Inside Now Up-to-Date™:
A System Administrator's Primer

David Riggle
Now Software, Inc.
January 1995

Introduction

This primer is intended for System Administrators at large sites who need to support thousands of Now Up-to-Date™ users with the greatest efficiency and ease. It describes the inner workings of Now Up-to-Date™'s networking and meeting arbitration features as well as giving several examples of network topology and server configuration.

Now Up-to-Date™ Overview

In addition to a wide range of personal calendar features, Now Up-To-Date™ allows users on a network to share information via *Public Categories* and to coordinate their schedules via *Meetings*.

Public Categories are like bulletin boards where notices and events of general interest are posted. For example, "Cafeteria Menu" and "Cafeteria Staff" could be examples of Public Categories. The first would contain events of interest to the eating public, while the second would contain events of interest to the cafeteria personnel. An event on the "Cafeteria Menu" category might be "Friday Afternoon Beer Bust." An event on the "Cafeteria Staff" category might be "Thursday Bread Delivery."

A Meeting is a wish that a group of people will be together at a certain place and time. A Meeting consists of an invitation list of users, optional resources, and an optional conference room, and a Public Category. Meetings exist on Public Categories just as ordinary events do. What makes them different is the list of users, resources, etc. invited to attend. An example of a meeting on the "Cafeteria Staff" category might be "Chef Salad Seminar, Thursday at 8:00 PM in the Kitchen with John, Bob, Cindy, and the Big Mixing Bowl invited to attend."

Users invited to Meetings get notification of the Meeting, and it shows up in their calendar. The Meeting also blocks out time on the invited users' schedules (more on that later). Users may accept, decline, or ignore meeting invitations. A meeting, whether accepted or not, will block out time on a user's schedule unless it is declined.

Sometimes a meeting may be “open.” In other words, anyone can attend if they want to. For example, another event on the “Cafeteria Staff” category might be “Bread Making Class, Tuesday night; space is limited, only the first ten people to sign up may attend.” This event is posted to the “Cafeteria Staff” category with no invited users, only a conference room and some resources are invited. As staff members see this event appear in their calendars, they can sign up for the class by “attending” the event, in effect adding themselves to the invitation list.

Network Architecture Overview

Now Up-to-Date™ uses a client - server network architecture. Now Up-to-Date™ is the client application. Public Event Server is the server application. The Now Up-to-Date™ calendar document stores all the server connection information and virtually all preference information as well. This self-contained storage makes it possible for a system administrator to distribute a single pre-configured calendar document that will get users up and running in an instant.

A Now Up-to-Date™ document can be connected to anywhere from 0 to 15 Public Event Servers. Each server offers up to 1100 Public Categories + Users + Resources + Conferences Rooms available for subscription. (As far as the server is concerned, Users, Resources, and Conference Rooms are all just categories.)

When Now Up-to-Date™ is running, each open document queries the servers it is connected to at regular intervals, usually every 5 minutes. The query contains 76 + 4 * X bytes, where X = the number of categories the document subscribes to from that server. If no data has changed since the last query, the server returns 26 bytes. In an idle steady state where each user is connected to 10 servers, subscribed to 100 categories per server, and updating every five minutes, the average network load is 16 bytes per second per active user. Since most people don’t keep their calendars open all the time, the number of active users at any given time is usually a small fraction of the number of installed users.

Polling, the model of clients initiating communication with servers instead of the other way around, has many benefits in a client - server architecture. First, polling is more efficient for a busy server on a busy network. Polling makes updating and network activity a “batch process.” Instead of the server having to notify every connected client every time something of interest changes, the server only sends the information when the client is ready for it. Second, third, fourth, and fifth, the passive servers in a polling model are smaller, simpler, faster, and much more robust and reliable.

In grand terminology, Now Up-to-Date™ uses a distributed, non-replicated network database. That means, if you want a particular piece of data, you have to connect to

the server it is on, rather than to some local copy of the real data. In replicated databases, the servers talk to each other, pass updates around, and argue amongst themselves. Sometimes the local copies are read-only, so you have to talk to the real server to change the data anyway. The theory and hope is that having a local copy of the data gives a large enough performance improvement to make up for the tremendously greater server complexity, server-to-server network overhead, and system administration headaches. At some point this is true, but probably not until you have a global network with tens of thousands of users.

In traditional network databases, when a server goes down users are inconvenienced to a greater or lesser degree. Greater in non-replicated databases, lesser in replicated databases. In Now Up-to-Date™, users scarcely even notice when a server goes down. They can continue working with and modifying public data. When the errant server comes back online, their changes are automatically sent to the server. This feature is called Offline Editing, and has set a new standard in distributed database design.

Network Load Calculations

In the example above, with each user connected to 10 servers and subscribing to 100 categories from each server, the idle steady state network load was 16 bytes per second per active user. The following example describes a busy steady state load.

When a public event is modified, Now Up-to-Date™ sends 162 + A + B + C + 6 * D + 70 * E + 16 * F bytes to the server, where A = the length of the event's title, B = the length of the event's description, C = the size of any attached PICT, D = the number of times the event repeats, E = the number of contacts attached to the event, F = the number of attendees if the event is a meeting. The server replies with 26 bytes. The client then begins an update, even if its next scheduled polling time has not yet arrived. The update request will be as described earlier. The update response will contain 26 + W * (128 + 20 * Z + A + B + C + 6 * D + 70 * E + 16 * F) bytes, where Z = the number of A-F values that are non-zero, and W = the number of events queried (W = 1 in this example). The client's polling timer is then reset to its full value.

Some reasonable values for an event give a total of 275 bytes sent to the server, 26 sent back, then an update query of 476 bytes (76 + 4 * 100 from the previous example), followed by an update response of 327 bytes. Let's assume a busy steady state where each active user is modifying one public event from the same server every minute. Let's further assume that on average 20% of the active users will be subscribed to an event that gets modified. Therefore in a five minute period each user has sent five edits and initiated five updates from one server, and one update each from the remaining nine servers. Each update from the last nine servers returns 5 *

N/10 * 20% = N/10 events, where N is the number of active users (and N is large). Each update from the first server returns N/10 * 20% = N/50 events.

The total bytes sent to and received from the first server is 4015 + 30 * N bytes. The total for each of the remaining nine servers is 502 + 30 * N bytes. The five minute total network load for each active user is 8533 + 300 * N bytes, or about 28 + N bytes per second per active user. If N is large, the network load in a busy steady state is about N bytes per second per active user. In a multiple network environment the loading described in this example will be divided fairly evenly among the participating networks.

Meeting Architecture Overview

The fundamental property of a meeting is that it "blocks out time" on someone's schedule. In Now Up-to-Date™ these schedules are special types of categories called Users, Resources, and Conference Rooms. When a meeting is proposed on a Public Category, a copy of the meeting (an *invitation*) is stored on each invitee's schedule category. In this way, users can be notified of a meeting even if they don't subscribe to the Public Category that it came from.

Inside a User, Resource, or Conference Rooms category the meeting scheduling preferences are stored. If a user works M-Th 8-2:30, that information is stored in the user's category, along with his "login" password, office phone extension, etc.

When a meeting is created, the first thing that happens is the event gets stored to its Public Category. If that succeeds, then invitations are sent out to each invitee. Resource and Conference Room schedules are arbitrated by the servers they are stored on. An invitation to a Resource or Conference Room can be rejected by the server if a conflict occurs between the proposed meeting and existing meetings. If no conflict occurs, the server automatically accepts the meeting on behalf of the Resource or Conference Room. If all invitations have been sent successfully, the user who created the meeting sits back and waits for the invitations to be accepted or declined.

Except for the server arbitrating Resource and Conference Room schedules, all Meeting Arbitration features are implemented by the client. This division of labor keeps the server workload light and makes time consuming operations like auto-pick a local operation for the client.

Server and network workload are significantly greater when meeting arbitration is in use. In addition to the meeting stored on a Public Category, each invitee has a copy of the meeting stored into his schedule. What's more, most likely he will accept or decline the invitation, thereby modifying it. Since most users subscribe to a good many other user's schedules, a good deal of schedule data is retrieved during

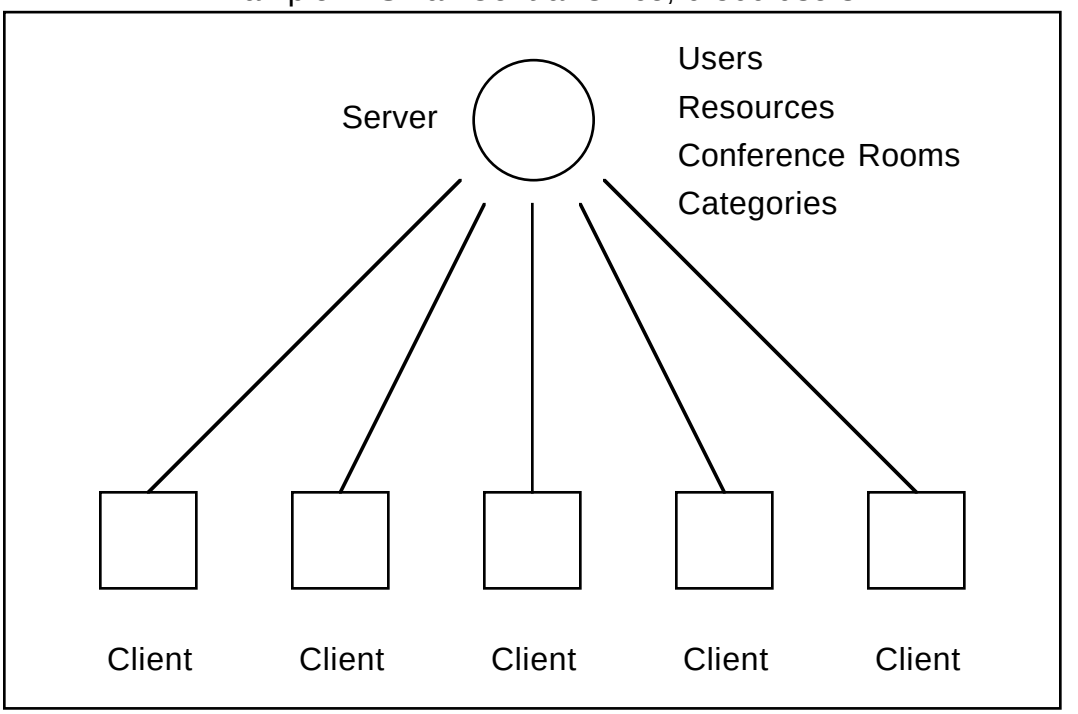
updates. The best way to improve meeting arbitration performance is to limit the number of Users, Resources, and Conference Rooms that users subscribe to.

For example, suppose each user from the previous example was scheduling one meeting every minute and inviting R users. Let's assume that all R users are on the same server. The additional workload would be R extra stores and one extra update per minute. The amount of data retrieved from updates would increase by R * V, where V = the percent of Users that everyone subscribes to. The total amount of additional data is 3786 + R * V * 60 * N bytes per minute, or about R * V * N bytes per second per active user for large N. As you can see, the server and network load increase directly with the meeting size and the percent of Users, Resources, and Conference Rooms that active users are subscribed to.

Sample Network Topologies

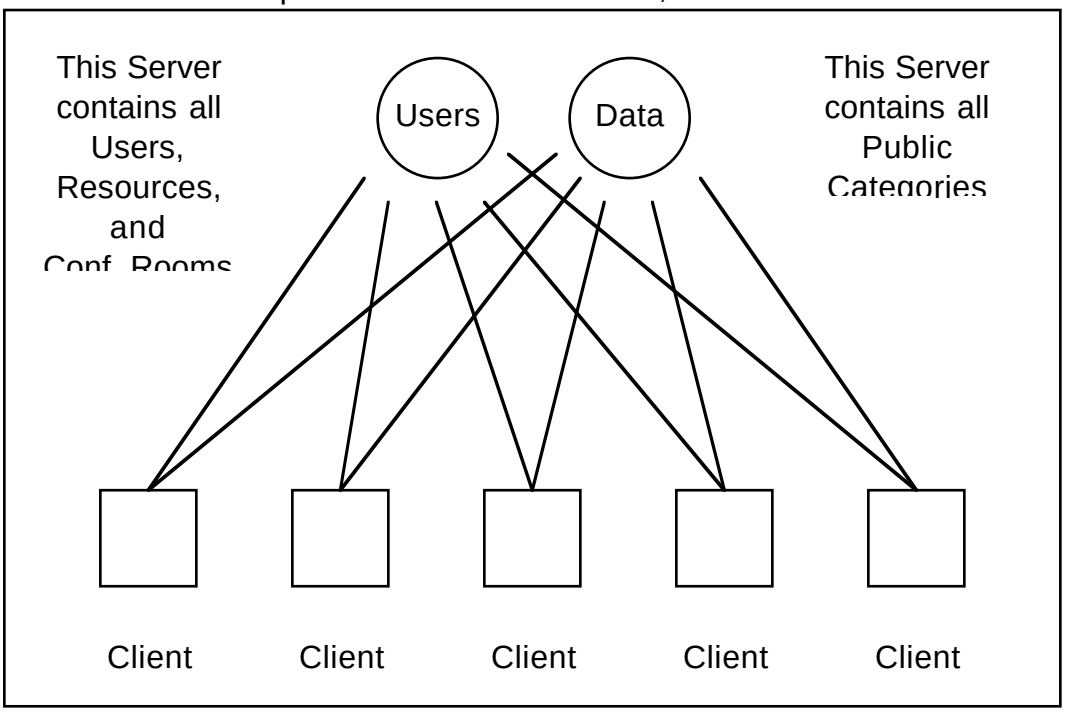
There are many different ways to set up a Now Up-to-Date™ network. Depending upon the number of users you intend to support, your geographical distribution, and your meeting arbitration goals, you may choose one arrangement over another. Here are some examples from one end of the spectrum to the other.

Example 1: Small Central Office, 0-500 users



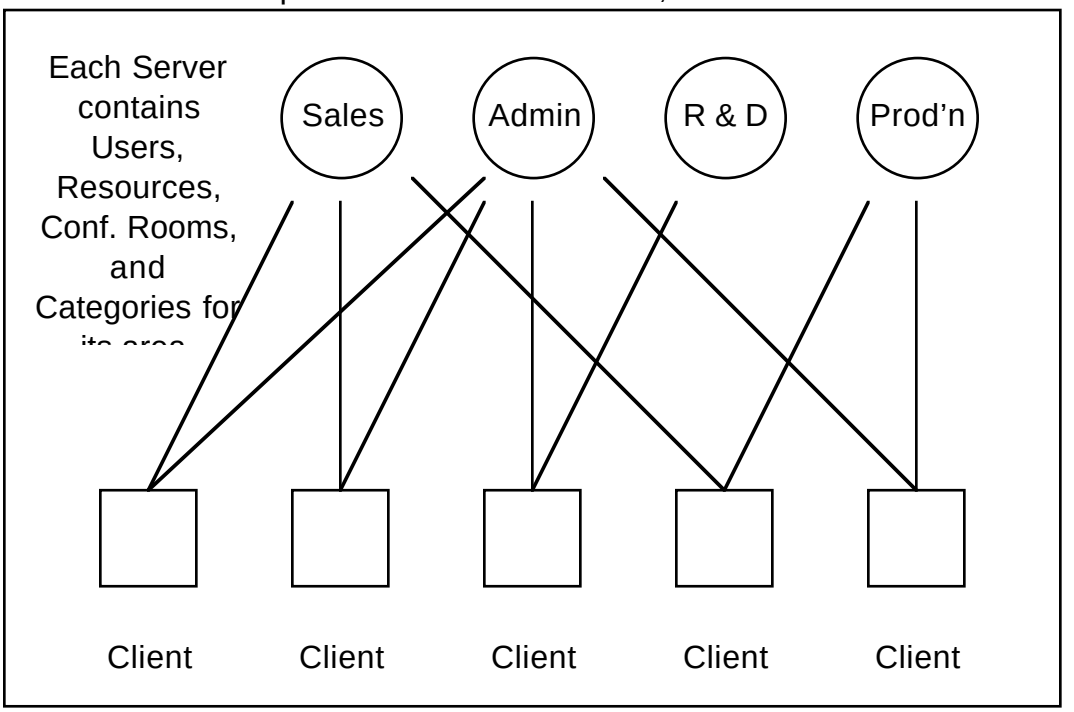
This is the simplest possible set up. Obvious disadvantages are limited server performance and limited number of users.

Example 2: Small Central Office, 0-1000 users



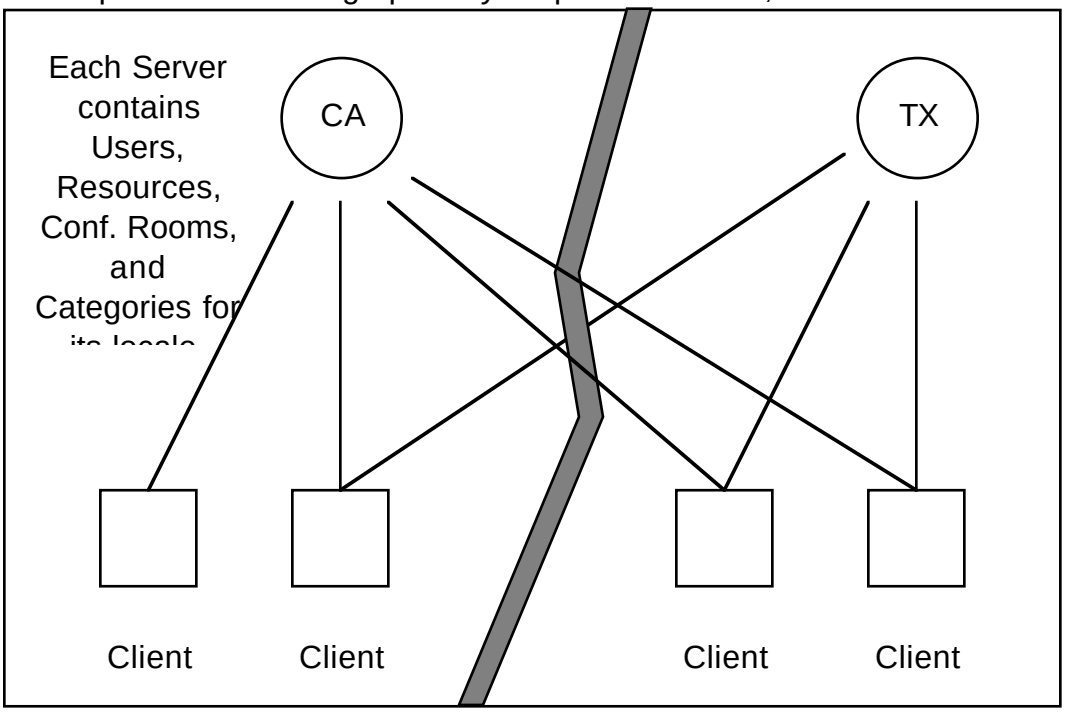
This configuration has the advantage of dividing the workload nicely between two servers. A second advantage is that clients who don't participate in meeting arbitration only need connect to the Data Server. Disadvantage: still a limited number of users.

Example 3: Small Central Office, 0-2000 users



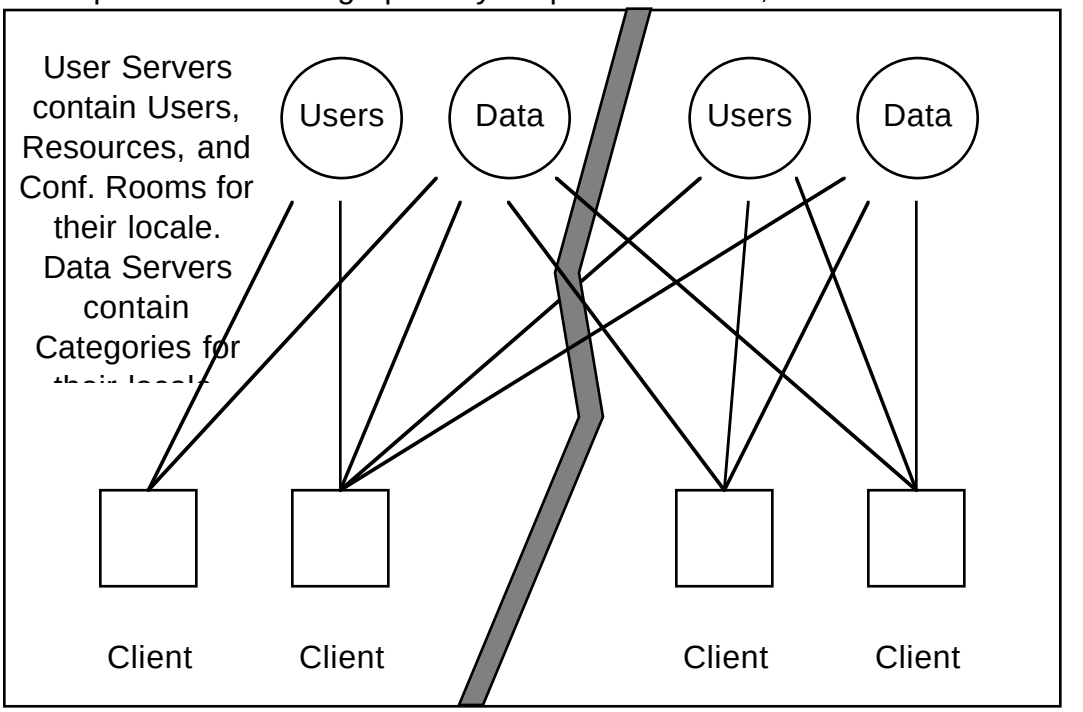
This example is different from all the others in that the servers are not specialized in a particular type of data. Each server holds all four types of data. The servers are split amongst organization lines. Advantages: it supports more users, and it divides the workload somewhat. Disadvantage: Users may occasionally need to be moved from one organizational server to another -- an administrative headache you don't need.

Example 4: Small Geographically Dispersed Offices, 0 -500 users each



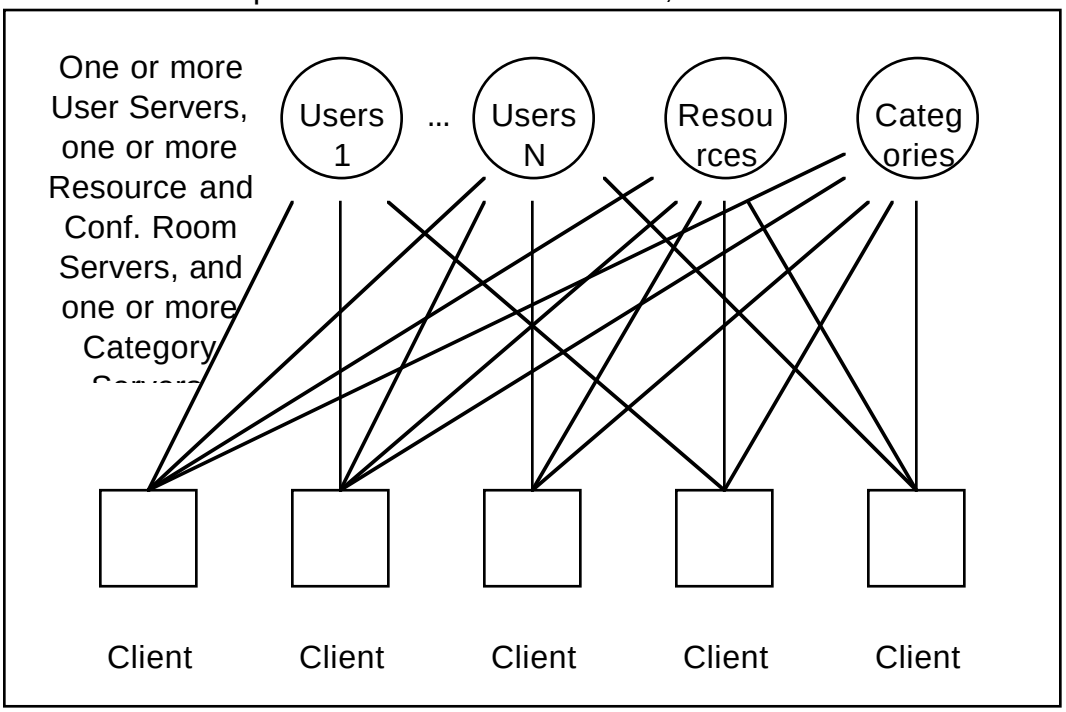
Two Example 1's glued together. Simple to maintain, but it supports a limited number of users.

Example 5: Small Geographically Dispersed Offices, 0-1000 users each



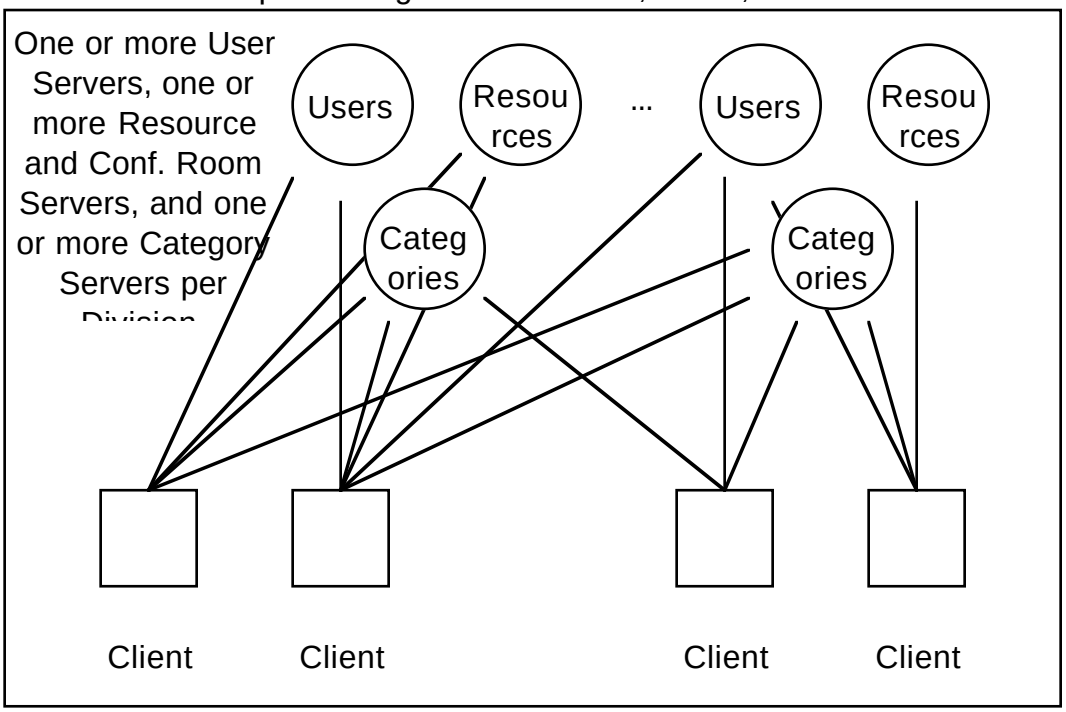
Two Example 2's glued together. Nice division of labor between the server pairs. Allows users in other geographic locales to connect to more interesting Category servers separately from less interesting User, Resource, and Conference Room servers.

Example 6: Medium Central Office, 0 - 5000 users



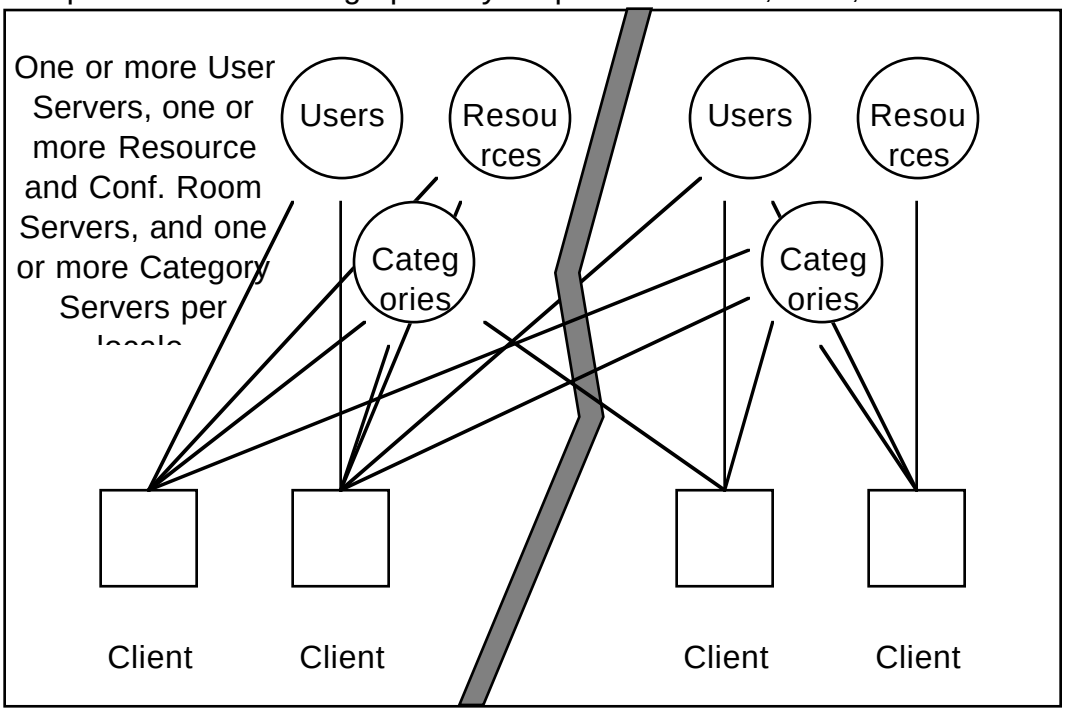
This design is for the Medium-sized company that is either too chaotic or too cohesive to have separate, mostly autonomous divisions. Advantages: good division of labor among the multiple User servers. Probably only one Resource & Conference Room server will be needed. Multiple Category servers can help to subdivide the public data space into more logical and manageable components.

Example 7: Large Central Office, 0 - 25,000 users



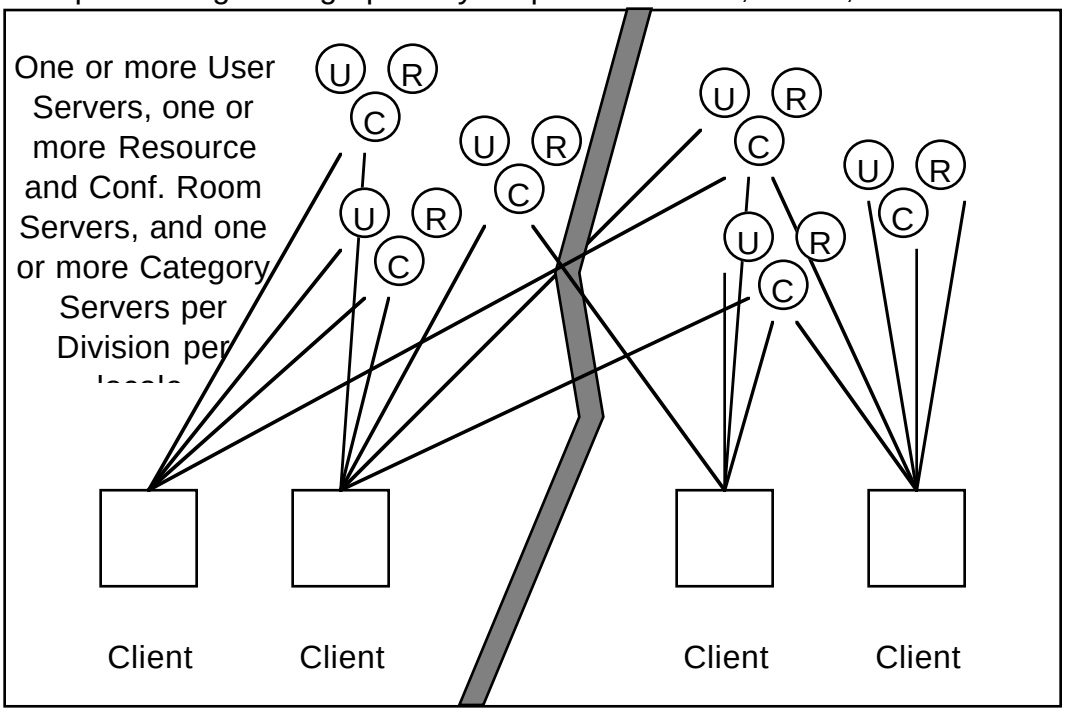
This is a good choice for a central campus of several thousand users. Each Division acts like its own geographical office, similar to Example 5. Further division of labor is achieved by separating the Users from the Resources and Conference Rooms. In such a set up, some Category servers will probably serve the entire company, or good portions of it, while others will be Division or group specific. Note that not all servers can be used by all clients. There are too many of them. Each client can connect to at most 15 servers, giving him access to at most 15,000 users for scheduling purposes.

Example 8: Medium Geographically Dispersed Offices, 0 - 5,000 users each



This is two Example 6's glued together. Advantages: good division of labor, long distance network traffic reduced, and foreign clients can connect to local Categories without being bothered with local Users, Resources, and Conference Rooms.

Example 9: Large Geographically Dispersed Offices, 0 - 25,000 users each



This is two Example 7's glued together. It is the best choice for large, world-wide corporations.

Server Configuration Guidelines

Once you decide how many servers you will need and what types of categories to put on each server, the following guidelines will be helpful in setting up your servers properly.

Keep several megabytes of memory free on the server machine. The servers use temporary memory for their normal operations. Allocating more memory to the server application actually reduces the amount of temporary memory on the machine and causes the server more problems. The only time you will need to increase the server memory allocation is if you are running out of memory trying to convert an old server database. In that case, increasing the server memory allocation can help. Be sure to lower it again after the conversion is successful.

Use a consistent naming convention for your server names. For example, if different departments have their own Category servers, name the servers after the departments they serve. For example, "Company Wide," "Sales & Marketing," "Engineering," and "Admin." If you have a small geographically dispersed company, name the servers after the location and type of data they store. For example, "Cupertino Personnel,"

“Cupertino Resources,” “Cupertino Categories.” If you have a large geographically dispersed company, use a combination of the above naming conventions. For example, “Cupertino Personnel 1,” “Cupertino Personnel 2,” “Cupertino Personnel 3,” “Cupertino Resources,” “Cupertino Company Wide,” “Cupertino Sales,” “Cupertino Marketing,” “Cupertino R & D”, “Cupertino Support”, “Cupertino Exec. Staff,” etc.

When scheduling meetings, users should be encouraged to place the meeting on the Public Category that makes the most sense for the meeting, even if the invitees are drawn from a cross section of the company. Use of a “catch-all” or “cross-departmental meetings” Public Category as a resting place for meetings is a bad idea. It will fill up with junk and nobody will want to subscribe to it. It is best to keep meetings under the categories that they most logically belong to. Invitees don’t need to subscribe to a meeting’s category if they don’t want to. They can accept or decline a meeting without being subscribed to its public category.

Supercharging Server Performance

Here are some tricks you can use to squeeze the maximum performance out of your server machines. In order of performance boost, they are as follows:

Add more servers. Generally the best solution to sluggish performance is to divide the workload among more server machines.

Reduce extraneous server machine workload. Dedicate the machine to one Now Up-to-Date server if possible.

Use faster machines. The faster the CPU and hard disk, the better performance will be. Now Up-to-Date™ servers are disk intensive.

Use a RAM Disk. Put the server and database file on an automatically backed-up RAM Disk. Use “CopyDoubler™” or the freeware “RAM Disk Helper” to automatically load and back-up the RAM Disk. Set the back-up frequency to be somewhere between five minutes and an hour, depending upon how much data you can afford to lose in a power failure.

Archive server data. A smaller database is a faster database. If the server database file is a few megabytes or more, it is time to archive.

If you don’t use a RAM Disk, set the Disk Cache to 256K in the Memory Control Panel. This cache will improve disk I/O performance somewhat.

Install a faster network. This may deserve a place higher in the list, but it is not really a

server tweak. Ethernet should be sufficient local area networks. For long distance links, do whatever you can. You don’t need to go overboard, however. For really remote users, Now Up-to-Date™ works great over AppleTalk Remote Access.