An Approach To High Performance Distributed Web Brokering

Prof. Geoffrey Fox –Director *Community Grid Labs, Indiana University*.
Shrideep Pallickara, PhD – *Department of Electrical Engineering & Computer Science*.

Web services proliferating the Internet can be abstracted as a distributed system of "clients" comprising "users", "resources" or proxies thereto. These web services are usually hosted on a broker or a network of brokers, where the term server is avoided to distinguish it clearly from application servers. Clients access web services by connecting to a broker, where the client's online presence is neither time constrained nor is it tied to a specific device, geographic location or communication channel. Clients then maintain active connections to the hosting broker throughout the duration that they use the service.

The single broker approach albeit a simple model constitutes a single point of failure, a problem compounded by the model's inability to scale. Scaling the service by employing a distributed broker network provides the base for a highly available solution resilient to single/multiple broker failures. However, efficient service solutions in distributed settings are littered with pitfalls. In the case of a distributed network of brokers providing a service, brokers can be added to the system simply by instantiating the broker process and initiating a connection to one of the brokers within the broker network. Devolving control over the modifications to the network fabric lead to scenarios where broker/connection instantiations result in the network being susceptible to network partitions, poor bandwidth utilizations and inefficient routing strategies. To deal with failures most systems implicitly invoke the "finite time recovery" (FTR) constraint requiring a failed broker to recover within a finite amount of time. FTR implies brokers have state and force every broker to be up and running at all times. Recovery involves state reconstruction from brokers that the recovering broker had maintained active connections to prior to the failure. This scheme breaks down during multiple neighboring broker failures. Stalling operations for certain sections of the network, and denying service to clients while waiting for failed processes to recover could result in prolonged, probably interminable waits. Uncontrolled settings, resulting in a broker network devoid of any logical structure make creation of abbreviated system views (ASV) an arduous if not impossible task. The lack of this knowledge hampers development of efficient routing strategies, which exploit the broker topology. Such systems then resort to "flooding" the entire broker network with service nuggets (events), forcing brokers to discard events that clients maintaining active connections to it are not interested in.

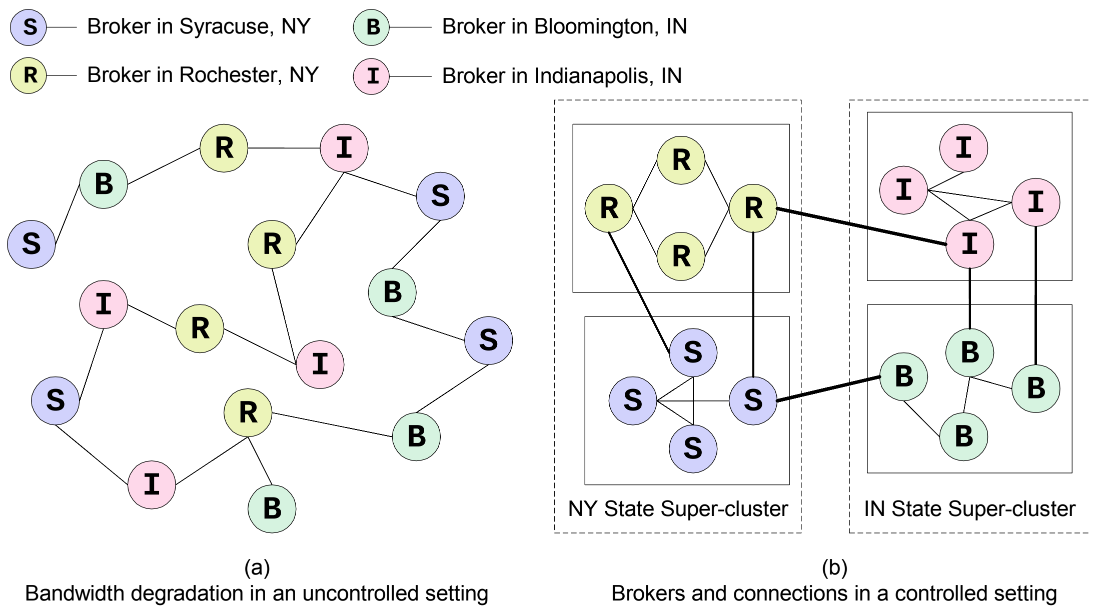
S — Broker in Syracuse, NY
R — Broker in Rochester, NY
B — Broker in Bloomington, IN
I — Broker in Indianapolis, IN
NY State Super-cluster
IN State Super-cluster
(a)
Bandwidth degradation in an uncontrolled setting
(b)
Brokers and connections in a controlled setting

Figure 1: Benefits of a controlled setting

We impose a hierarchical structure on the broker network, where a broker is part of a cluster that is part of a super-cluster, which in turn is part of a super-super-cluster and so on. Clusters comprise strongly connected brokers with multiple links to brokers in other clusters, ensuring alternate communication routes during failures. Broker additions are controlled by protocols sensitive to IP-addresses, geographical location and cluster size. This organization scheme results in "small world networks" where average communication pathlengths between brokers increase logarithmically with geometric increases in network size as opposed to exponential increases in uncontrolled settings. Creation of ASVs and the detection of network partitions are easily achieved in this topology.

Every broker keeps track of connections that exist between - nodes in its cluster, clusters in its super-cluster and so on. This is achieved by controlling the horizon associated with the information pertaining to connections between any two brokers within the system. Thus, information pertaining to connections between two brokers within the same cluster is disseminated only within the corresponding cluster. Similarly information pertaining to connections between brokers in different clusters within the same super-cluster is not propagated beyond the realms of the super-cluster. Identical constraints are imposed on propagating connection losses. ASVs are maintained in a graph, with "graph-nodes" corresponding to units (brokers, clusters etc.) and "edges" corresponding to the connections between the units. Link counts associated with the edges reflect the number of alternate routes available between units maintaining active connections to each other. An edge is eliminated when the link count reduces to zero. This plays an important role in determining if a connection loss would lead to partitions. Also associated with every edge is the traversal-premium, a cost scheme that can be dynamically updated to reflect changes in link behavior with the passage of time. ASVs are different at each broker and provide a consistent overall view of the network while allowing for the calculation of shortest/fastest paths to reach any set of destinations within the broker network.

Every broker serves in two capacities, the first is of course as a conduit for clients to interact with the service. The second role is that of a node within a vast broker network, responsible for maintaining network snapshots and making intelligent decisions aiding efficient disseminations. Every event has an implicit or explicit destination list, comprising clients, associated with it. The system as a whole is responsible for computing broker destinations (targets) and ensuring efficient delivery to targeted brokers. Each targeted broker in its dual role is also responsible for routing events to interested clients attached to it. Events as they pass through the broker network are be updated to snapshot its dissemination within the network, which in tandem with ASV at each broker is used to deploy a near optimal routing solution. The routing is near optimal since for every event the associated targeted set of brokers are usually the only ones involved in disseminations. Further every broker, either targeted or en route to one, computes the shortest path to reach target destinations while employing only those links and brokers that have not failed/failure-suspected. The dissemination information associated with the events eliminates continuous echoing of events in the system. Stable storages existing in parts of the system are responsible for introducing state into the events. The arrival of events at clients advances the state associated with the corresponding clients. Brokers do not keep track of this state and are responsible for ensuring the most efficient routing. Since the brokers are stateless we eliminate FTR.

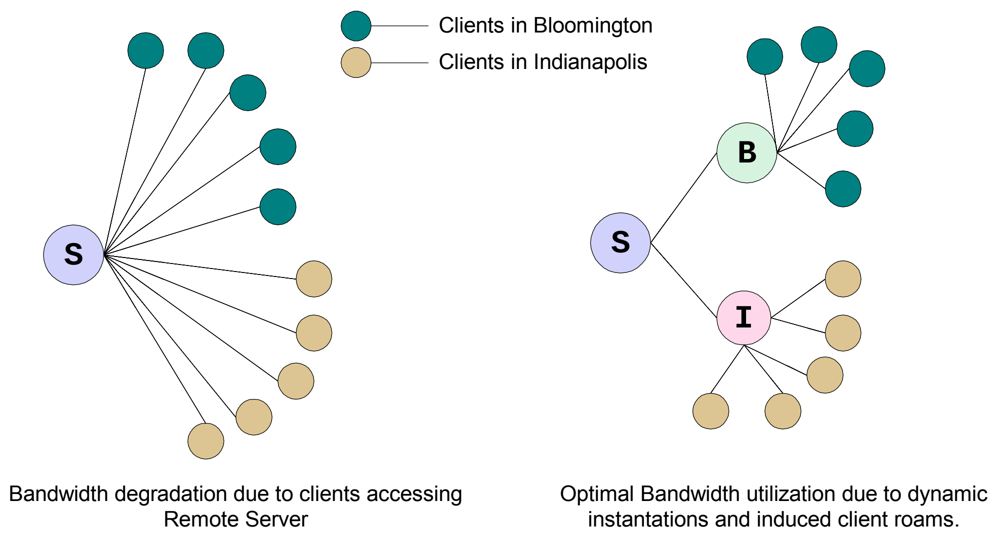
Clients in Bloomington
Clients in Indianapolis
S
S
B
I
Bandwidth degradation due to clients accessing Remote Server
Optimal Bandwidth utilization due to dynamic instantations and induced client roams.

Figure 2: Simple example demonstrating dynamic topologies

Clients can "roam" and attach themselves to any broker within the system. Clients roam during broker failures and are also induced to roam by brokers that have scheduled downtimes. Clients reconnecting after prolonged disconnects, connect to the local broker instead of the remote broker that it was last attached to. This eliminates bandwidth degradations caused by heavy concentration of clients from disparate geographic locations accessing a certain known remote broker over and over again. Support for local broker accesses and client roams, along with elimination of FTR provide an environment extremely conducive to dynamic topologies. Brokers and connections could be instantiated dynamically to ensure the optimal bandwidth utilizations. These brokers and connections are added to the network fabric in accordance with rules that are dictated by the agents responsible for broker organization. Brokers and connections between brokers can be dynamically instantiated based on the concentration of clients at a geographic location and also based on the content that these clients are interested in. Similarly average pathlengths for communication could be reduced by instantiating connections to optimize clustering coefficients within the broker network. Brokers can be continuously added or fail and the broker network can undulate with these additions and failures of brokers. Clients could then be induced to roam to such dynamically created brokers for optimizing bandwidth utilization and the service can be utilized uninterrupted.