GRAPEVINE: AN EXERCISE IN DISTRIBUTED COMPUTING (SUMMARY)

Andrew D. Birrell, Roy Levin, Roger M. Needham*
and
Michael D. Schroeder

Xerox Palo Alto Research Center,
Palo Alto, CA 94304, USA

1. Background

Grapevine is a distributed, replicated system running on a large internet within the Xerox research and development community. The internet extends from coast to coast in the USA, to Canada and to Europe, and contains more than 50 Ethernet local networks linked by leased telephone lines. Over 1500 computers are attached to the internet. Most computers are used as personal workstations, but some are used as *servers* providing access to shared facilities such as printers, large-scale secondary storage, or data bases. Computers on the internet are uniformly addressable using the *PUP* family of protocols.

2. Goals

The primary aim of the Grapevine project was to design and implement a substantial distributed system in order to improve our understanding of practical structures and techniques for distributed computing. The functional area we chose for this exercise was delivering computer mail. We set four major design requirements for the system. First, the system should not be limited in size or extent by the size or power of the constituent computers. It should expand gracefully to handle more load by adding more computers. Second, the system should have high availability in the presence of computer or internet failures. No failure of a single Grapevine computer should make any service unavailable to any client. Third, the system should protect itself against incorrect or malicious client programs. If a user runs incorrect software in his personal workstation, it should affect only that user and not cause incorrect or substantially degraded service for other users. Finally, the system should allow decentralized administration. There are many organizations participating in the Xerox internet, and each of them handles such events as employees arriving or leaving and internal reorganizations independently. It should be practical for these independent administrations to give effect to their decisions without consulting any central authority, except in rare cases.

---

* Roger M. Needham's regular address is: University of Cambridge Computer Laboratory, Corn Exchange Street, Cambridge CB2 3QG, United Kingdom.

3. Services

We chose to provide the following services with Grapevine. The selection was based on utility for the computer mail application, although all have utility for other applications. Grapevine will transport a *message* from one client to a given list of recipients; Grapevine does not interpret the contents of the message in any way, so it need not be a textual message. Grapevine provides a registration data base; message recipients are names in this data base. The names may identify individual recipients or distribution lists; the content of a distribution list is a set of names. If a distribution list is a recipient of a message, the message is sent to all the names in that list; these names may themselves be other distribution lists. This registration data base also offers an *authentication* service (by having a password associated with individual names), and may be used to provide uniform *access control* and *resource location* services on the internet. Grapevine also provides interactive facilities for changing the registration data base.

4. Distribution of Data and Function

There are three logical components of the Grapevine system: the *registration servers*, which provide the facilities of the registration data base; the *message servers*, which provide the message transport and buffering service; and the *GrapevineUser* package, which provides the interface between the software clients of Grapevine and these servers. Grapevine uses dedicated computers each running one registration server and one message server. The GrapevineUser package runs in other computers along with client software. A typical client is a computer mail system interface program that runs on a personal workstation. Grapevine trusts that the registration and message servers are correct. Each client of Grapevine (but not Grapevine itself) trusts that the GrapevineUser package running in the same computer is correct. No client need trust any other client's version of the GrapevineUser package.

Every message server is equally able to accept a message from any client for delivery to any set of recipients. The message servers will then cooperate to forward that message to one of a small set of message servers for each recipient, where the message is buffered until the recipient cares to retrieve it. Thus, provided *any* message server is available, a client may submit a message for delivery; provided any of a particular small set of message servers is available, a client may still receive messages for a particular individual. Since messages are not replicated, the temporary unavailability of a single message server may make certain messages for particular individuals temporarily unretrievable, but no single message server failure will prevent any individual receiving new messages. The message servers use the registration data base for finding the contents of distribution lists, for deciding where to forward a message for a particular individual, for locating registration servers and other message servers, and for authenticating individuals.

The registration servers support the registration data base. The name space is structured as a two-level hierarchy. This hierarchy is the basis for independent administration of subsets of the name

space and for distribution of the data base among the servers. The upper level of the hierarchy is called a *registry*. There is one registry for each administrative authority. Each registration server contains complete data for some subset of the registries; no registration server contains all registries. Each registry is stored in more than one registration server. Any suitably authorized user may alter the contents of the data base by contacting any registration server that contains the appropriate registry; the registration server will then ensure that the update is propagated to each other registration server that knows about the affected data base entry. Thus the registration servers provide a distributed, replicated data base. The registration servers use the message delivery service to propagate updates.

The GrapevineUser package that executes with client software implements the communication protocols for using the services provided by the message and registration servers, and uses the resource location facilities of the registration data base to locate an appropriate message or registration server for each operation. GrapevineUser makes the multiple Grapevine servers look like a single service to a client.

In addition to its use for supporting the message servers, the registration data base describes the configuration of Grapevine itself. The data base is also used to enforce access controls on internet administrative functions, and as the administrative data base for a file server, and as the binding data base for a remote procedure call facility.

5. The Registration Data Base

A *name* in Grapevine is of the form *firstPart.registry*. There are two *types* of names: *individuals* and *groups*. Individuals represent human users of Grapevine and the various Grapevine servers; any individual may receive messages. Groups represent distribution lists, access control lists, services, and the configuration of Grapevine; a single group may be used for all these purposes. The values maintained for an individual are: a password, a list of message server names indicating where the individual prefers his messages to be buffered, and an internet address used to establish an internet connection when the individual is a server.

The values maintained for a group are: a textual *remark* used to give an indication to a human of what the group is for, the set of names that are the *members* of the group, and two access control lists used by the registration server for controlling updates to the group. If the group is being used as a distribution list for message delivery, then the members indicate the appropriate set of recipients; if as an access control list, then the members are those names that have some particular access rights; if for system configuration or resource location, then the members are the names of particular servers.

The registration data base is used to describe its own replication and distribution. Each registry is represented as a group. The members of the group are the registration servers that contain data for that registry. All registry names are in the special *gv* (for Grape vine) registry. This group is used by clients to locate an appropriate registration server for a data base operation, and by the

registration servers to propagate updates to the data base and to determine what subset of the data base should be maintained in each server.

A client program, for example one that implements an interactive administrative interface to Grapevine, may initiate an update to the data base by contacting any registration server that knows about the registry to be changed. Any of the values associated with a name may be altered, subject to the access controls; if the value is a set of names (such as the members of a group), then the update is of the form "add *x.y* to the set" or "remove *x.y* from the set". Updates are independent and atomic on each server. The registration server propagates the update to the other replicates of this registry by sending Grapevine messages to every other server in the group describing the registry. For example, an update to the name *p.q* causes a message to be sent to the servers in the group *q.gv*. Propagation of an update to all registry replicas can take several minutes.

Every registration server knows about names in the *gv* registry. Thus, to locate a registration server that knows about names in some other registry, say *q*, a client contacts *any* registration server, finds the server names which are the members of the group describing the registry (for this example *q.gv*), then contacts any of those servers. This is a special case of Grapevine's resource location facility: to describe a resource which is supported by multiple servers, have a group in the data base whose members are the names of those servers. As another example, *Maildrop.ms* is a group which contains the names of all the Grapevine message servers. To locate a message server for message submission, a client finds the members of *Maildrop.ms*, then retrieves the internet address of each from the data base, then contacts one of them.

6. Message Delivery

To have a message delivered, a client program submits the message and the recipient names to any message server. The message server finds the complete set of individuals who should receive the message by (recursively) finding the members of each group in the recipient list. For each individual recipient, the message server obtains from the data base the names of the message servers where mail for that individual should be buffered. These server names are in a preference order. The message server chooses a destination message server for each individual on the basis of this preference order, modified by this server's opinion of the present accessibility of the various other message servers. The server partitions the list of individual recipients into a set of *steering lists*, one for each message server to which the message should be forwarded, and one for recipients whose mail should be buffered on *this* server. The message server then forwards the message and appropriate steering list to each of the other servers, and buffers the message in *in-boxes* for local recipients. Upon receiving a message with steering list from another another message server, the same algorithm is performed, using the steering list as the recipient list. When processing a forwarded message most recipients' inboxes will

be local, unless the registration data base is changing.

A cache of *hints* about the location of individuals' in-boxes is maintained by each message server to ensure that this delivery algorithm is acceptably efficient, and suitable measures are taken to invalidate parts of this cache so as to ensure convergence of the delivery process. With high probability, no recipient receives multiple copies of a message.

For a client, such as a computer mail interface program, to retrieve an individual's new messages, the GrapevineUser package periodically polls each message server in that individual's inbox list, and notifies the client when new messages have arrived. The client may then contact each of these inbox servers and retrieve the messages. The client may provide message reading, filing, answering, and forwarding facilities for its human user.

7. Discussion

Grapevine has been in full service for almost a year. Seven Grapevine computers presently provide mail and data base facilities to a community of about 1500 users. The busiest message server has handled 31000 messages in the 1000 hours since it was last restarted. Users find the system highly efficient, reliable and available. Distributed administration of Grapevine has proved possible and convenient. For example, most mail system users now add and delete themselves from various distribution lists, removing this tiresome job from administrative staff. The distributed, replicated registration data base is starting to be used by other applications for access control, authentication and resource location. Expansion of Grapevine to reflect increased or redistributed use of its services has been quite easy. We believe that the system easily can meet an order of magnitude greater load simply by employing more servers.