SP 5: Biologically Inspired Techniques for "Organic IT"

Final Year Report

Participants

UniBO, UPF, Telenor, RAL

Lead partner: Bologna (UniBO)

SP5 Initial Goals

- Identify desirable, life-like properties of large-scale information systems.
- Investigate results from experimental and theoretical biology explaining the mechanisms underlying these properties, identifying areas with potential engineering applications.
- Design algorithms for specific information system functions such as network design and optimization, self-repair and management, information sharing and search, routing.
- Test algorithms in simulation environments.
- Investigate optimal strategies for collaboration between human engineers and artificial systems during the design process

SP5 Overall Achievements

- Application and development of “group selection” approach
  - Copy and re-wire algorithm in P2P to reduce free riding
  - Applied to various simulated P2P domains
  - Towards a “design pattern” or “design approach” for info. systems.
- Empirical analysis and modelling of software development
  - Open source code evolution
  - Open source developer social network evolution
- Fundamental evolutionary theory of complex networks
  - Emergence of modularity
  - evolutionary tinkering and degeneracy

SP5 wider academic impact

- Involved in organising several workshops and conferences promoting the self-* / bio- socio-inspired approach (self-*, esoa, ece, sic, mabs, css-tw1, saso)
- SASO2007, a new IEEE conference was a significant success organised at MIT in July. Next conference in late 2008, Venice
- Many invited talks

SP5 activities in 2007

- Empirical analysis and modelling of non-local evolution of open source programmer e-mail networks (D5.2.5)
- Applications (D5.3.2):
  - Group selection inspired secure gossip P2P sampling service
  - Towards an improved Bittorrent
  - Firefly-inspired synchronisation in P2P overlay networks
- Multi-level selection (D5.4.3):
  - Towards a group selection design pattern
  - Multi-level structures in software and networks

Identifying Malicious Peers Before It's Too Late:

A Decentralized Secure Peer Sampling Service

Jesi, G., Hales, D., van Steen, M. (2007)

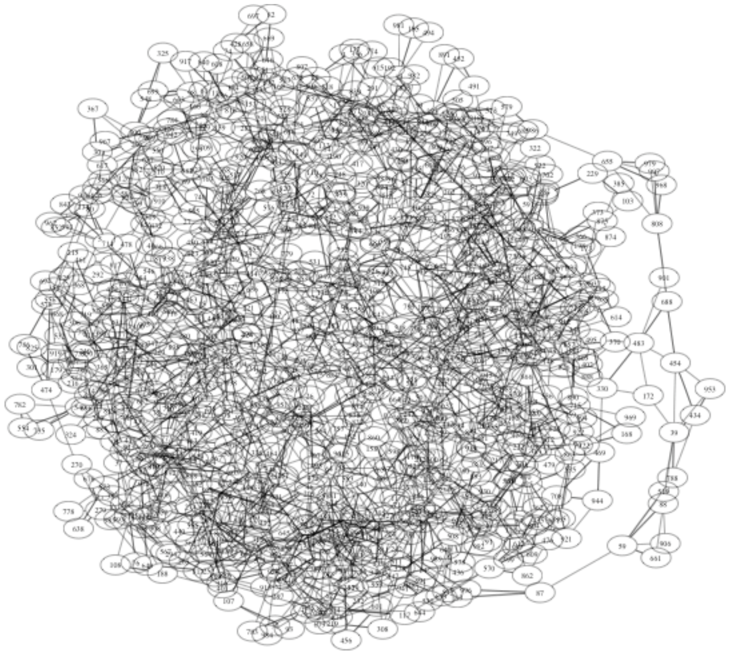
Secure Peer Sampling Service

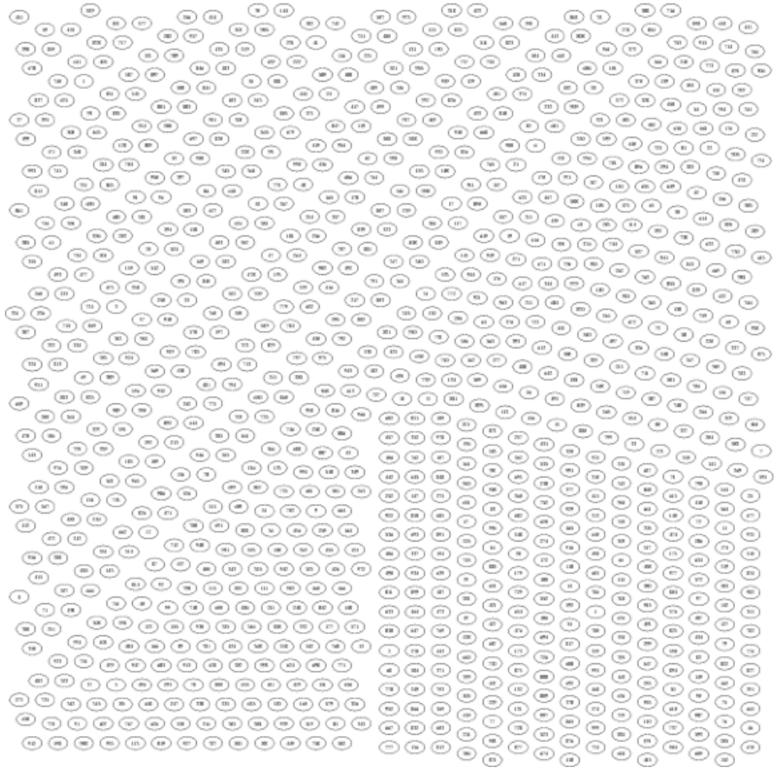
- Many proposed P2P protocols rely on a peer sampling service (PSS)
- The PSS provides a random node from the entire network
- One decentralised PSS method is to use Gossiping
- Nodes maintain a bounded list or view containing links to other nodes
- Periodically a random link is chosen and both nodes exchange view information
- It turns out that such an approach can give good results approximating random sampling and keeping the network connected
- But the approach is vulnerable to certain kinds of malicious attacks
- One such attack, termed here the “hub attack”, we attempted to address

Secure Peer Sampling Service

- Hub attack involves some set of colluding nodes always gossiping their own ID’s only
- This causes a rapid spread of only those node links to all nodes in the network - we say their views become “polluted”
- At this point all non-malicious nodes are cut-off from each other
- The malicious nodes may then leave the network leaving it totally disconnected with no way to recover
- Hence the hub attack hijacks the speed of the Gossip approach to defeat the network

Secure Peer Sampling Service

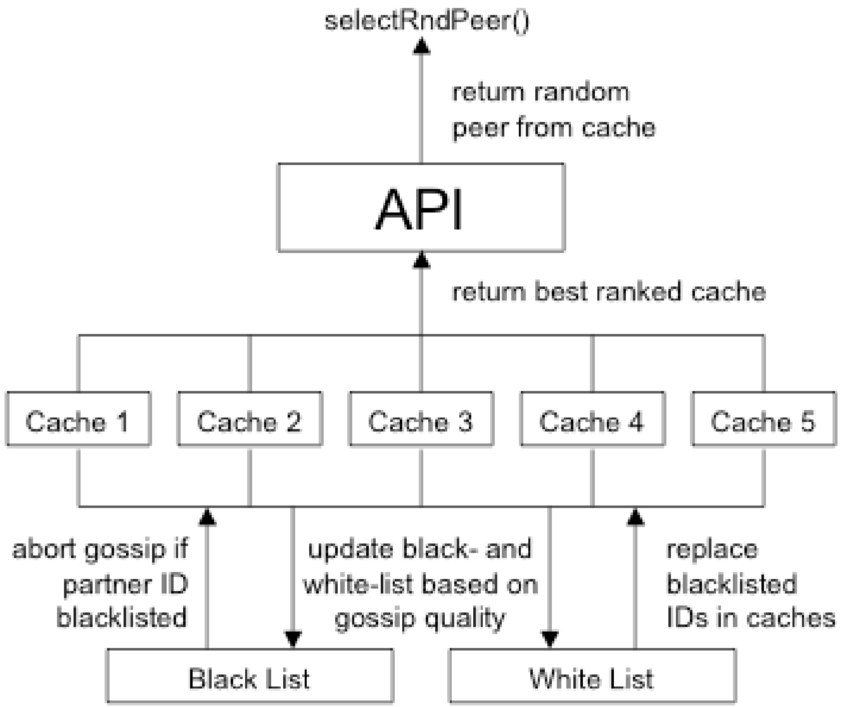


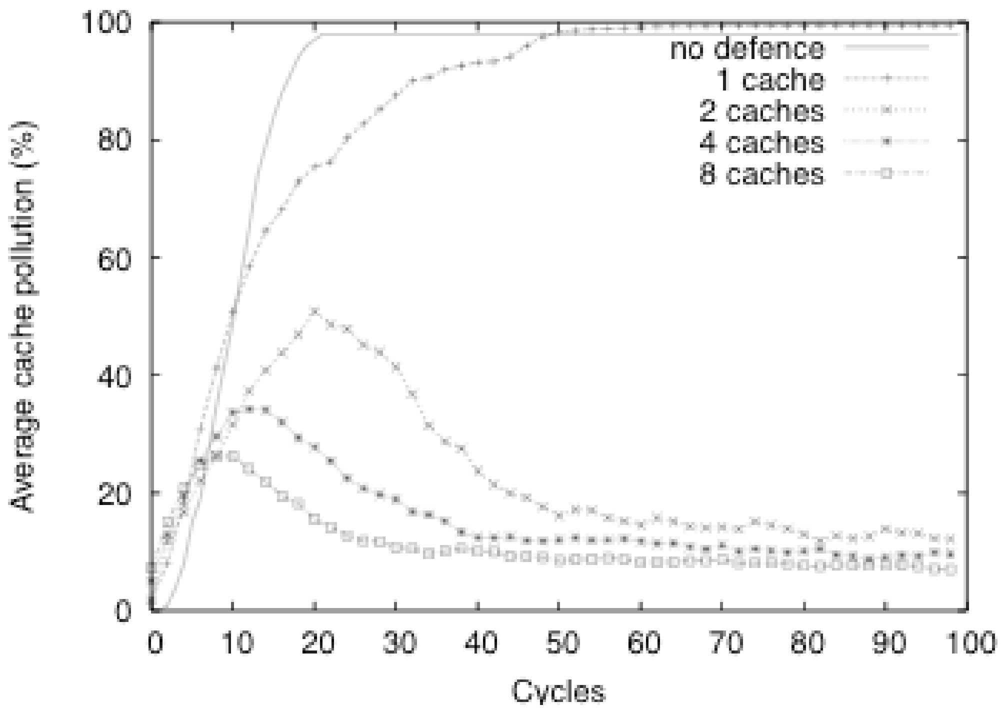


Secure Peer Sampling Service

- Solution: Maintain multiple independent views in each node
- During a gossip exchange measure similarity of exchanged views
- With probability equal to proportion of identical nodes in two views reject the gossip exchange and blacklist the node
- Otherwise, whitelist the node and accept the exchange
- Apply an aging policy to to both white and black lists
- When supplying a random peer to API select the current “best” view

Secure Peer Sampling Service





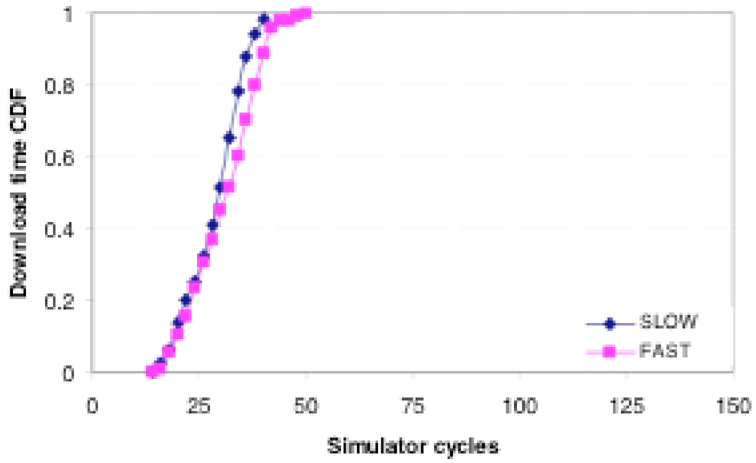
Multi-Swarm Dynamics for Fairness in BitTorrent.

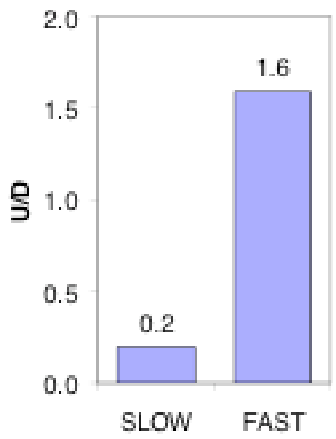
Picconi, F., Arteconi, S. (forthcoming)

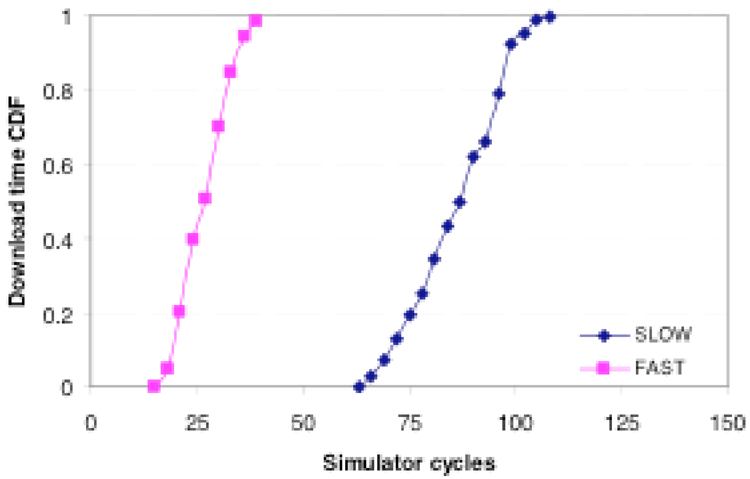
Improved BitTorrent

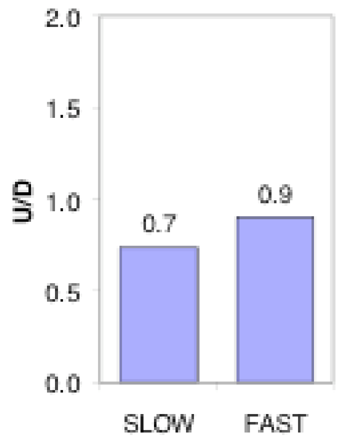
- BitTorrent uses a form of tit-for-tat to share files in a distributed way
- A shared file is distributed via a BitTorrent P2P swarm
- This enforces some degree of fairness
- But slow or hacked selfish clients (e.g. BitTyrant - Washington, BitTheif - ETH) exploit fast good guys
- By harnessing multi-swarm dynamics it should be possible to limit this form of exploitation
- Nodes stay in swarms that offer “fair” returns and leave swarms that offer “unfair” returns

Improved BitTorrent









Towards a Group Selection Design Pattern

Hales, D., Arteconi, S., Marcozzi, A., Chao, I. (2007)

Group Selection Design Pattern

- A number of novel “group selection” models are coming from theoretical biology and computational social science
- We gave initial work towards a “group selection” design pattern or approach for creating cooperative distributed systems
- We presented a number of previous simulation models that use the approach in the form of a standard design template:
  - Tag, Filesharing, Grid VO’s, Broadcasting, Content replication
- There are still open issues

Group Selection Design Pattern

- Assumptions:
  - A system is composed of individual entities that can benefit from interaction with other entities
  - The population of entities is partitioned into groups such that interaction is mainly limited to entities within the same group
  - Entities measure their own performance periodically producing a utility value
  - Entities may spontaneously change their behavior and group membership
  - Entities may view and copy some state of other entities
  - Entities desire to increase their performance (utility)

Group Selection Design Pattern

- Key Aspects:
  - Collective **Goal** - A desirable goal that the population of entities should attain.
  - **Group** Boundary Mechanism - How an entity can locate and communicate with in-group members.
  - Intra-Group **Interaction** - What kinds of utility effecting interactions an entity participates in with other in-group members.
  - **Utility** Calculation Metric - How an entity calculates a utility value based on its individual goal and in-group interactions.
  - Group **Migration** Mechanism - How migration between groups is performed.

Group Selection Design Pattern

- Emergent Process:
  - Entities are grouped in some initially arbitrary way
  - Interactions between entities within groups determine entity utilities
  - Based on utility comparisons between entities, and possibly randomized change, group memberships and interaction behavior (strategy) change over time
  - Groups which produce high utility for their members tend to grow and persist as entities join
  - Groups which produce low utility for their members tend to disperse as entities leave
  - Hence group beneficial behavior tends to be selected

Group Selection Design Pattern

Collective Goal	Maximise the total number of queries served by harnessing unused capacity in underloaded nodes.
Entity	Peer node - a node in a peer-to-peer overlay network with the ability to receive and serve queries, for a content item, from clients external to the overlay network. Each node has a maximum capaciy limiting number of queries serviceable over a time period. Each node can be thought of as a web server, for example, and stores is own content item and a replicated copy of each of its neighbours content items.
Group	The neighbour list (or view) of a node defines its group
Interaction	Receiving redirected queries from overloaded nodes or conversely redirecting queries to a random neighbour when overloaded. When a node makes a connection to a new neighbour both nodes mutually replicated their contents.
Utility	A simple binary satisfaction function: if all queries received by a node are eventually served then the node is satisfied otherwise it is unsatisfied.
Migration	Periodically, unsatisfied nodes move randomly in the network. But a node will only accept an incoming connection from a moving node if it is in a receptive state. A node is only receptive if it has spare capacity or is itself unsatisfied.

Figure 19. Key aspects for the CacheWorld model.

Empirical analysis and modelling of non-local evolution of open source programmer e-mail networks

(D5.2.5)

Fini!