

CS443 Paper Review
Lei Yang
2005-5-09

Title

Manageability, availability and performance in Porcupine: a highly scalable, cluster-based mail service

Author

Yasushi Satio, Brian N. Bershad, and Henry M. Levy

Summary

This paper presents Porcupine, a scalable mail server using a large cluster of commodity PCs. Porcupine distributes sessions, data, and services homogeneously and dynamically across nodes in a cluster to ensure the manageability, availability, and performance of the system.

Most important ideas

The primary goal of Porcupine is scalability defined in terms of manageability, availability, and performance requirements. In turn, these requirements are met through combinations of four key techniques: functional homogeneity, replication, dynamic transaction scheduling, and automatic reconfiguration. In Porcupine, any node can perform any function, so called functional homogeneity. This ensures that a service is always available. Replication is used to store persistent data, or hard state data. The key data structures in Porcupine are: mailbox fragment, mailbox fragment list, user profile database, user profile soft state, user map, and cluster membership list. These data structures are distributed and maintained on each node by several managers. User database and mailbox fragments are replicated to ensure the availability. Finally, Porcupine uses dynamic load balancing to distribute the workload across nodes in the cluster in order to maximize throughput. More specifically, Porcupine clients select an initial contact node either to deliver or to retrieve mail, that contact node then uses the system's load-balancing services to select the "best" set of nodes for servicing the connection.

The Porcupine system was evaluated on a 30-node cluster. Performance, availability, and manageability were measured.

Flaws/Questions

Well, the first thing I am wondering about is, had no one else think of similar idea presented in this paper before them? The clustering solutions for mail servers seem to be quite a mature field, dynamic load balancing is nothing new. Besides, the reliability problem had been raised quite frequently.

Also, security issues (not fault tolerance) were not raised at all; caching should have been better explored to improve performance. Overall, I'd say this is a good paper in the sense that it was well written, and very clear. But I don't see why it got the best paper award in SOSP. Their perspective is very good, though; if nobody else indeed had thought of this before.

Relevance/Potential Future Research

This project is pretty much dead now. It wasn't commercialized, and is not practical for mail service providers, mostly because Porcupine is not supported, and some basic functionalities, such as spam filtering, LDAP integration, were not implemented.