A distributed system consists of, possibly heterogeneous, computing nodes connected by communication network that do not share memory or clock. One of the main benefits of distributed systems is resource sharing which speeds up computation, enhances data availability and reliability. However resources must be discovered and allocated before they can be shared. Virtual caching is a new caching scheme which allows a host node to grant authority of caching pages in some fraction of its own cache to nearby nodes. However the virtual caching protocol doesn't mention how a client node obtains virtual cache from remote host. To address this problem we formulate a resource discovery and allocation problem. We are focusing our attention on how to locate resources-surplus donor nodes and to determine how much of the request for resources of deficient nodes will be satisfied, efficiently in a connected network especially within a finite hop of the resource deficient node. We intend to
minimize the amount of unfulfilled request of deficient nodes. Virtual cache allocation can be changed any time depending upon the requirement. Hence the proposed heuristics are efficient both in terms of time and amount of communication performed.

We also estimate the quality of distribution achieved by comparing the distribution yielded by the heuristics and by the solution of ILP formulation of the problem. We propose and compare a few heuristics for minimizing the amount of unfulfilled request for resources, of deficient nodes when nodes look for resources within finite hops. In this paper we are restricting ourselves to single hop only. For the bounded hops we restrict ourselves to the resource distribution within one hop. By using non-anonymous arbitrary topology with sequence number of request to resolve deadlocks and distributing resources over the original arbitrary network. Sequence number of the request is the unique ID of sender node. We proposed a heuristic to distribute resources over anonymous arbitrary topology by passing a token. The token is privilege to distribute the resources. Each resource - surplus node is giving its extra nodes in such a way so that it itself doesn't becomes resource-deficient in the process. Load is not infinitely divisible. We are focusing our attention only to determine how much of the request of each resource - deficient node will be satisfied.

Reference

Index Terms

Computer Science

Distributed Systems

Key words

donor
deficient
token
caching
optimal