Tapestry: A Resilient Global-Scale Overlay for Service Deployment Reading Summary

Saige Liu, CSC 466,Feb 10, 2018

At the time of paper publication, most of the underlying infrastructure's ability on fixing failures and changes is restricted. The classic way of setting up new application needs to match with old infrastructures or establish new Internet Protocols. The scalable and fault tolerant infrastructure is the most efficient and essential.

To begin with, this paper introduced Tapestry, an extensible P2P overlay routing infrastructure, which uses soft state routing application programming interface to preserve fault tolerance. Tapestry also provides decentralized object location and routing(DOLR) which uses only local resources for location-independent routing to nearby object replica. Compared to first generation of P2P system, Tapestry is structured and achieved a distributed hash table. Comparing with chord that ignores network distance, Tapestry creates an ideal routing table and manage it locally. Nowadays, Tapestry's applications is deployed in OceanStore, Mnemosyne, Bayeux and Spamwatch.

The algorithms of Tapestry are explained in this paper. Firstly, every node in Tapestry has a unique nodeID, with endpoints linked to globally unique identifiers(GUIDs). The DOLR includes four APIs: PublishObject, which make object available to others; UnpublishObject,which remove object from location mapping; RouteToObject, which route to the object and RouteToNode, which route to specific nodeID. Besides, Tapestry's routing mesh uses neighbor map which contains i entry in j layers. The nodes with nearest neighbour starts with prefix(N, j-1)+"i". User publish object to the root node. All the nodes that object passes by recordding its location mapping. While locating the object, each node checks the location mapping and modify the direction towards the root node.

Furthermore, paper also defined the dynamic nodes. The related nodes need to be notified when new node comes into the system. Then an optimal routing table is generated for new node and all the nearby nodes need to update their routing table. If the new node is the root, it will send a broadcast message to available nodes with the same prefix. Following the related nodes send publish object to both the node and substitute node. Tapestry uses intermittent request to check node failures, which might trigger a reparation boosted by soft-state republishing of node location. This mechanism produces a 100% success rate in the best case.

The authors explored the performance of Tapestry on various realistic condition. However, there is some concerns on Tapestry. One is security issue, for example, one of the node is hacked and its routing table leads all the other nodes to a virus software. To prevent the security issue, one of the possible solution is to attach the location mapping with certifications by using only trusted routing table from Tapestry system.The other concern is network overload. Since Tapestry updates the routing table and check availability consistently. During each node insertion, all the related nodes' mapping table need an update. If deploy Tapestry world-wide, the lasting updates might cause overload.