

Project Summary for ACM SIGOPS workshop on  
"Making Distributed Systems Work"

From: Jerome H. Saltzer  
Technical Director  
Project Athena  
Massachusetts Institute of Technology

Date: April 9, 1986

M.I.T. Project Athena is in the third year of a five year project to use networked personal computers in the support of undergraduate university education. The project is structured in two major phases, with the first, now installed, phase being an approximation of the future environment. The approximation consists of 50 UNIX time-sharing systems linked with local area networks.

The next phase of Project Athena is a client-server model with 1500 so-called 3M (1 Mips, 1 Megapel, and 1--actually 2 or 3--Megabyte) workstations as the clients and 50 servers providing library storage, mail, naming, and authentication services. The overall architecture for this model is now set, and the components are in various stages of design, implementation, or experimental deployment.

Although not intended as a distributed systems research project, the design, implementation, and deployment of the Athena workstation-server model is bringing into sharp focus some sometimes-fuzzy concepts current in the distributed systems research community, and at an operating scale that forces a very candid appraisal of both the real value and the state of readiness of each proposed idea. As one might expect, some mundane but real problems of large-scale distributed systems are showing up as major hurdles.

For example:

- The most useful client-server example we have so far encountered is a division of a windowing display manager into a device driver service and an application client library separated by a network connection. This architecture allows one to "pop up a window across the net," a surprisingly handy ability, not just for remote login to old-fashioned systems and for network service management, but for cooperation among workstation users.
- There is an unresolved tension between packing as much software as possible into a workstation (so that a student can take it home over the summer or can use it in an off-campus apartment) and maximizing the software found in centrally managed libraries, where update is easy to accomplish. The sometimes proposed compromise of "put it all on a CD rom and distribute a new copy quarterly" involves a technology that has been just around the corner for several years and that has significant performance problems.

- The assumption that the student might own the workstation, and could therefore tinker with the kernel, seems to be at odds with 25 years of designing operating system with a ground rule in which the supervisor/user boundary is sacred; one result is that some distributed file systems cannot be protected.
- Most candidates for distributed file systems don't perform well enough to make deployment with 1500 workstations economically feasible.
- The ease with which a mis-configured workstation can generate network chaos by misusing broadcast protocols leads to a new requirement for compartmentalization of local area networks to limit damage.
- There is a need for a workstation to be a low-technology item from the point of view of a first-year college student trying to uncrate it and set it up. The complexities of a distributed system, with interlinked and interdependent services, layered on to the complexity of a 3M workstation, do not contribute to this goal.
- Despite all the discussion about naming in the operating systems literature over the last several years, it isn't obvious just how much (or how little) function is really needed to glue a particular community of users and services together.