Design Of An Efficient Search Algorithm For P2P Networks Using Concepts From Natural Immune Systems

Niloy Ganguly, Andreas Deutsch
Center for High Performance Computing, Dresden University of Technology, Germany

Geoff Canright
Telenor Research and Development, 1331 Fornebu, Norway

Peer-to-peer Networks

Salient Features:

- Totally decentralized networks unlike traditional client server architecture.
- Direct connection between peers.
- Unlike traditional Internet where peers are mere recipients of information, in these systems, peers are also providers of information.
- Popular systems - Napster, Kazaa, Freenet, Gnutella etc.

THE PROBLEM - SEARCH IN PEER-TO-PEER NETWORKS

Present Shortcomings

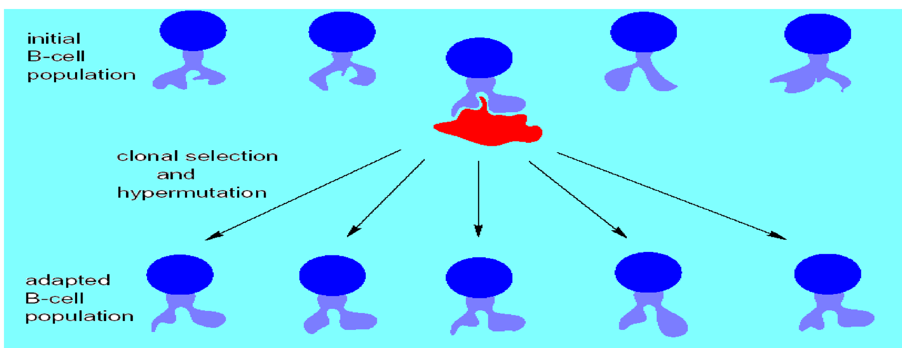
- Searching for content in peer-to-peer networks is very slow.
- Peers are extremely transient.
- A large fraction of peers continuously enters and exits the network; consequently deteriorating the performance of any search algorithm even further.

Solution Strategy

- Develop an algorithm which will not only enhance speed and efficiency, but will also be robust to withstand the transient nature of the peers.
- Derive inspiration from the Immune System to develop such algorithm.
- The algorithm developed and discussed here is termed as *ImmuneSearch*.

Why the Immune System?

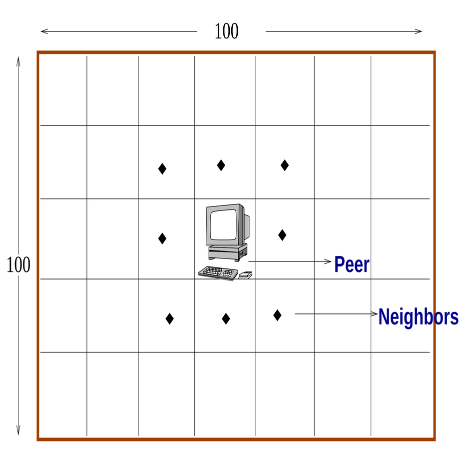
- Distributed system
- Pattern recognition
- Anomaly detection
- Noise tolerance, robustness
- Feature extraction
- Diversity
- Learning, memory
- Adaptation



ImmuneSearch Algorithm is inspired by the simple and well known concept of the humoral immune system where B cells upon external stimulation undergo proliferation and mutation to generate antibodies which track the antigens (foreign bodies).

MODELING

- Peers: 10.000, each peer carries a. information profile (P_I) b. search profile (P_S) [c. (query) message (M)]
- Overlay network: toroidal grid (100 x 100)
- Each node hosts a peer
- Profile & message is represented by a 10-bit binary string
- Distribution of profiles according to Zipf's law
- Profile/message affinity: denoted by sim(M,Px) = [10 - HD(M,Px)], HD: Hamming distance



Algorithm - ImmuneSearch

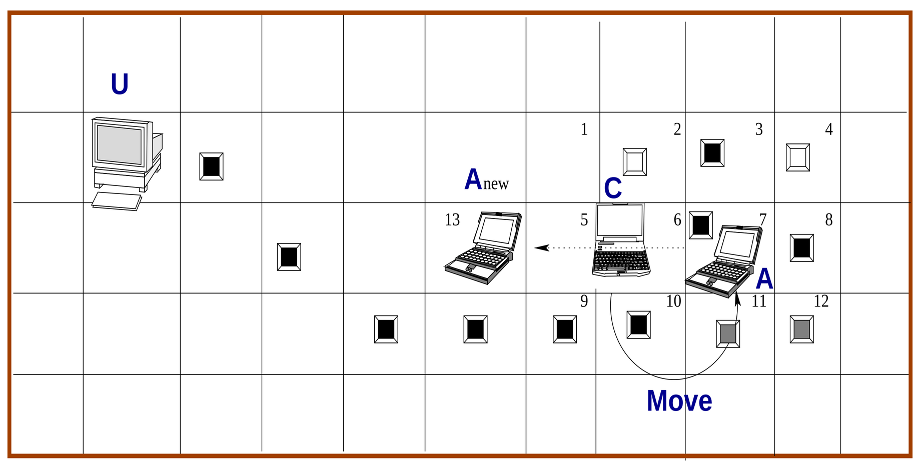
Salient Features

- Distributed algorithm.
- Each node executes the algorithm independently.
- Query messages are originated from the user (U) who initiates the search.
- A node (A) runs the algorithm when it encounters a query message.

Algorithm 1 *ImmuneSearch(A)*
Input : Message packet (M)
Output : Search Result
if ($sim(P_I,M) \geq T(Search)$) */*T(search) - Similarity threshold required to declare a successful */*
  Output (*Successful Search*)
Topology_Evolution(A)
Reaction_p2p(A)

Algorithm 2 *Topology_Evolution(A)*
Input : Message packet(M)
If ([$sim(P_I,M)$ *or* $sim(P_S,M)$] $\geq$ $T(Evol)$)
  / T(Evol) - Similarity threshold required to perform topology evolution */*
  Move peer closer to user node U
  */*Amount of movement depends on distance between A and U and amount of* $sim(P_i,M)$**/*

Algorithm 3 *Reaction_p2p(A)*
Input : Message packet(M)
If ($sim(P_I,M) \geq T(Pro/Mut)$)
  */*T(Pro/Mut) - Similarity threshold required to launch proliferation/mutation */*
  Proliferate the packet M in neighborhood
  Mutate some of the proliferated packets
else
  Send the packet M to a neighbor peer



- Node U initiates the search.
- Subsequently, packets from U reach A.
- A has matching profile as U, therefore the packets proliferate and mutate.
- A is pushed 'closer' to U.
- A is brought closer to U by changing its neighborhood configuration.

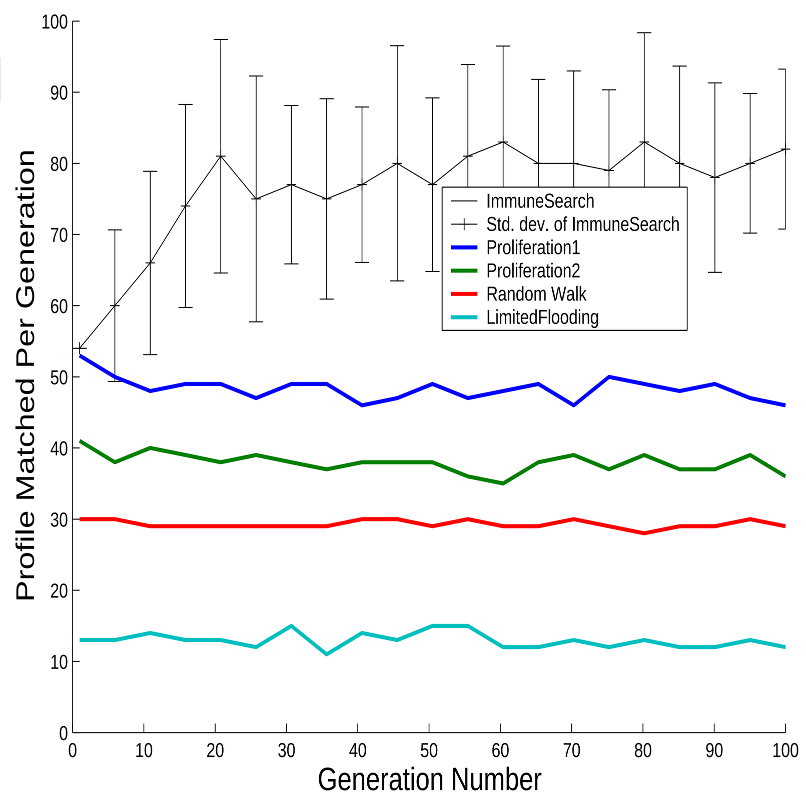
SIMULATIONS

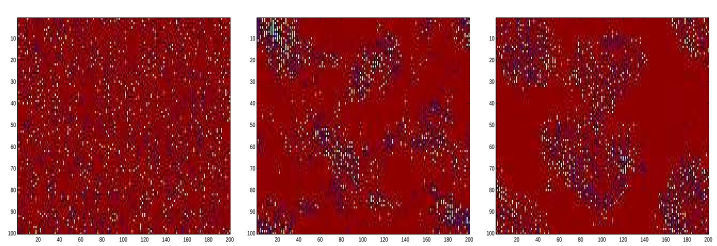
- Two types of experiments - (a). Stable conditions - peers don't leave the network; (b). Transient conditions - peers constantly enter/leave the network.
- Measurement is done in terms of generations (one generation: 100 searches).
- Each search is initiated by a peer residing at a randomly chosen node and the number of search items (n_s) found within 50 time steps from the commencement of the search is calculated.
- The search output (n_s) is averaged over 100 different searches (a generation), whereby we obtain N_s, where $N_s = \frac{\sum_{i=1}^{100} n_s}{100}$. The value of N_s directly reflects the efficiency of the network.
- Comparison: *Random walk, limited flooding*, two simple proliferation/mutation schemes without any topology evolution - $proliferation_1$, $proliferation_2$, *ImmuneSearch*.
- $proliferation_1$ and *ImmuneSearch* have the same proliferation/mutation rate while $proliferation_2$ has higher.
- To ensure fair comparison, care is taken to ensure that each of the above mentioned processes participates in the network with equal power - that is with equal number of message packets.

Experiments in stable conditions

- Each of the processes is run for 100 generations and the result shown through the graph is the average of 20 such runs.
- x-axis of the graph shows the generation number while the y-axis represents the average number of search items (N_s) found in the last 100 searches.
- Standard deviation of the results is shown for *ImmuneSearch* algorithm. The standard deviation of the output is roughly around 10% of mean in each case.
- It is seen that the number of search items (N_s) found is progressively higher in limited flooding, random walk, $proliferation_2$, $proliferation_1$, *ImmuneSearch* respectively.
- In *ImmuneSearch*, the first 25 generations can be termed as 'learning' phase. During this time, similar to natural immune system, the p2p network *develops memory* by repositioning the peers.
- Repositioning of the peers results in cluster formation of similar peers whereby the search efficiency increases.

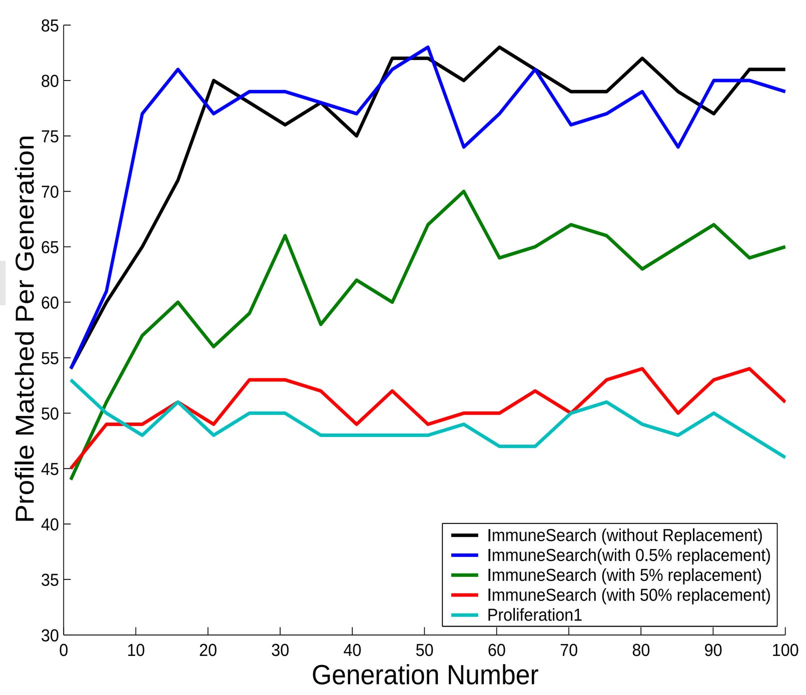
The three figures show the clustering of peers (peers possessing most frequent tokens) in the p2p network, the snapshots are taken at generation no. 0, 3, 24 respectively.





Experiments in transient conditions

- **Experiment**: Change 0.5%, 1%, 5%, 50% of peers at each generation.
- Each of the processes is run for 100 generations and the result shown through the graph is the average of 20 such runs.
- x-axis of the graph shows the generation number while the y-axis represents the average number of search items (N_s) found in the last 100 searches.
- **Observations**:
  - *ImmuneSearch* is better till 50% replacement than simple proliferation.
  - 0.5% replacement is mostly better than without replacement scheme. Thus, a little transient is a boon rather than a bane.



DISCUSSION AND OUTLOOK

The above design, which draws its motivation from natural immune systems, typically highlights the spirit of BISON*. The beauty of the algorithm lies in its simplicity. However, this simple decentralized algorithm generates emergent properties like a *complex adaptive system*. Consequently, it provides a robust solution for the important search problem. The basic strengths displayed by the *ImmuneSearch* algorithm need to be further explored and developed, by applying it in more realistic circumstances in the near future.

*This work was partially supported by the Future & Emerging Technologies unit of the European Commission through Project BISON - *Biology-Inspired techniques for Self-Organization in dynamic Networks* (IST-2001-38923).

http://rcswww.urz.tu-dresden.de/~ganguly