Abstract

Existing and future Internet information services will provide large, rapidly evolving, highly accessed, yet autonomously managed information spaces. Internet news, perhaps, is the closest existing precursor to such services. It permits asynchronous updates, is replicated at thousands of autonomously managed sites, manages a large database, yet presents adequate response time. It gets its performance through massive replication. Other Internet services, such as archie, and WWW/Mosaic must massively replicate their data to deliver reasonable performance.

The problem of replicating information that can be partitioned into autonomously managed subspaces has well-known solutions. Naming services such as the Domain Name Service (DNS) and Grapevine use administrative boundaries to partition their hierarchical name space into *domains*. These domains need only be replicated in a handful of services for adequate performance.

On the other hand, efficient massive replication of wide-area, flat, yet autonomously managed data is yet to be demonstrated, since existing replication algorithms do not address the scale and autonomy of today's internetworks. They either ignore network topology issues entirely, or rely on system administrators to hand-configure replica topologies over which updates travel. Lampson's widely cited Global Name Service (GNS) propagates updates over manually configured topologies. However, the complexity of manually configuring fault-tolerant update topologies that use the underlying physical network efficiently increases with the scale of today's internetworks.

Furthermore, GNS-like replication mechanisms also manage single, flat groups of replicas. While this is appropriate for applications with 20 to 30 replicas that operate within single administrative boundaries, it is unrealistic for wide-area, massively replicated services whose replicas spread throughout the Internet's thousands of administrative domains.

This research investigates scalable replication mechanisms for wide-area, autonomously managed services. We propose a new architecture that extends existing replication mechanisms to explicitly address scalability and autonomy. We target replication degrees of thousands or even tens of thousands of weakly-consistent replicas.

Imitating the Internet's administrative domain routing hierarchy, the proposed architecture organizes replicas into multiple *replication groups*. Organizing replicas into multiple groups limits the size of the consistency state each replica keeps. It also preserves autonomy, since it insulates replication groups from other groups' administrative decisions.

We argue that efficient replication algorithms keep replicas weakly consistent by flooding data between them. Our architecture automatically builds a *logical update flooding topology* over which replicas propagate updates to other replicas in their group. Replicas in a group estimate the underlying physical topology. Using the estimated network topology, we compute a logical update topology that is *k-connected* for resilience, tries to use the physical network efficiently, and yet restricts update propagation delays. Replication groups compute and adopt a new update topology every time they detect changes in group membership and network topology.

We describe our architecture in detail and its implementation as a hierarchical, network cognizant replication tool, which we implemented as part of the Harvest resource discovery system. Through simulations, we investigate the benefits of using hierarchical replication groups and distributing updates over the logical update topologies our architecture computes. We present the results from the wide-area experiments we conducted to evaluate our network topology estimation strategy. We also present our logical topology calculation algorithm in detail, and the results obtained when running our algorithm over randomly generated topologies. Finally, we explore the use of internet multicast as the underlying update propagation mechanism. We argue that since having a single multicast group with all replicas of a service will not scale, each replication group can be mapped to a multicast group. Through *corner replicas*, updates in one group make their way to all groups.

Lampson's Global Name Service has been widely accepted as the solution to the problem of replicating wide-area, distributed, weakly-consistent applications. However, almost 10 years ago, when Lampson wrote "... The system I have in mind

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.