

Skipnet paper

A datastructure such as a dictionary or an array normally belongs to a single executing program. It is often desirable to be able to easily increase the availability of a single data structure to multiple processes and multiple machines. This paper explores the skipnet algorithm which is a type of scalable data structure that allows for growing from one process to many processes over many machines.

The skipnet algorithm provides an interface to store key/value pairs. As the data structure grows to multiple machines, the data storage and processor resources for the skipnet data structure also grows. This allows for disk and cpu resources to be used in aggregate as the size of the datastructure grows. Distributing the load also allows for efficiencies over standard client/server paradigms. As data is requested frequently, the storage for that data can be migrated closer to the requesting machine. It also allows for redundancy of storage which trades reliability for speed.

Two important attributes of a skipnet are content locality and routing locality. Content locality means being able to optionally specify which computer holds which pieces of the data structure. This is important when a single skipnet spans multiple organizations. Sometimes for security reasons or cost reasons the available computers for storing a particular piece of data needs to be limited to a specific group of computers. For example, coke.com and pepsi.com may share data in the same skipnet, but pepsi.com may have sensitive data that they do not want kept on any coke.com computers. Routing locality has the same motivation - as sensitive data is moved between pepsi.com computers, a skipnet can guarantee that the route does not include any computers outside of pepsi.com.