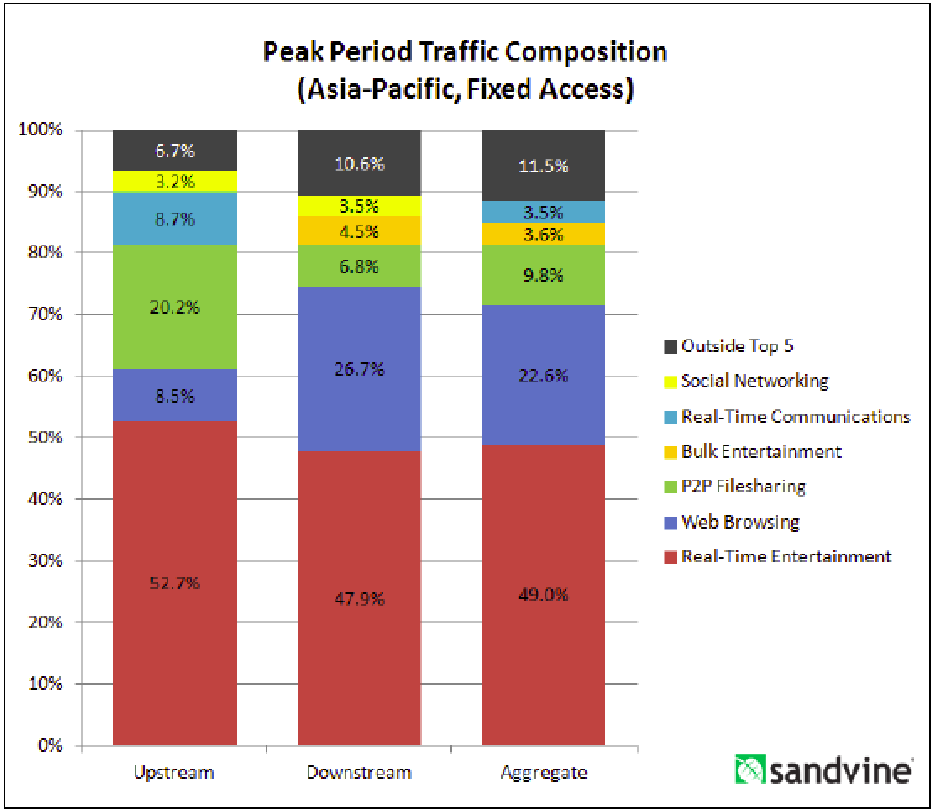
- Google trends are often (and arguably) shown as
  - evidence for the decline of a subject or
  - to advocate the rise of another



Is Peer-to-Peer in Decline?

- Facts [source: *Sandvine's Global Internet Phenomena Report: Fall 2011*]
  - P2P file sharing traffic as % of overall IP traffic has declined
  - overall IP traffic and P2P file sharing traffic have increased





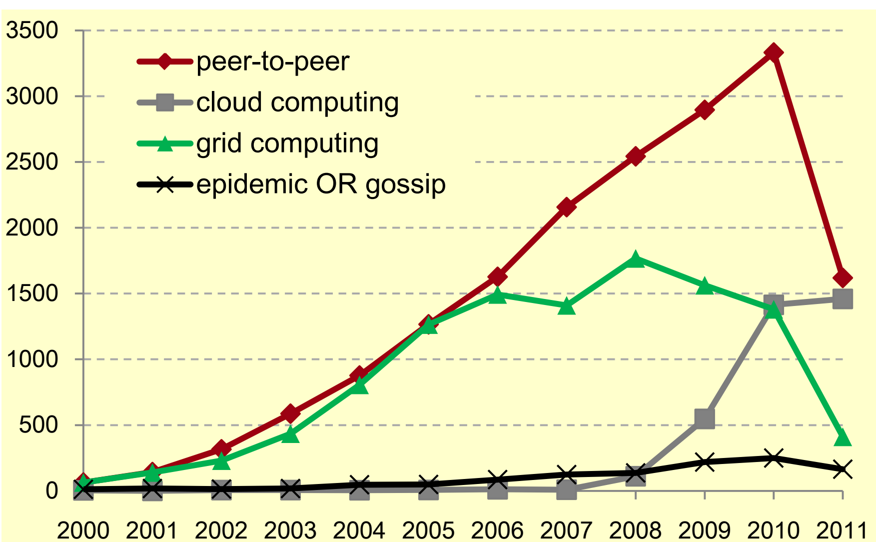
Is Peer-to-Peer in Decline?

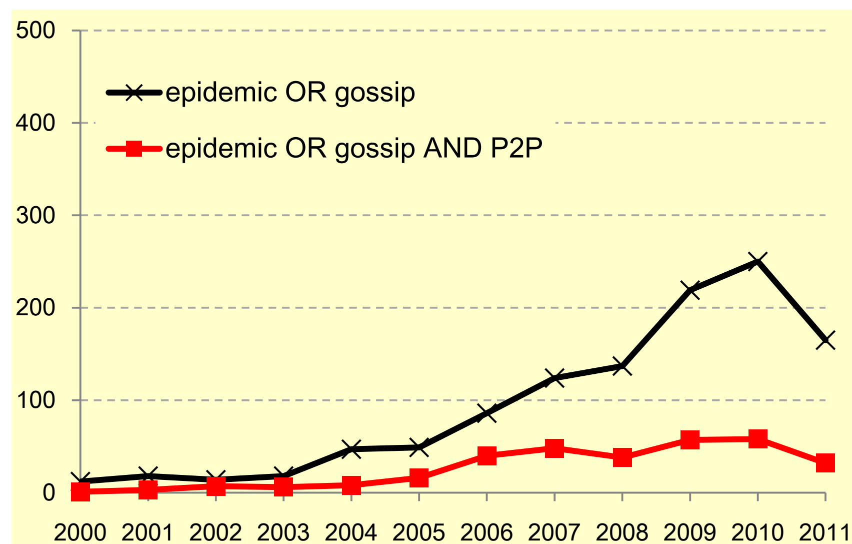
- Decline of P2P file sharing applications
  - Security and legal issues
    - Malware distributed in place of content
    - Many organisations block ports of P2P applications
  - P2P has been replaced by other means of file sharing
    - RapidShare, Megavideo, iTunes, iPlayer, Hulu, Netflix, etc.

- P2P paradigm emancipation
  - applications beyond file sharing
    - VoIP, video chat, live video streaming,
    - data-intensive ad-hoc applications, e.g., the CERN Advanced Storage system (CASTOR)
    - volunteer computing, Clouds integration
    - social media, online social networking

Papers Statistics

- Source: IEEE Xplore
  - Keyword search: Metadata Only
  - Publisher: IEEE
  - Content Types: Conferences, Journals
  - Subjects: Computing & Processing (Hardware/Software), Communication, Networking & Broadcasting





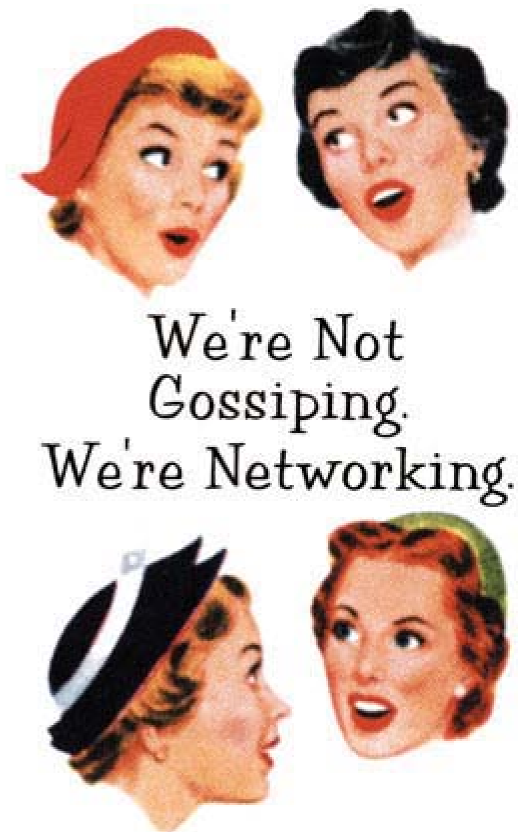
Gossip

- Etymology: "gossip" is from Old English *godsibb* (= godparent)

- Gossip is rumor, possibly the oldest and most common mean of sharing facts and opinions.
  - peer to peer information spreading

- From an evolutionary biology point of view, it aids social bonding in large groups.
  - overlay networks

- From an evolutionary psychology point of view, it aids building cooperative reputations and maintaining widespread indirect reciprocity: altruistic behaviour is favoured by the probability of future mutual interactions (randomly chosen pair-wise encounters).
  - tit for tat



Epidemic

- Etymology: “epidemic” is from Greek words *epi* and *demos* (= upon or above people).
- In epidemiology it is a disease outbreak. It occurs when new cases exceed a "normal" expectation of propagation (a contained propagation).
  - The disease spreads person-to-person: the affected individuals become independent reservoirs leading to further exposures.
  - In uncontrolled outbreaks there is an exponential growth of the infected cases.

Epidemic curve of the outbreak of new influenza A(H1N1) in Mexico and fitted exponential growth over the period 9 to 24 April 2009

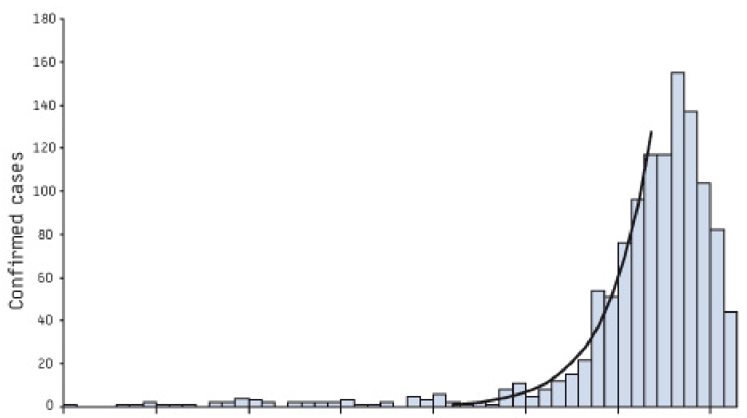


Figure from: “Rapid communications A preliminary estimation of the reproduction ratio for new influenza A(H1N1) from the outbreak in Mexico, March-April 2009", P Y Boëlle, P Bernillon, J C Desenclos, Eurosurveillance, Volume 14, Issue 19, 14 May 2009

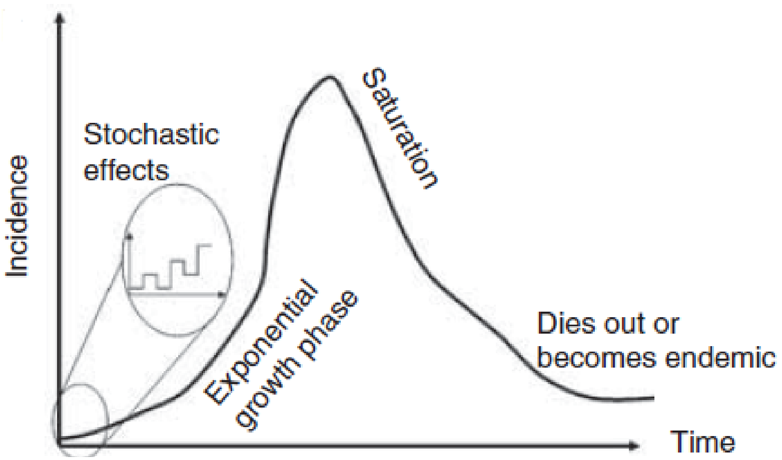


Figure from: “Controlling infectious disease outbreaks: Lessons from mathematical modelling”, T Déirdre Hollingsworth, Journal of Public Health Policy 30, 328-341, Sept. 2009

A Bio-Inspired Paradigm

- Epidemic or Gossip protocols are a communication and computation paradigm for large-scale networked systems
  - based on randomised communication,
  - provides
    - scalability,
    - probabilistic guarantees on convergence speed and accuracy,
    - robustness, resilience,
    - fault-tolerance, high stability under disruption,
    - computational and communication efficiency.

Seminal Work and History

- Clearinghouse Directory Service, Demers et al., Xerox PARC, 1987
- The refdbms distributed bibliographic database system, Golding et al., 1993
- Bayou project, Demers et al., Xerox PARC, 1993-97
- Bimodal Multicast, Cornell, 1998
- Astrolabe, Cornell, 1999
- 2000-2005, a few papers studied and extended the use of Epidemic approaches in communication networks and distributed systems

Applicability

- Information Dissemination
  - Epidemic protocols can be used to disseminate information in large-scale distributed environments.
    - broadcasting, multicasting, failure detection, synchronisation, sampling, replica maintenance, monitoring, management, etc.
- Data Aggregation
  - Epidemic protocols can also be adopted to solve the data aggregation problem in a fully decentralized manner.
- Complex applications can be built from these basic services for very dynamic and very large-scale distributed systems.
  - e.g., fully decentralised Data Mining applications for large-scale distributed systems.

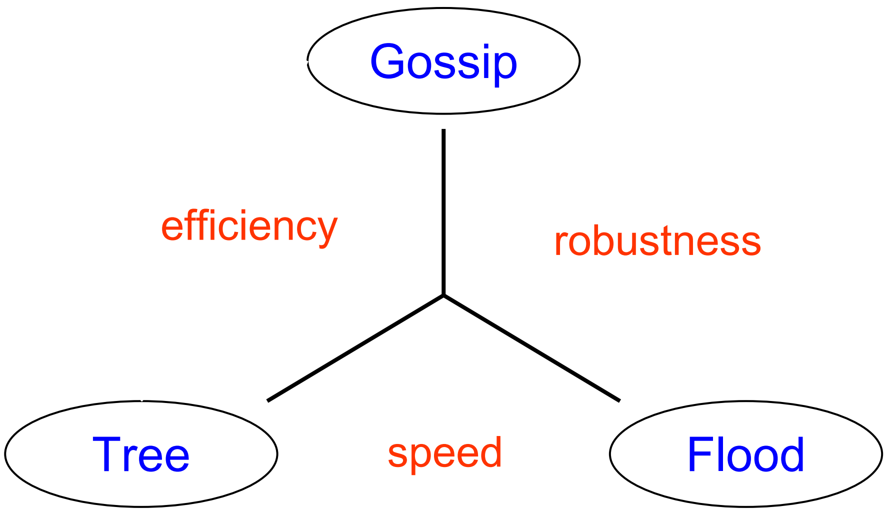
Information Dissemination

- Epidemic information dissemination with probabilistic guarantees:
  - Anti-entropy
    - every node periodically chooses another node at random and resolves any differences in state
  - Rumour mongering
    - infected nodes periodically choose a node at random and spread the rumour
  - Gossiping
    - each node forwards a message probabilistically

Information Dissemination

University of Reading

- Protocols for information dissemination in large-scale systems should have the following properties:
  - Efficiency, Robustness, Speed, Scalability
- Alternative approaches:
  - Tree-based: efficient, but fragile and difficult configuration
  - Flooding: robust, but inefficient
  - Gossip-based: both efficient and robust, but has relatively high latency



Gossip-based Protocol

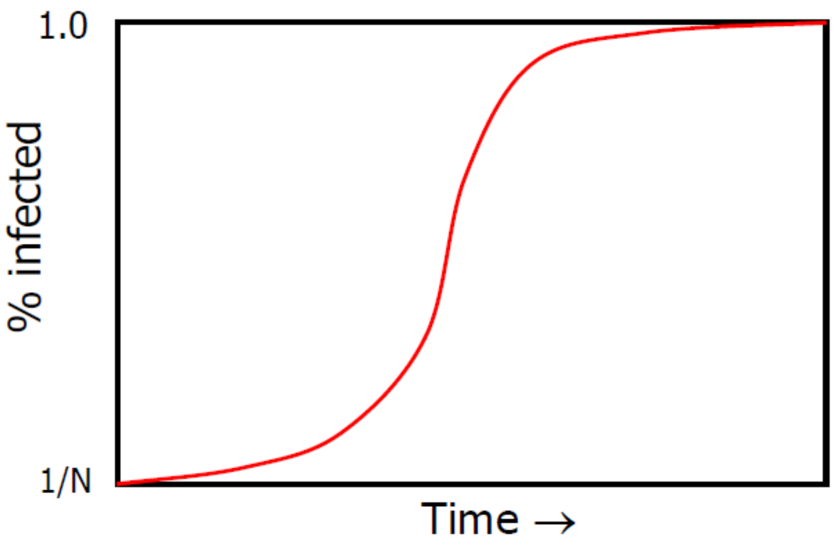
- Based on randomised communication and
  - peer selection mechanism
  - definition of state and merge function

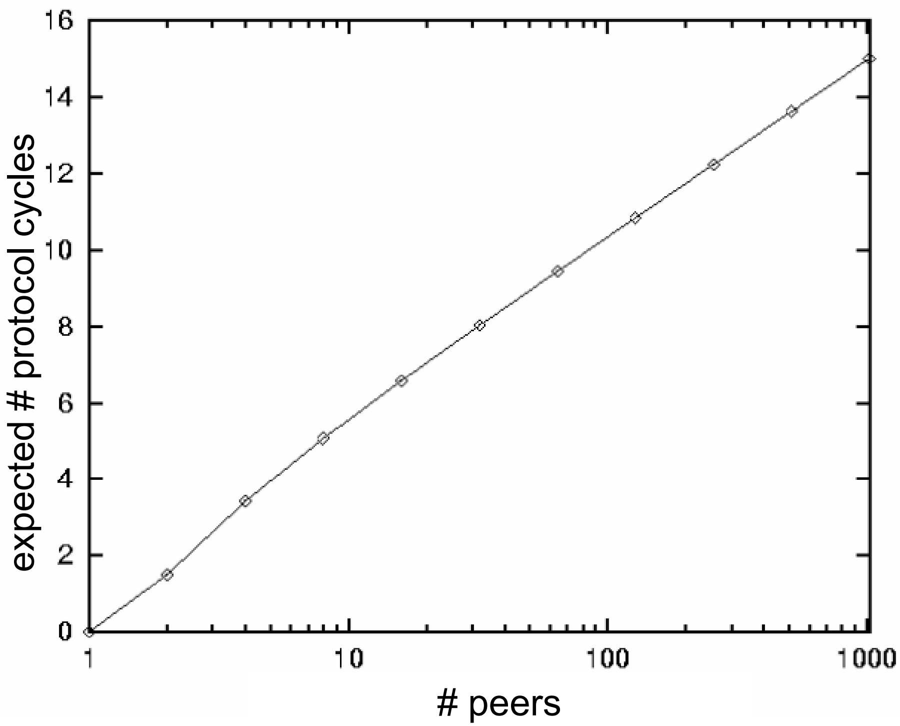
- Repeat
  - wait some ΔT
  - chose a random peer
  - send local state

- Repeat
  - receive remote state
  - merge with local state

Gossip Propagation Time

- Time to propagate information originated at one peer





Time to complete “infection”: O(log N)

Variants

- Push epidemic
  - each peer sends state to other member
- Pull epidemic
  - each peer requests state from other member
  - starts slowly, ends quickly
  - expected #rounds the same
- Push/Pull epidemic
  - Push and Pull in one exchange
  - reduces #rounds, but increases overhead

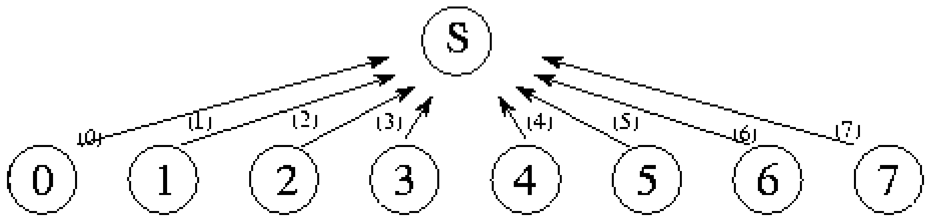
Data Aggregation

- (a.k.a. the "node aggregation" problem)
- Given a network of N nodes, each node i holding a local value x_i,
- the goal is to determine the value of a global aggregation function f() at every node:

$$f(x_0, x_1, ..., x_{N-1})$$

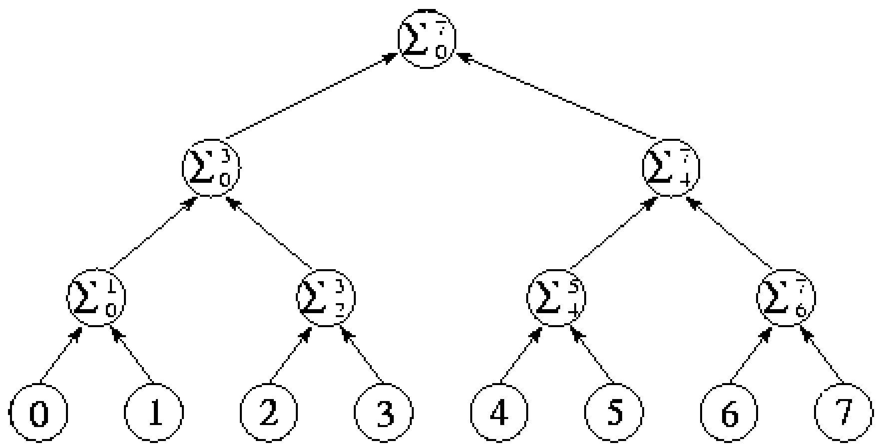
- Example of aggregation functions:
  - sum, average, max, min, random samples, quantiles and other aggregate databases queries.

Aggregation: e.g., Sum



$$s = \sum_{i=0}^{N-1} x_i$$

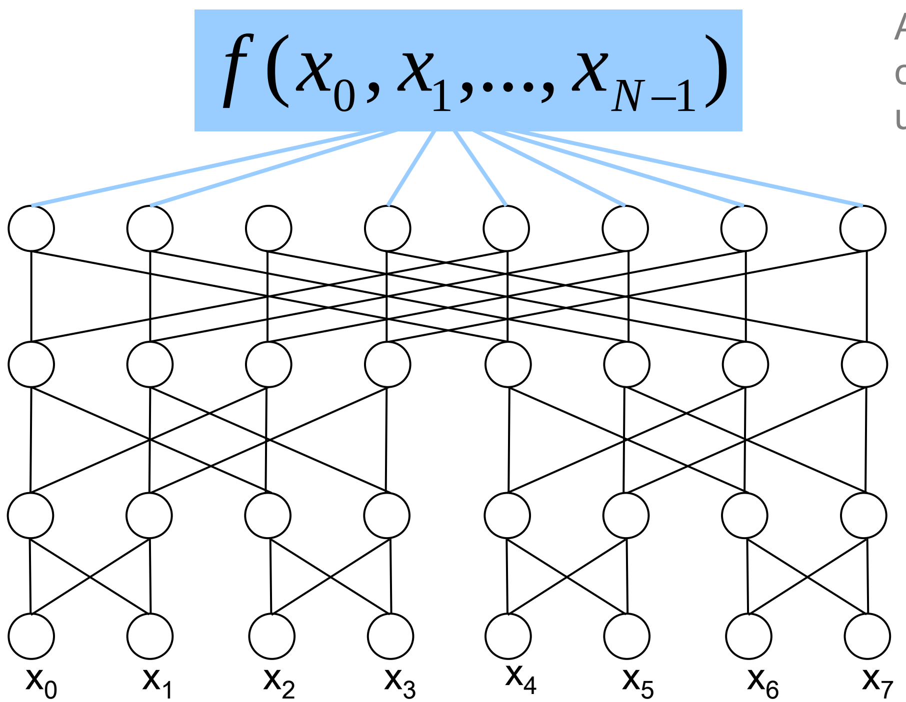
- Centralised approach: all receive operations, and all additions, must be serialized: O(N)



- Divide-and-conquer strategy to perform the global sum with a binary tree: the number of communication steps is reduced from O(N) to O(log(N)).

All-to-all Communication

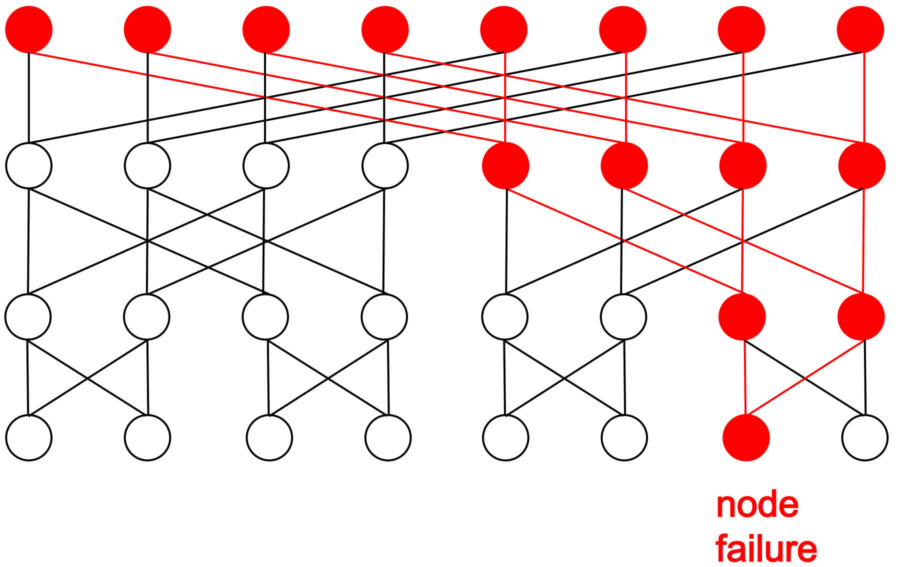
- MPI AllReduce
  - MPI predefined operations: max, min, sum, product, and, or, xor
  - all processes compute identical results
  - number of communication steps: log(N)
  - number of messages: N*log(N)



Any global function which can be approximated well using linear combinations.

Fault-Tolerance and Robustness

- The parallel approach is not fault tolerant.
- Even a single node or link failure cannot be tolerated.
- A delay on a single communication link has an effect on all nodes.



- In large-scale and dynamic distributed systems we require the protocols to be decentralised and fault-tolerant.

The Push-Sum Protocol (PSP)

University of Reading

- Each node i holds and updates the local sum $s_{t,i}$ and a weight $w_{t,i}$.
- Initialisation:
  - Node i sends the pair $<x_i, w_{0,i}>$ to itself.
- At each cycle t:

Algorithm 1 Protocol Push-Sum

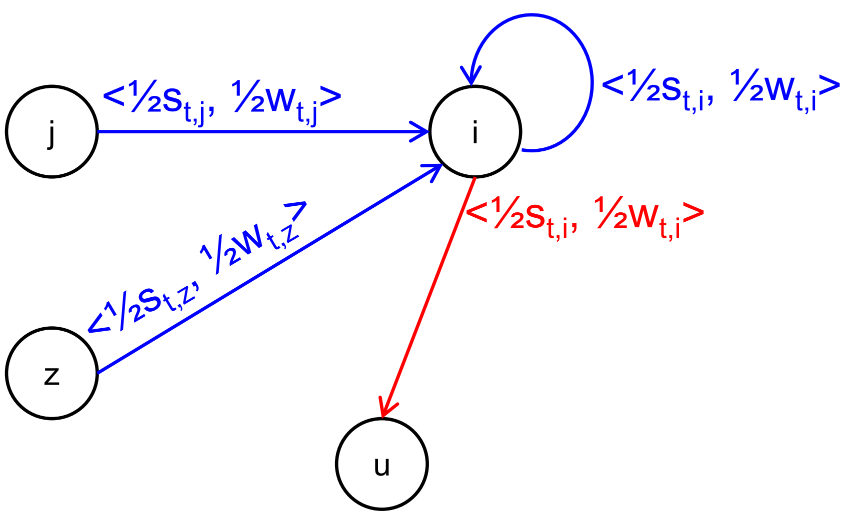
Let $\{(\hat{s}_r, \hat{w}_r)\}$ be all pairs sent to i in round $t-1$

Let $s_{t,i} := \sum_r \hat{s}_r$, $w_{t,i} := \sum_r \hat{w}_r$

Choose a target $f_t(i)$ uniformly at random

Send the pair $(\frac{1}{2}s_{t,i}, \frac{1}{2}w_{t,i})$ to $f_t(i)$ and i (yourself)

$\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average in step t



- Update at node i:

$$\begin{cases} s_{t+1,i} = \frac{1}{2}s_{t,j} + \frac{1}{2}s_{t,i} + \frac{1}{2}s_{t,z} \\ w_{t+1,i} = \frac{1}{2}w_{t,j} + \frac{1}{2}w_{t,i} + \frac{1}{2}w_{t,z} \end{cases}$$

variance reduction step

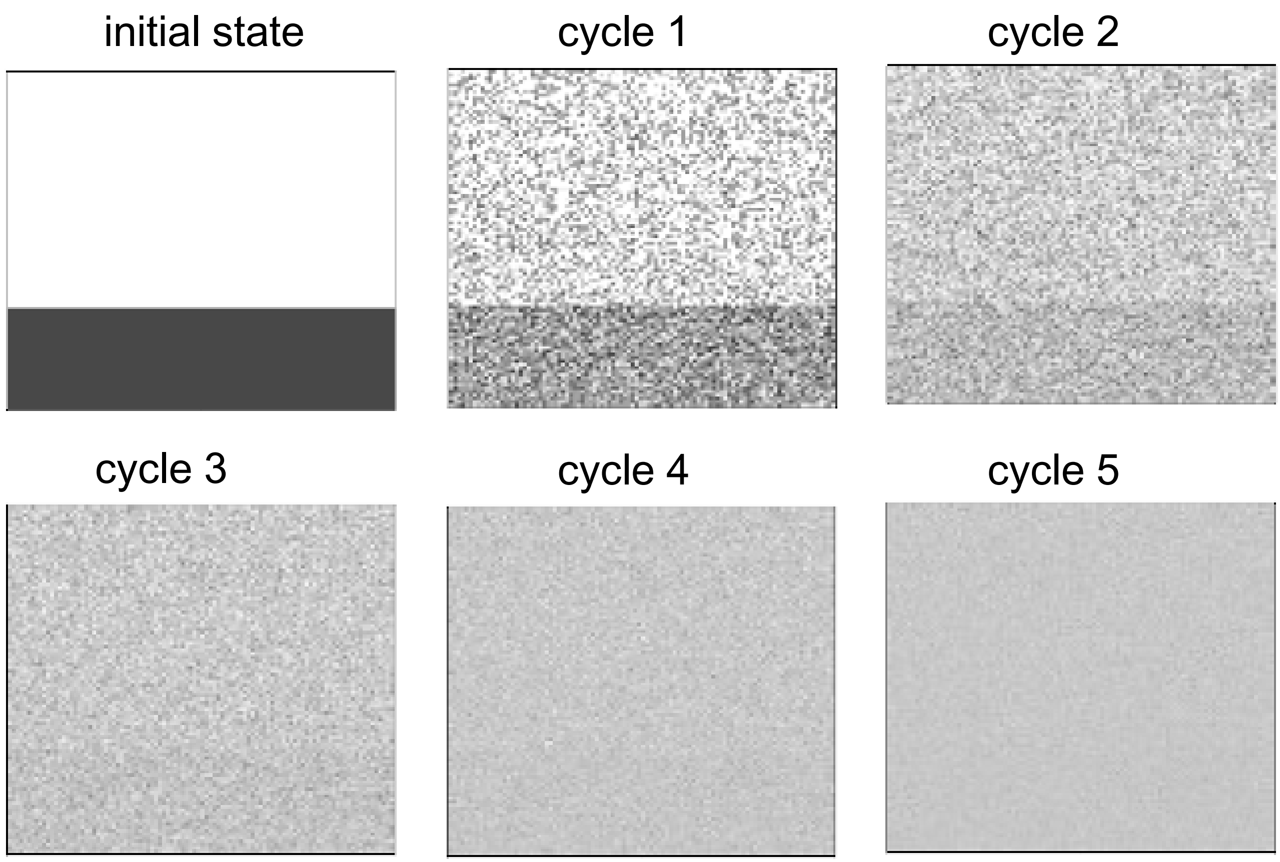
The Push-Sum Protocol (PSP)

- Settings for various aggregation functions:

Function	Description
Sum	v_i = local value $w_i = 1$ at a single node, 0 at all other nodes
Count	$v_i = 1$ $w_i = 1$ at a single node, 0 at all other nodes
Average	v_i = local value $w_i = 1$
Weighted Average	v_i = local value $\times$ local weight w_i = local weight

- Convergence: with probability 1-δ the relative error in the approximation of the global aggregate is within ε, in at most $O(\log(N) + \log(1/\varepsilon) + \log(1/\delta))$ cycles.

Example: Average

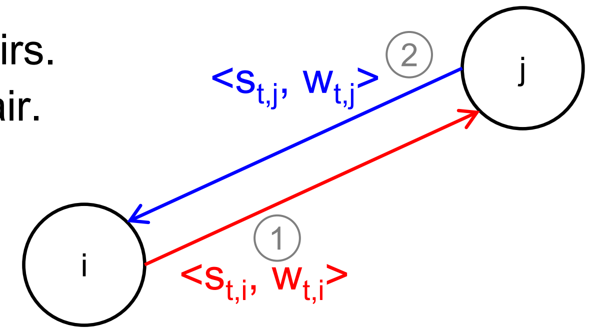


(Figure from: Mark Jelasity, RESCOM 2008)

The Push-Pull Gossip (PPG) Protocol

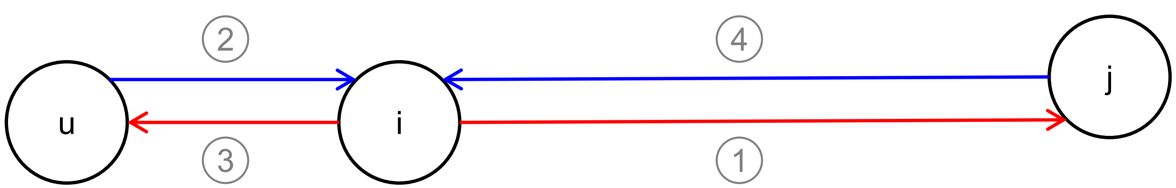
University of Reading

- At each push PPG introduces a symmetric pull operation: local pairs are exchanged.
  - Node i selects a random node j to exchange their local pairs.
  - Each node compute the average and updates the local pair.



variance reduction step:
$$\begin{cases} s_{t+1,i} = \frac{1}{2}(s_{t,j} + s_{t,i}) \\ w_{t+1,i} = \frac{1}{2}(w_{t,j} + w_{t,i}) \end{cases}$$

- The push-pull operations need to be performed atomically.
  - If not, the conservation of mass in the system is not guaranteed and the protocol does not converge to the true global aggregate.



Mass Conservation Invariant

- The mass conservation invariant states that the average of all local sums is always the correct average and the sum of all weights is always N.

- Protocols violating this invariant cannot converge to the true global aggregate.

Diffusion Speed

- The diffusion speed is how quickly values originating at a source diffuse evenly through a network (convergence).
  - number of protocol iterations such that the value at a node is diffused through the network, i.e., a peak distribution is transformed in a uniform distribution.
  - The diffusion speed is typically given as the complexity of the number of iteration steps as function of the network size, maximum error and maximum probability that the approximation at a node is larger than the maximum error.

- Diffusion speed: with probability $1-\delta$ the relative error in the approximation of the global aggregate is within ε, in at most $O(\log(N) + \log(1/\varepsilon) + \log(1/\delta))$ cycles, where ε and δ are arbitrarily small positive constants.

Convergence Factor

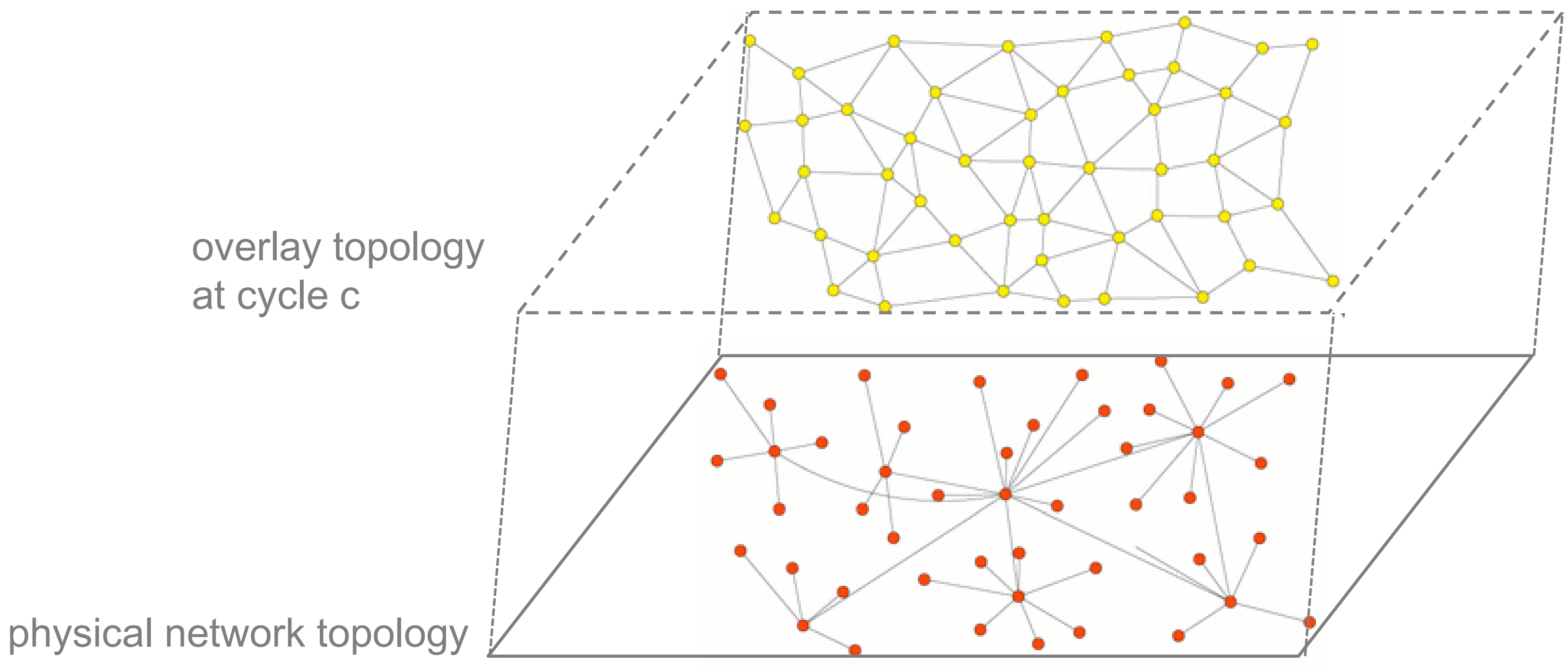
- At each cycle, each node estimates the global aggregate.
- This estimated value converge exponentially fast.
- The convergence factor is the speed with which the local approximations converge towards a target value (not necessarily the true global aggregate).
- The convergence factor between cycle t+1 and cycle t is given by the ratio of the variance:

$$E(\sigma_{t+1}^2) / E(\sigma_t^2)$$

- A smaller factor gives faster convergence.

Peer Selection

- At each cycle (synchronous model), the peers involved in communication operations define a transient random overlay network.

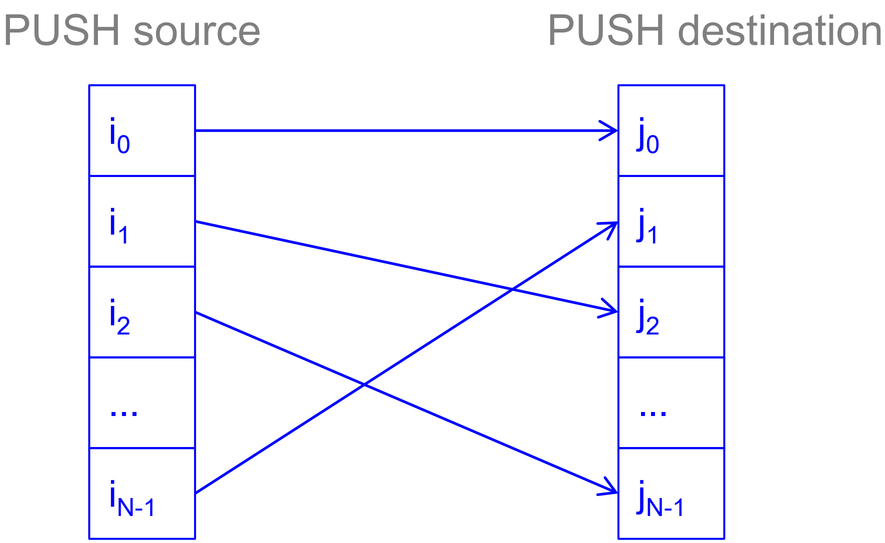


Random Overlay Network

- Directed network edge <i,j>: peer p_i sends a PUSH msg to peer p_j.
- At each cycle, there is a list of edges, i.e., two lists of peers (src and dest)

Source list: $p_{i_0}, p_{i_1}, \ldots, p_{i_{N-1}}$

Dest. list: $p_{j_0}, p_{j_1}, \ldots, p_{j_{N-1}}$



Random Overlay Network

- Random peer selection for push/pull operations

Convergence factor

  - perfect matching (PSP): matching of pairs to achieve perfect distribution of push operations: each node sends a push and receives a push.

    $\frac{1}{2}$ optimal (PSP)

  - perfect matching (PPG): matching of pairs to achieve perfect distribution of push and pull operations: each node sends a push and a pull and receives a push and a pull.

    $\frac{1}{4}$ optimal (PPG)

  - random pairs (PPG): push operations both sent and received by a node follow the binomial distribution.

    $\frac{1}{e} \approx \frac{1}{2.718} \approx 0.368$

  - random PUSH target: matching of pairs to achieve perfect distribution of push (not pull) operations: each node sends a push and may receive zero, one or more push messages.

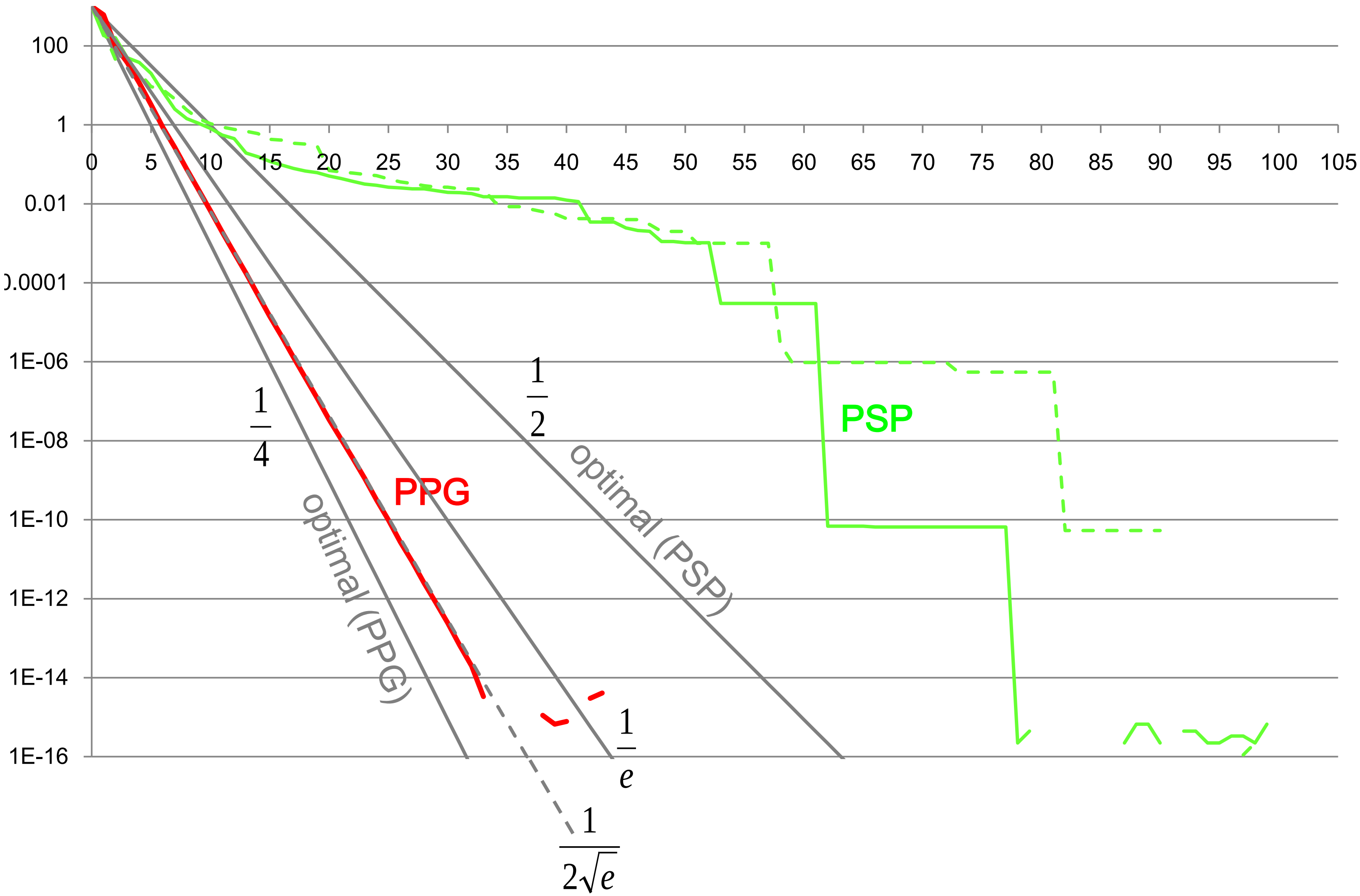
    $\frac{1}{2\sqrt{e}} \approx \frac{1}{3.297} \approx 0.303$

Practical Peer Sampling

- Practical peer selection in a large-scale distributed system for push/pull operations:
  - Peer Selection Protocol:
    - A local cache of (max size) peer IDs is maintained and used to draw a random sample of peers.
      - The node cache is initialised with the known physical neighbours.
      - Caches are exchanged (likewise push/pull messages) and randomly trimmed to a maximum size.
  - This is equivalent to multiple random walks: the cache entries quickly converges to a random sample of the peers with uniform distribution (in expander graphs).

PPG vs PSP

- Convergence factor



PPG vs PSP

University of Reading

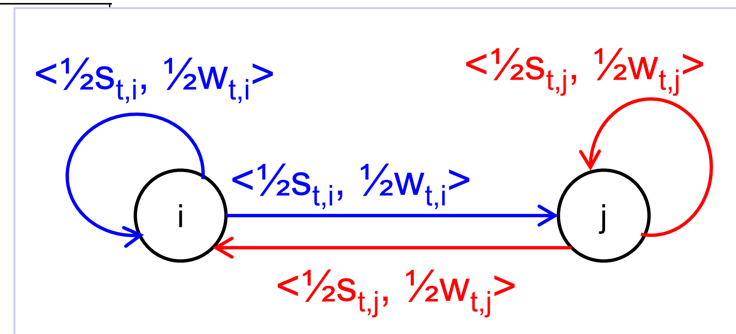
- Not surprisingly PPG has faster diffusion speed than PSP.
  - At each cycle, in PPG twice #messages are sent w.r.t. PSP.
  - The symmetry in the push-pull scheme allows every single node to be involved in at least one variance reduction step per cycle.

- In PSP at each cycle, a node has 37% chance of not receiving any push. In practical implementations of the peer sampling operation, this may generate connectivity problems.

- PPG requires atomic push-pull operations to guarantee the mass conservation invariant.
  - Atomic push-pull operations can be complex.

The Symmetric Push-Sum Protocol (SPSP)

- SPSP is a Push-Pull scheme with asynchronous communication
  - no atomic operation is required.

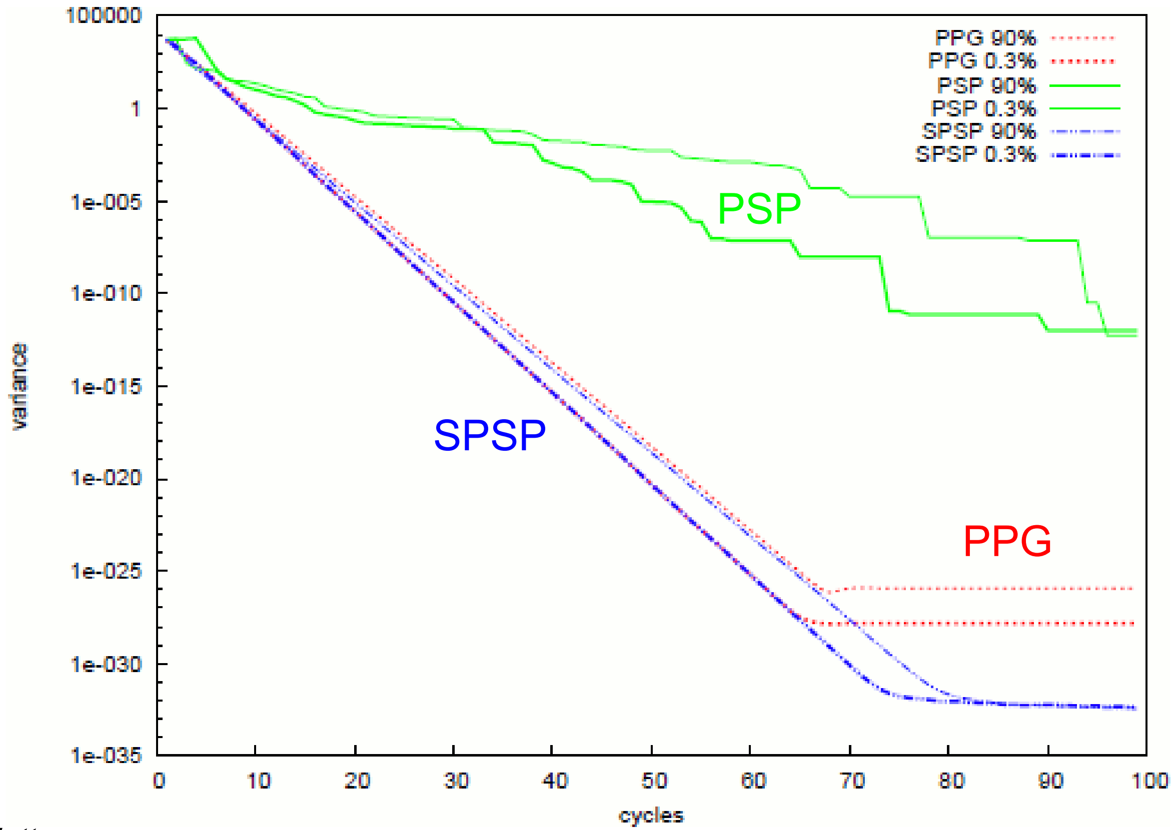


At each node i
Require: v_0, w_0
  The initial local value, v_0;
  The initial local weight, w_0.
Initialisation:
  $(v, w) = (v_0, w_0)$
At each cycle:
  $j \leftarrow getNode()$
  $v \leftarrow v/2$, $w \leftarrow w/2$
  send an *aggregation message* to j, $\langle (v, w), \textbf{true} \rangle$
At event: received an *aggregation message* $\langle (v', w'), r \rangle$
  from j
  if r is **true then**
    $v \leftarrow v/2$, $w \leftarrow w/2$
    send an *aggregation message* to j, $\langle (v, w), \textbf{false} \rangle$
  end if
  $v \leftarrow v + v'$, $w \leftarrow v + w'$

Comparative Analysis (PSP, PPG, SPSP)

University of Reading

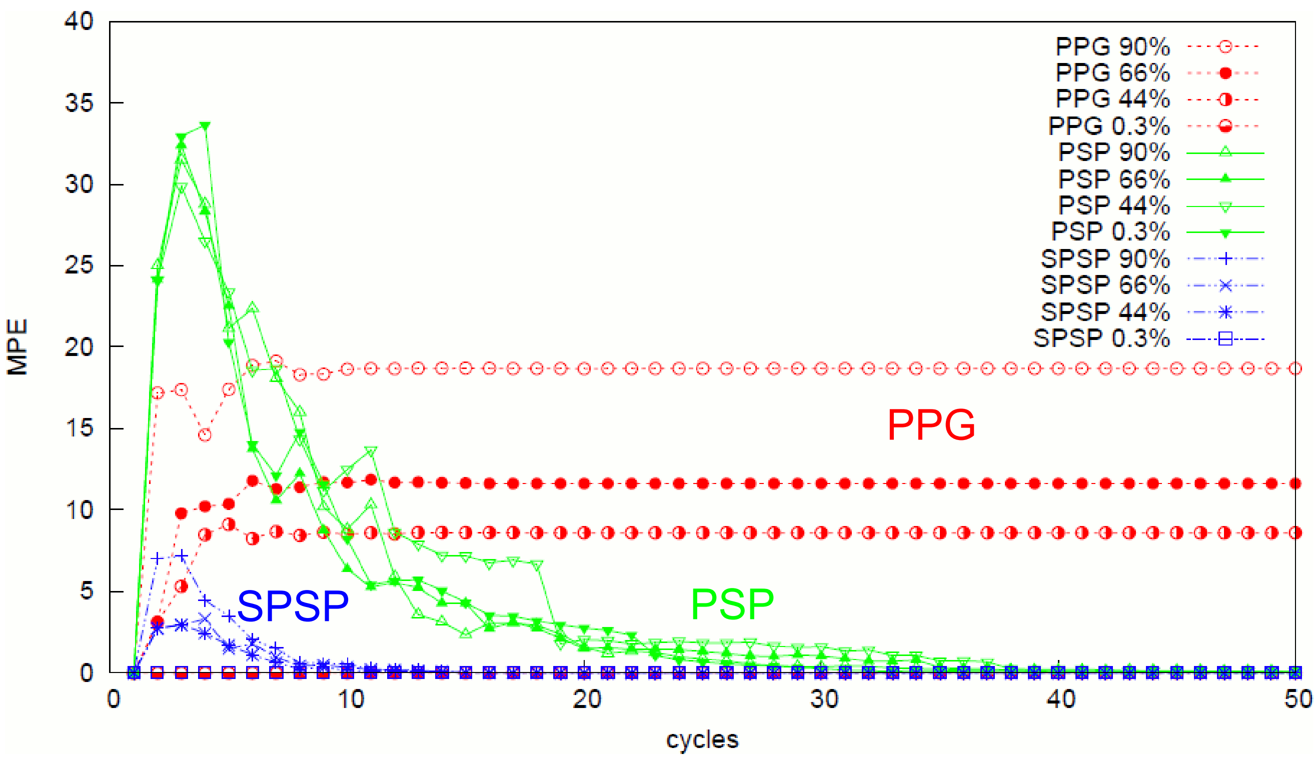
- Convergence speed: variance of the estimated global aggregate over time
  - Percentage of operations with atomicity violation (AVP): 0.3% and 90%,
  - Internet-like topologies, 5000 nodes.
  - PPG and SPSP convergence speed is similar w.r.t. AVP.



Comparative Analysis (PSP, PPG, SPSP)

University of Reading

- The mean percentage error (MPE) over time
  - different AVP levels (from 0.3% to 90%)
  - averages over 100 different simulations: Internet-like and mesh topologies, 1000-5000 nodes, different data distributions.
  - Only PSP and SPSP converge to the true global aggregate value.



Applications

- Gossip-based protocols have been adopted for applications in
  - network management and monitoring, failure detection, DB replica synchronisation and maintenance, etc.

- Gossip-based protocols can be adopted to build complex applications in P2P systems.
  - global vs. total knowledge: aggregation
    - values of aggregate functions more important than individual data
    - discovery of global patterns and trends

Epidemic Data Mining for Global Knowledge Discovery in Peer-to-Peer Networks

Online Social Networks and P2P

- Online Social Networks (OSNs)
  - Web-based services that allow building relations among people to share information, activities and interests.
  - based on a centralised approach
  - several concerns: data ownership, privacy policies and scalability

- Decentralised Online Social Networks (DOSNs)
  - based on P2P overlay networks
  - motivated by privacy concerns and software freedom considerations
  - currently many serverless OSN frameworks and platforms are being studied and developed (e.g., *Diaspora*, *Tribler*, *Spar*, *What's up*, *Scope*, *SuperNova*, *PrPl*, OneSocialWeb)

Diaspora - the privacy aware, personally controlled, do-it-all distributed open source social network

KICKSTARTER | Discover great projects | Start your project | BLOG | HELP | SIGN UP | LOG IN

Decentralize the web with Diaspora

An Open Software project in New York, NY by Daniel G. Maxwell S. Raphael S. Ilya Z. · send message

PROJECT HOME | UPDATES 17 | BACKERS 6479 | COMMENTS 911

Like | 7450 likes. Sign Up to see what your friends like. | Tweet | <> EMBED | http://kck.st/9QC2zk

ABOUT THIS PROJECT

We're fully funded! Check out some of the other great projects on Kickstarter

Diaspora - the privacy aware, personally controlled, do-it-all distributed

6,479
BACKERS

$200,641
PLEDGED OF $10,000 GOAL

0
SECONDS TO GO

FUNDING SUCCESSFUL
This project successfully raised its funding goal on June 1, 2010.

PLEDGE $5 OR MORE
1473 BACKERS

Once the software is released as open source, we will send you a CD with diaspora all set up and ready to go, with a note from our team!

PLEDGE $10 OR MORE
1083 BACKERS

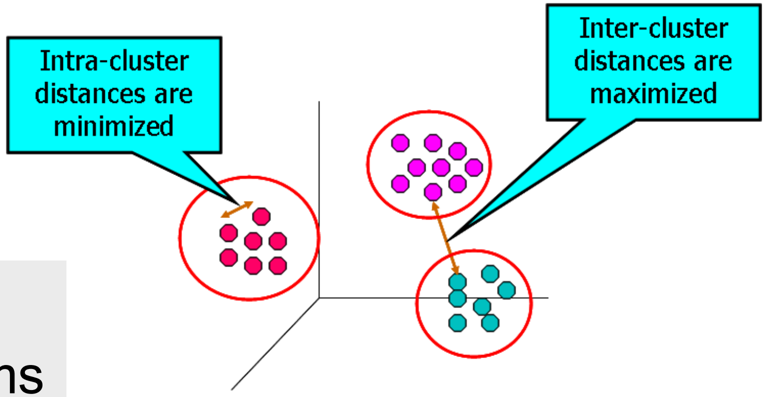
Clustering in DOSNs

- Scenario:
  - let us consider the case people in a DOSN want to find out about other people with similar orientation/preferences for socio-political issues, music, movies, etc.
  - We'd first need to deploy a distributed and fully decentralised Clustering algorithm to determine the groups of similar users globally, without the possibility to collect global data in a single server.

- Solution: Epidemic K-Means Clustering

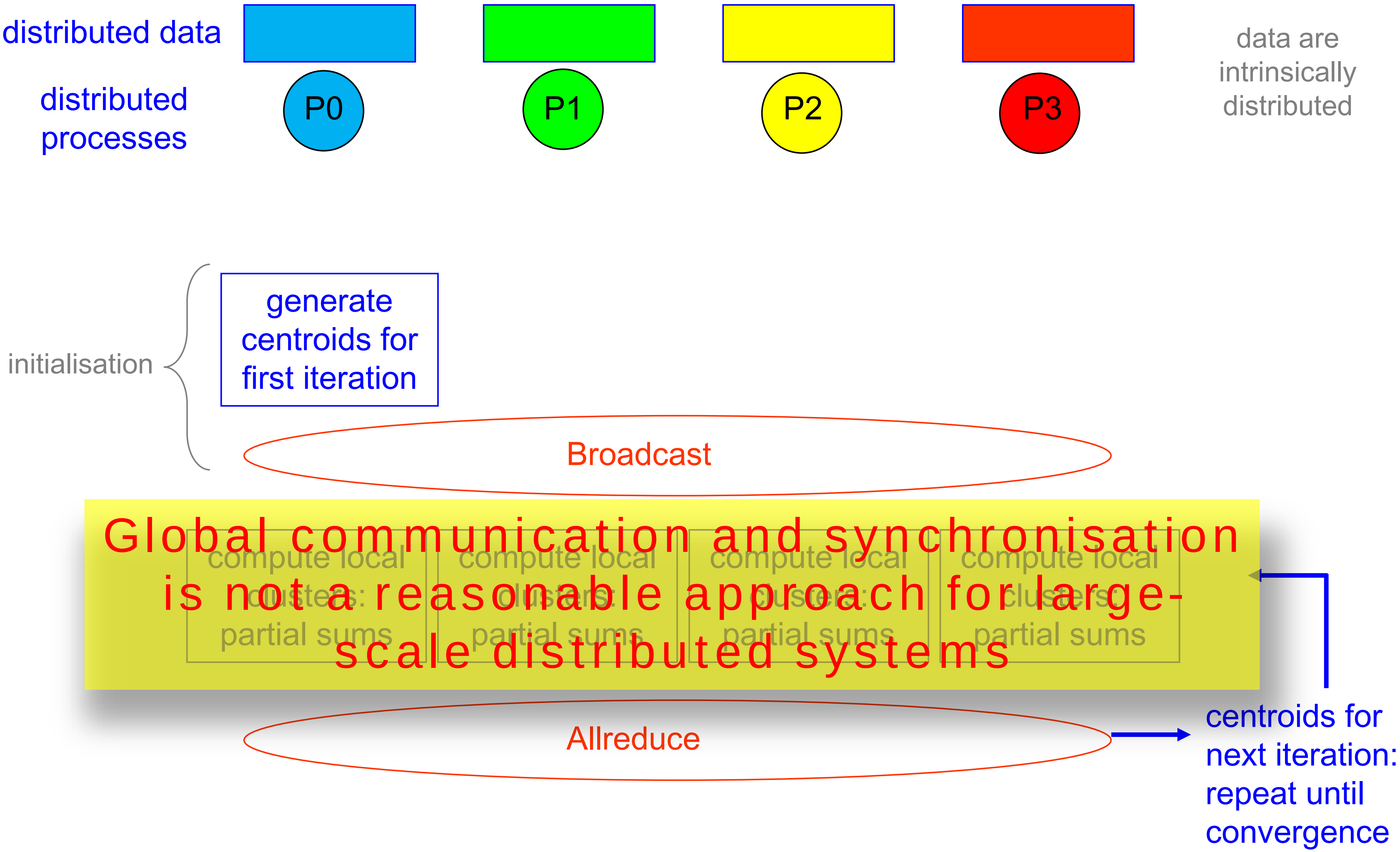
Clustering Analysis

- **Cluster Analysis** is the process of partitioning a set of data (or objects) in a set of meaningful sub-classes, called **clusters**.
  - natural grouping or structure in a data set.
- Cluster analysis = Grouping a set of data objects into clusters
- Cluster: a collection of data objects
  - similar to one another within the same cluster
  - dissimilar to the objects in other clusters
- Clustering is unsupervised classification:
  - no predefined classes



- K-Means Clustering is one of the most popular and influential Data Mining algorithms

Distributed K-Means



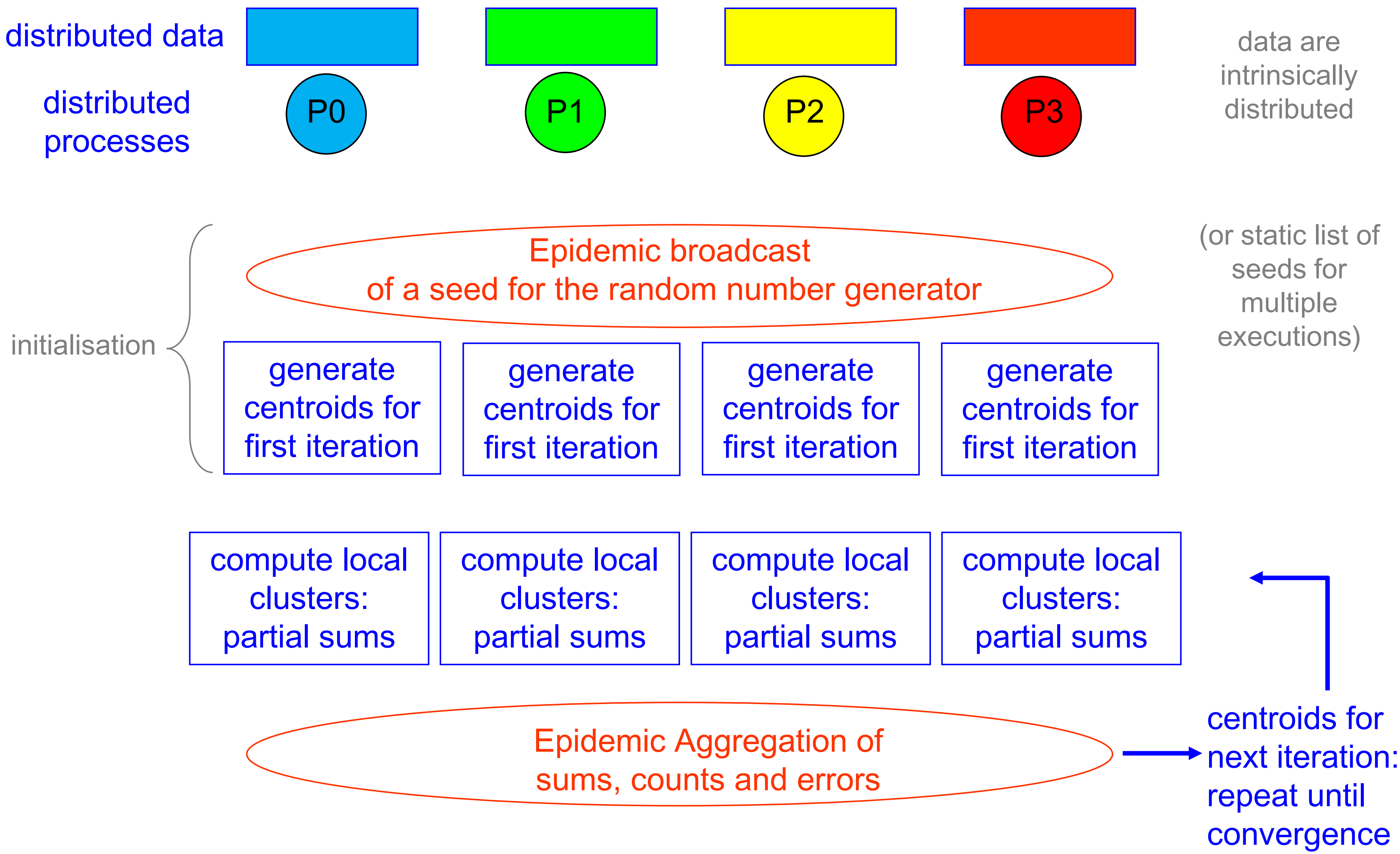
Global communication and synchronisation is not a reasonable approach for large-scale distributed systems

P2P K-Means Clustering

- Distributed K-Means (state of the art) algorithms for large-scale systems are based on a sampling strategy.
  - The parallel K-Means algorithm is applied to a subset of network nodes.

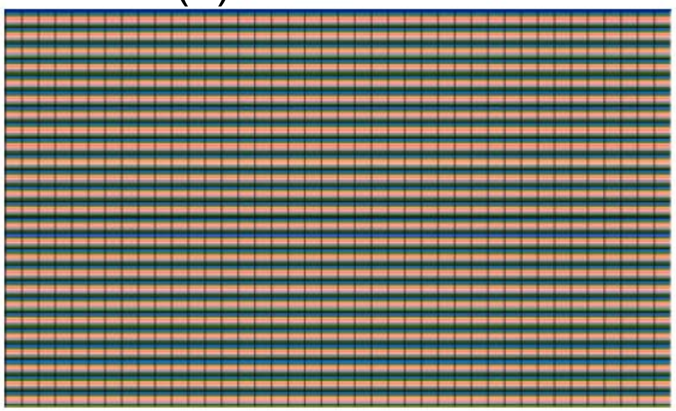
- Variants:
  - Local P2P Sampling-based K-Means
    - Each node communicates and synchronises only with its physical neighbours
  - Random Sampling-based P2P K-Means
    - Each node communicates and synchronises with a random sample of network nodes. The sample changes at each K-Means iteration.
  - Uniform Sampling-based P2P K-Means
    - Master-slave approach: only a leader node determines the final solution.

Epidemic K-Means

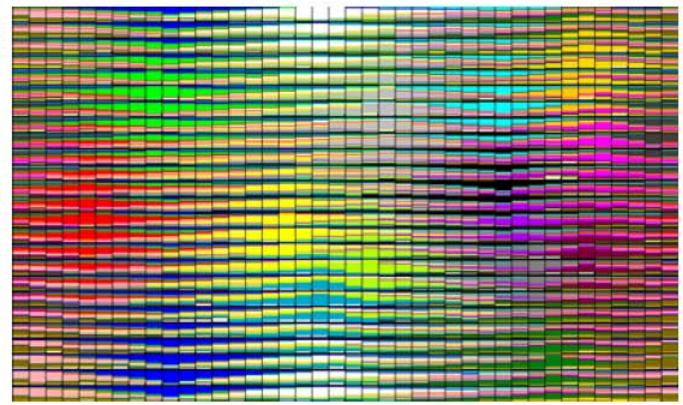


Simulations - Data Distributions

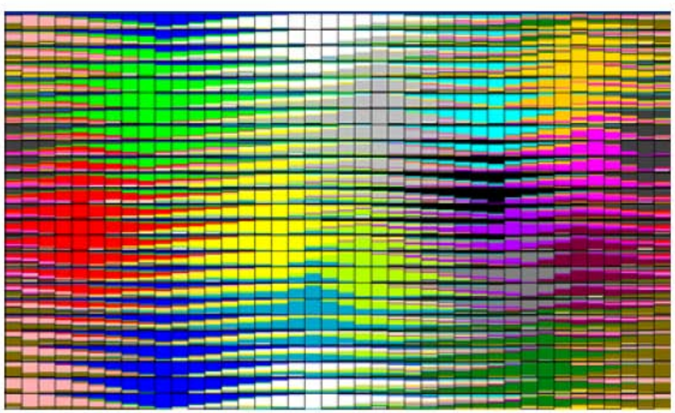
- Each node has a fixed number of data points (100).
- Each data point belongs to a category (colour).
- Data points are assigned to nodes from uniformly at random (a) to locality-dependent allocation (d).



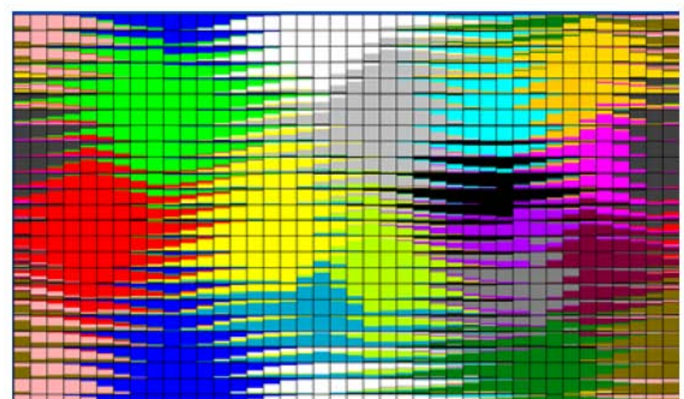
(a) $\lambda = 0$ (J=0.9936)



(b) $\lambda = 2$ (J=0.4068)



(c) $\lambda = 3$ (J=0.2297)

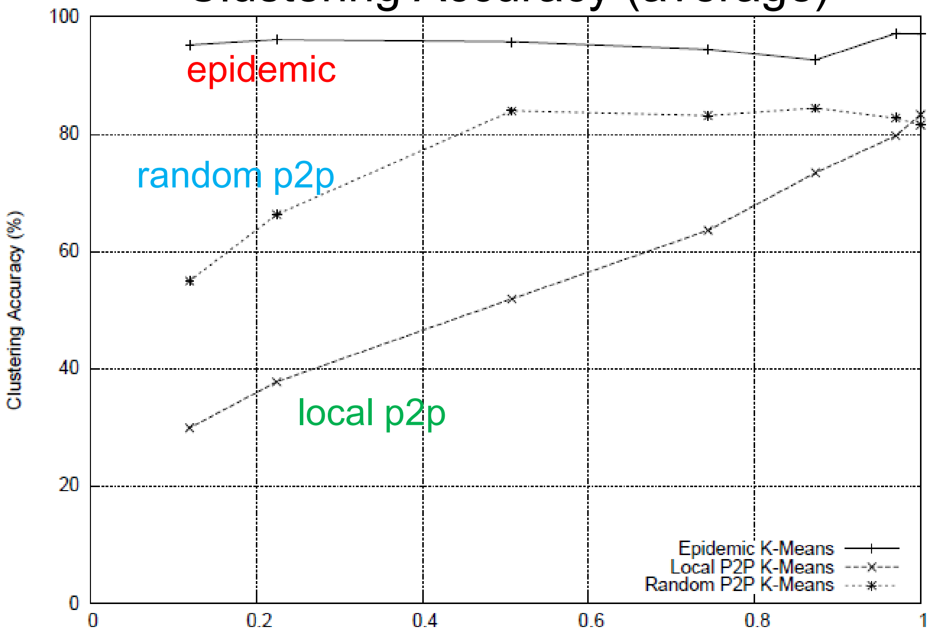


(d) $\lambda = 5$ (J=0.1181)

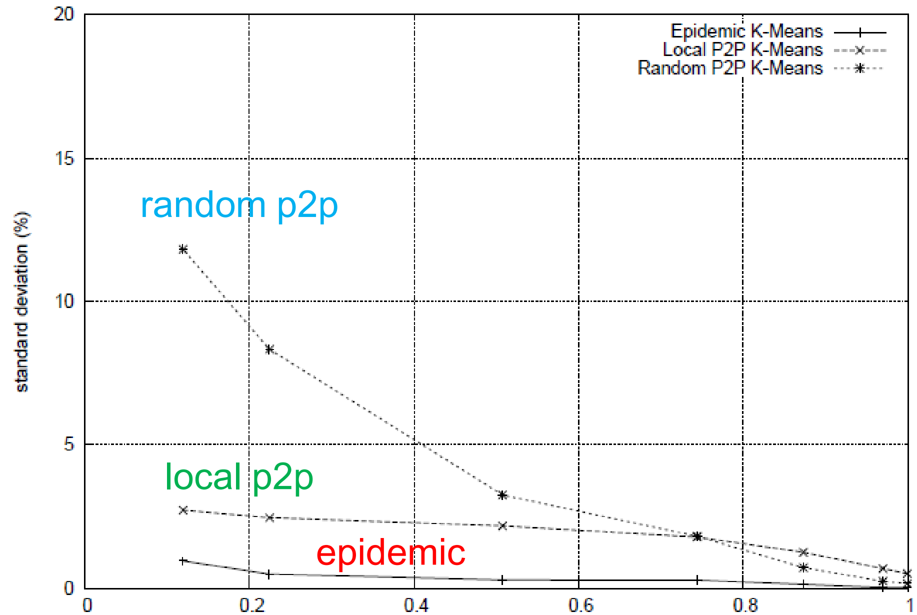
Clustering Accuracy

- Accuracy w.r.t. the "ideal" (centralised) data clustering

Clustering Accuracy (average)

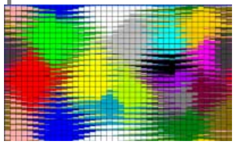


Standard Deviation

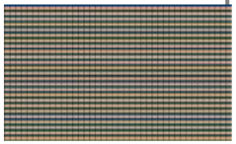


Cluster distribution (Jain Index)

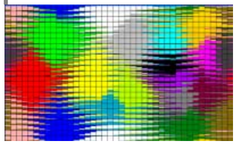
Cluster distribution (Jain Index)



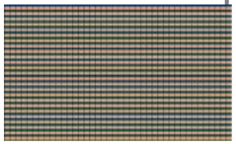
skewed data distribution



uniform distribution



skewed data distribution

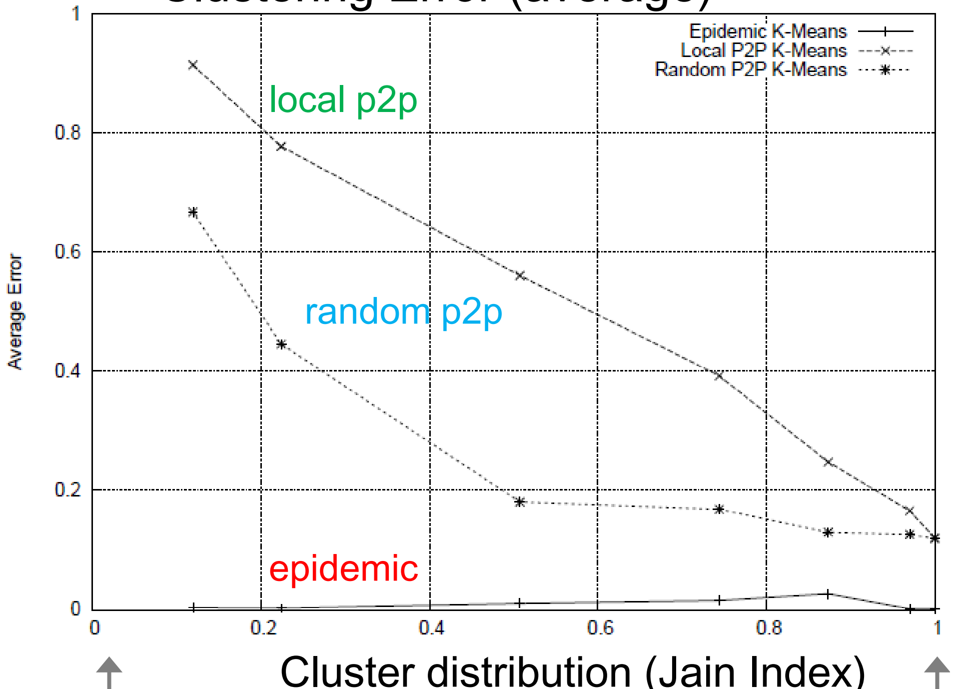


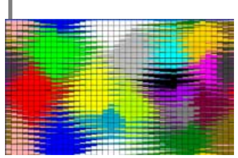
uniform distribution

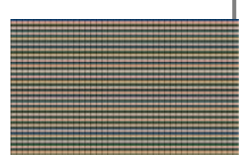
Mean Square Error of Centroids

- Error w.r.t. the "ideal" (centralised) centroids

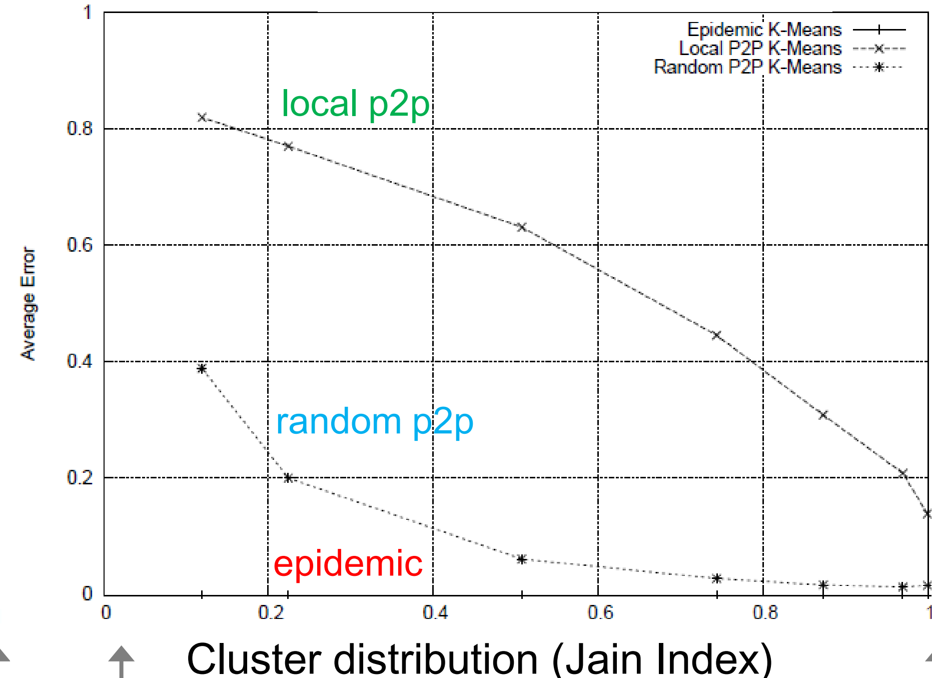
Clustering Error (average)

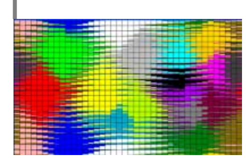


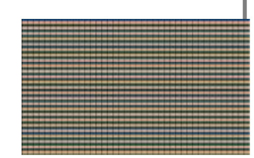

skewed data distribution


uniform distribution

Standard Deviation




skewed data distribution


uniform distribution

Conclusions

University of Reading

- Is P2P in decline?
  - Yes, file sharing P2P is in relative decline.
  - No, the P2P paradigm is no longer identified with “file sharing”.

- Epidemic or Gossip protocols are a bio-inspired paradigm for communication and computation in large-scale distributed systems
  - scalability: do not rely on central coordination, nor in deterministic overlay networks
  - global vs. total knowledge: values of aggregate functions more important than individual data

- Information Dissemination and Aggregation have been studied extensively. Their practical applicability to complex applications is only beginning to be shown.
  - Epidemic K-Means Clustering

- Open issues and research directions
  - Bootstrap, synchronisation and termination
  - Self-stabilisation: with massive distribution comes massive instability

References

University of Reading

- Mathematical models of Epidemics
  - Nicholas C. Grassly & Christophe Fraser, "Mathematical models of infectious disease transmission, Nature Reviews Microbiology 6, 477-487 (June 2008)

- Gossip-based protocols for information dissemination:
  - A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry, Epidemic algorithms for replicated database maintenance, in: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing, PODC '87, ACM, **1987**, pp. 1–12.
  - R. Karp, C. Schindelhauer, S. Shenker, B. Vocking, Randomized rumor spreading, in: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, IEEE Computer Society, **2000**, pp. 565–.
  - Eugster, P.T.; Guerraoui, R.; Kermarrec, A.-M.; Massoulie, L.; , "Epidemic information dissemination in distributed systems," Computer , vol.37, no.5, pp. 60- 67, May **2004**.

- Gossip protocols for the data aggregation problem:
  - D. Kempe, A. Dobra, J. Gehrke, Gossip-based computation of aggregate information, in: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, **2003**, pp. 482 – 491.
  - M. Jelasity, A. Montresor, O. Babaoglu, Gossip-based aggregation in large dynamic networks, ACM Transactions on Computer Systems 23, **2005**, 219–252.
  - S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Randomized gossip algorithms, Information Theory, IEEE Transactions on 52 (6), **2006**, 2508 – 2530.
  - **F. Blasa, S. Cafiero, G. Fortino, G. Di Fatta, "Symmetric Push-Sum Protocol for Decentralised Aggregation", The International Conference on Advances in P2P Systems (AP2PS), Lisbon, Portugal, Nov. 20-25, 2011.**

- Gossip-based protocols surveys and general studies:
  - Samir Khuller, Yoo-Ah Kim, and Yung-Chun Wan, "On generalized gossiping and broadcasting", Journal of Algorithms, 59, 2, May **2006**, 81-106.
  - "Dependability in aggregation by averaging," P. Jesus, C. Baquero, and P. Almeida, 1st Symposium on Informatics (INForum 2009), Sept. **2009**, pp. 482–491.
  - Rafik Makhloufi, Gregory Bonnet, Guillaume Doyen, and Dominique Gaiti, "Decentralized Aggregation Protocols in Peer-to-Peer Networks: A Survey", The 4th IEEE International Workshop on Modelling Autonomic Communications Environments (MACE), **2009**.
  - P. Jesus, C. Baquero, and P. Almeida, "Dependability in aggregation by averaging", 1st Symposium on Informatics (INForum 2009), Sept. 2009, pp. 482–491.

References

- Parallel and Distributed K-Means Clustering:

  - I. S. Dhillon and D. S. Modha, “A data-clustering algorithm on distributed memory multiprocessors,” Workshop on Large-Scale Parallel KDD Systems, pp. 245–260, Mar. **2000**.
  - S. Datta, C. Giannella, and H. Kargupta, "K-means clustering over a large, dynamic network", in Proceedings of the Sixth SIAM International Conference on Data Mining, Bethesda, Maryland, USA, **2006**, pp. 153–164.
  - S. Datta, C. Giannella, and H. Kargupta, "Approximate distributed k-means clustering over a peer-to-peer network", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 10, pp. 1372–1388, **2009**.
  - **G. Di Fatta, F. Blasa, S. Cafiero, G. Fortino, "Epidemic K-Means Clustering", IEEE ICDM Workshop on Knowledge Discovery Using Cloud and Distributed Computing Platforms (KDCloud), Vancouver, Canada, 11 Dec. 2011.**