Why Centralizing Microsoft Servers Hurts Performance

By Peter Sevcik and Rebecca Wetzel
March 2006

Unstoppable economic and regulatory tides are evicting servers from distributed locations and resettling them into central data centers to be shared by many far flung users. Unfortunately, the impeccable business logic behind server centralization has an unintended consequence—it can debilitate the performance of vital applications.

Among the applications most likely to be hurt by server centralization are those like Word, PowerPoint, and Excel, which use Microsoft's Common Internet File System (CIFS) file sharing protocol, as well as Outlook which uses Microsoft's Messaging Applications Programming Interface (MAPI) email protocol for Exchange. We will explain why performance of CIFS and MAPI-based applications degrades as distance to users increases, and how you can alleviate the problem without fixing—or even touching—the affected applications.

What Is Driving Server Centralization

You no doubt know that cost savings and conformance with regulations like the US Sarbanes-Oxley Act are the business currents propelling server centralization. Centralizing servers cuts server-related hardware, software, management, and equipment space costs—while bolstering security, simplifying data replication, improving business integration through information sharing, and enabling tighter control and more efficient use of corporate assets. Server centralization also helps businesses secure and manage information to meet increasingly stringent regulatory requirements.

But what you may not know is that if you are unprepared to navigate the performance implications of these business currents, you may find yourself bailing hard to keep your networked Microsoft applications afloat.

Server Centralization and Application Performance

Despite its many business benefits, server centralization inevitably stretches the distance between users and servers, which can degrade performance to the point that users become disgruntled and unproductive. Such performance woes can give IT a black eye, and can even cause a server centralization initiative to fail.

To make matters worse, another business initiative—workforce globalization—further distances users and servers, as distributed offices, mobile workers, partners, and outsourcing contractors access centralized servers over ever longer network spans.

Distance hurts performance because it lengthens the round-trip time between the user and the application server. Although time added due to distance (a.k.a. latency) may go unnoticed for applications with few client-server software interactions (or turns), it poses a problem for applications requiring many interactions, like those based on the "chatty" application protocols of CIFS and MAPI—and the problem cannot be solved by adding bandwidth.

Why CIFS and MAPI Pose Problems

CIFS and MAPI were designed to run over local area networks, where the performance price for application protocol "chattiness" is negligible. Unfortunately, when run over a WAN, such chattiness exacts a heavy toll.

NetForecast Report 5081

CIFS runs slowly over a WAN because when a client requests a server to open, close, or read a file, rather than send the whole file, CIFS breaks it into blocks and transmits the blocks sequentially. For example, if a client requests a 1Mb file from a remote server, CIFS breaks the file into hundreds of data blocks and sends the blocks one at a time. The client acknowledges receipt of each block, and only after receiving the acknowledgement, does the server release the next block. This back and forth chattiness adds unwanted seconds, or even minutes to the simple task of opening the file. Microsoft's MAPI protocol exhibits similar behavior (or misbehavior) over a WAN.

To place CIFS and MAPI-based application performance into perspective, it is useful to compare performance profiles for CIFS and MAPI tasks with other common application tasks. Figure 1 shows application turns and payload per task on logarithmic scales. As you can see, the number of turns required for CIFS and MAPI-based applications are exponentially larger than for client server, SNA, and even most Web-based applications.

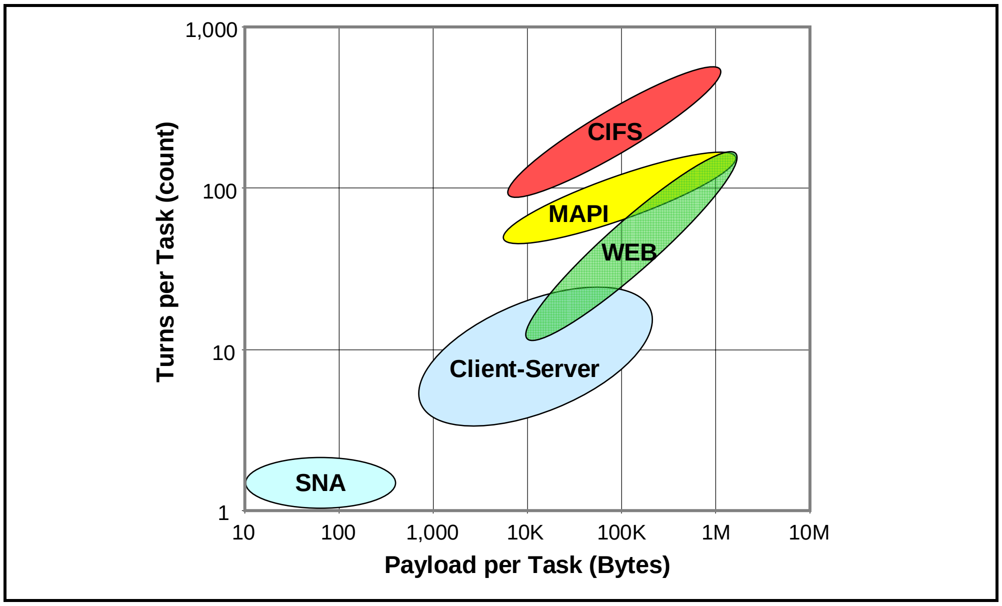

Figure 1 – Application Profiles

NetForecast created the Figure 1 profiles using actual, not theoretical, profile data for typical user tasks. The CIFS profile summarizes turn count and payload results from accessing PowerPoint files ranging from 10KB to 1MB on a Microsoft network file server. The MAPI profile shows results from loading email plus attachments (in the 10KB to 1MB range) from a Microsoft Exchange server.

Because Figure 1 scales are logarithmic, small distances on the chart reflect huge differences in the user's experience. With this in mind, you can see that the turns and payload burden for MAPI and CIFS-based applications is exponentially greater than for SNA and client-server applications.

It is worth noting how the payload and turns required to refresh a user's screen jumped dramatically as enterprises migrated from SNA to client-server, and then to Web and Microsoft-based network applications.

Understanding Application Response Time
To grasp the relationship of distance and chattiness to performance, it is important to understand the factors affecting application response time. These factors include: application turns (application chattiness), application payload, network bandwidth, network round-trip time (RTT), server compute time (Cs), and client compute time (Cc). The following formula shows how these factors combine to determine a user's response time experience for a given task (see the Glossary of Terms at the end of this report for detailed formula definitions).

$$ResponseTime \approx Turns(RTT) + \frac{Payload}{Bandwidth} + Cs + Cc$$

The equation makes it clear that because data center centralization increases round-trip time, and because the longer round-trip time must be multiplied by the high turn counts characteristic of CIFS and MAPI-based applications, task response times experienced by WAN-based users become untenably long.

Long Response Times Are Bad for Business
Long task response times are bad news for business. When response time hobbles the user experience, it can hurt productivity, annoy customers (even drive them away), overwhelm help desks with user complaints, demotivate partners—as well as erode the business benefits of server centralization and workforce globalization.

The ill effects that long task response times exert on business can be subtle. For example, employees may cope with poor performance by hoping that performance will improve later, so they defer work until a hoped for improvement occurs. Such a delay can slow or even paralyze a project, and/or hurt competitiveness, cost, and efficiency. If performance is bad during normal business hours and better after hours, an enterprise may be forced to take steps that can harm the business like add employees to work additional shifts, thus raising costs.

NetForecast Performance Model
NetForecast has studied application performance for over a decade and has embodied that knowledge in a detailed model that accounts for the performance factors that vary with network distance.

Among the factors accounted for in the model is the inescapable fact that round trip time is directly proportional to distance divided by the speed of light. Since no one yet has managed to accelerate the speed of light, it is safe to infer that the longer the distance, the longer the round trip time. Also, the model takes into account the fact that packet loss escalates with distance because the risk of congestion increases with higher network hop counts.

NetForecast applied its model to this analysis to show the effects of user distance on performance for representative tasks. Figure 2 shows various scenarios of users accessing a server centralized in a New York City data center. Because they are near the server, New York-based users experience excellent performance for all application types modeled. Users at different distances from the New York sever, however, experience remarkably different results depending on the application, with CIFS and MAPI-based applications showing the most dramatic performance gap.

NetForecast has found that it is not useful to compare raw response time values for each application because task definitions and judgments about acceptable task completion times vary widely for each application type. For example, what might be considered a painfully slow task time for one application task type might be entirely acceptable or even excellent for another.

Therefore, to equalize the view of performance across applications and network scenarios, we apply an application performance index methodology called Apdex, developed by the Apdex Alliance as an industry standard for measuring and assessing user satisfaction with application performance (see sidebar *How Apdex Works*).

How Apdex Works

Apdex provides a numerical measure of user satisfaction with the performance of enterprise applications. Apdex reports are arrived at using a four-step method.

Step 1 – Select Target Time
Define a target response time of T seconds. Response times between 0 and T are considered satisfactory by the users within a business scenario.

Step 2 – Measure Performance
Measure performance and place each measurement into one of three user-perceived performance zones: satisfied (Sat), tolerating (Tol), or frustrated. The satisfied zone is defined in Step 1. Frustrated times are above F seconds, where F=4T. Tolerating times are between T and F seconds.

Step 3 – Apply the Apdex Formula
Sum the incidents of response times that fall into each zone, and divide by the total number of measurements using the formula (the sum of frustrated incidents is not in the numerator but is included in the denominator):

$$Apdex = \frac{Sat + Tol/2}{Total}$$

Step 4 – Show Results
Show the Apdex Index value (on a scale of 1 to 0) together with the corresponding threshold T. The Apdex Scale is subdivided into performance quality regions ranging from excellent to unacceptable.

In this report, we describe how T is selected for each application (Step 1). For brevity, the details of Steps 2 and 3 are not shown. Results of Step 4 illustrating the Apdex Quality Regions are shown in Figure 2.

Apdex is an open standard developed by the Apdex Alliance. See www.apdex.org to learn more.

Good Performance Depends on Context

Legacy SNA applications are highly repetitive and designed for short query responses. IBM SNA users typically require a target time of 1 second to remain productive, so we assigned SNA traffic an Apdex target of 1 second (Apdex T=1).

In contrast to SNA applications, the Web typically provides rich information that the user digests more slowly. Consequently, business-to-business Web site performance is often satisfactory with a target time of 6 seconds.

Microsoft MAPI is used to access email, and for analysis purposes we assume each email has a modest-sized attachment that the user will view. Our experience shows that this type of interaction provides satisfactory results if an email plus its attachment is retrieved within 9 seconds.

Microsoft CIFS manages the interactions with shared files on a server. Users generally do not need to load a file frequently, and therefore will tolerate a longer task time—so we applied CIFS a target time of 12 seconds.

Figure 2 shows the Apdex performance results for SNA, Web, CIFS and MAPI traffic traveling over increasing distances. Apdex quality regions go from excellent (blue), to good (green), fair (yellow), and poor (red). You can see that the Apdex performance for each application is excellent (in the blue region) within a 100 or so mile radius from the New York data center, and degrades with distance. Although the Apdex scale goes all the way to zero, the values below 0.5 are considered unacceptable by the Apdex Alliance and are not shown in Figure 2. We view the performance of the applications at distances greater than where they fall off the chart as "not working" for this analysis.

The figure shows that SNA, with a target T of 1 second, still provides fair performance as far away from New York as 10,000 miles (a distance equivalent to Melbourne, Australia). No wonder the airline industry still relies on this computing platform to support agents world wide! Remember this chart the next time you are at a crowded ticket counter on the other side of the world asking an agent to change your itinerary, or even assign you another seat. Be thankful that the airline did not switch to an all-Microsoft platform.

Web applications typically fall into the good to fair range within a large region like North America or Europe, however, their performance degrades to poor as traffic traverses oceans to reach the server. Web applications become unacceptable at about 7,000 miles. Bear in mind that this Web model is based on typical business-to-business applications, and media-rich Web sites will not perform as well.

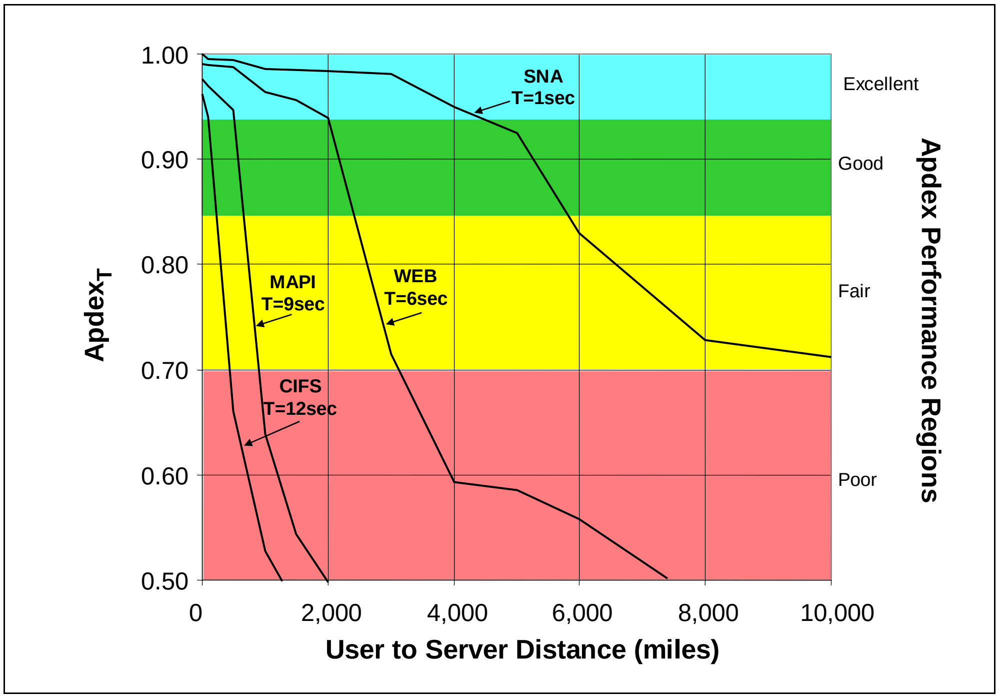

Figure 2 – Performance Curves for Different Application Platforms

Microsoft CIFS and MAPI performance drops precipitously with even small distance increases from the server. The Apdex curves for both CIFS and MAPI (even with generous T values of 12 and 9 seconds respectively) plummet into the poor zone within several hundred miles of the server. With servers centralized in New York, performance for both application types would be poor for users in Chicago and utterly unacceptable for users on the west coast of the United States.

Figure 2 drives home the dramatic and inescapable consequences of CIFS' and MAPI's high turn counts. Without a "fix", users simply cannot be rescued from abysmal performance. To match the performance of the Web Apdex curve (which dives for non-US users), CIFS' target time would need to relax to 30 seconds, which would push the frustrated zone to 2 minutes! Not a promising business move.

What Happens if You Do Nothing

No doubt you have a collection of critical applications that operate on their own performance/distance curves like those shown in Figure 2, and without question centralizing servers for those applications will force performance down the curves. If you are also migrating to Microsoft's file sharing architecture, you are about to subject your users to a "worst case" set of performance curves. Simply put, without ameliorating steps, stretching user-server distance for CIFS and/or MAPI based applications plots a sure course to an application performance shipwreck.

How to Fix the Problem

Contrary to conventional wisdom, the performance shipwreck predicted by the Microsoft server centralization curves in Figure 2 cannot be avoided by adding bandwidth or faster servers, because you cannot escape the physics of latency. Fortunately, however, you can alleviate the problem.

A number of techniques can be brought to bear on CIFS and MAPI traffic to fix poor performance without fixing—or even touching—the adversely affected Microsoft applications themselves. The aim of these techniques is to improve aspects of the applications' behavior over a WAN to lower user task response times. Figure 3 shows "performance levers" that can transform how CIFS and MAPI-based applications operate over the WAN to accelerate them and thus improve performance.

These performance-enhancing techniques include:

- Reducing turn counts by streamlining application protocols
- Reducing application payload through content caching and compression
- Improving network protocol behavior and thus increasing available bandwidth
- Offloading functions to speed server processing

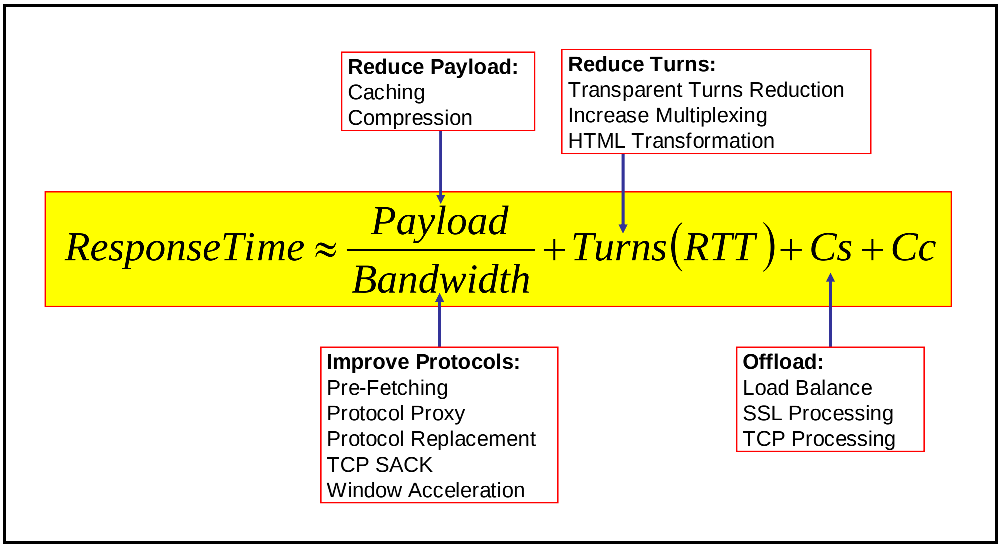

Figure 3 – The Application Performance Levers

You can dramatically accelerate CIFS and MAPI-based application performance by streamlining CIFS and MAPI protocols. By properly combining techniques cited in Figure 3, you can reduce both the payload and turn count by 80 to 90 percent, and this will significantly improve the resulting performance curves.

The Path to Fastest Acceleration

Of the techniques that can help fix CIFS and MAPI performance problems, the most effective are "symmetrical" WAN application acceleration products. They are referred to as symmetrical because they require two devices—one at the data center and the other at the remote site—that work in tandem to transparently improve the behavior of CIFS and MAPI traffic as well as other applications. Figure 4 shows a typical symmetrical acceleration deployment.

Asymmetrical acceleration solutions (using a device in the data center and a browser or plug-in on the user end) exist, but to date they provide modest acceleration only for HTTP traffic. Thus far they cannot accelerate CIFS and MAPI traffic.

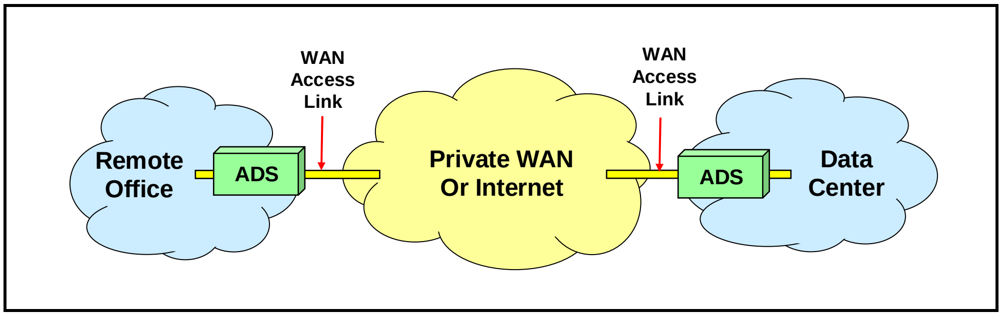

Figure 4 – Application Delivery System (ADS) in the IT Architecture

Acceleration Is Not Enough
Although symmetrical acceleration devices effectively boost CIFS and MAPI performance, they alone do not constitute a complete application performance solution. Complementary control and reporting are essential components to an "application delivery system" (ADS), to assure consistent, long-term CIFS and MAPI performance. Control ensures that CIFS and MAPI performance remains acceptable despite challenging network conditions, and reporting enables you to monitor, troubleshoot, and further improve CIFS and MAPI-related performance.

The control function enables you to identify important traffic, tag it for proper downstream handling, and assign it priority treatment through quality-of-service (QoS). Figure 4 highlights the most performance-constraining factor after latency, access bandwidth. A variety of forces constantly threaten to consume access bandwidth. Some of these forces are malevolent, like worms and denial-of-service attacks, and some are business related, like end-of-day traffic surges. Even every-day activities can hog access bandwidth, leaving users high and dry. It is also important to understand that without control in place, accelerating traffic speeds the propagation of problem traffic such as worms along with legitimate traffic—with potentially dire consequences.

Without the ability to measure task-level CIFS and MAPI application performance and report on it through a useful and easy to understand methodology such as Apdex, you are essentially operating your IT ship without a navigation system. Without effective measurement and reporting, you cannot understand and assess the seriousness of performance problems, nor can you plot a course to a better future.

An application delivery system that includes acceleration, control, and reporting is a must if you plan to centralize your Microsoft servers. Such a system provides a layer of IT infrastructure that links users and the applications they need to do their work. The form your ADS system should take depends on the specifics of your applications, deployment choices and business scenarios. To arm yourself with information for deploying the right application delivery system components, NetForecast recommends that you profile your applications, analyze the effects of potential solutions, and then conduct controlled tests during a pilot deployment. There is no substitute for real measurements!

Conclusion
The economic and regulatory tides driving businesses to centralize servers are unlikely to ebb any time soon. It is, therefore, vital for IT departments to understand that without pre-emptive measures, moving servers away from users can beach Microsoft's CIFS and MAPI-based application users, and subject unwary IT executives to the mutinous response of users.

Apdex is an open performance reporting standard defined by the Apdex Alliance. See: www.apdex.org

A proper ADS architecture should be an integral part of the server centralization plan. Management will not look kindly on a scramble for a fix after having set sail. Such mid-course corrections are often more costly and less optimal than the well planned solution.

Fortunately, accelerating CIFS and MAPI-based applications can counteract the performance-degrading effects of server centralization, and enable clear sailing for you, your users, and the economic and regulatory requirements of your business.

Glossary of Terms

Apdex Performance Regions denote the effective performance quality being provided on the Apdex scale.
Bandwidth is the minimal bandwidth (bits per second) across all the network links between the user and the application server. The slowest link is typically the user's access line to the network. Bandwidth may be reduced by the effects of conflicting traffic (congestion) and protocol efficiency (e.g., TCP window).
Cc (Compute Client) is the total processing time (seconds) required by the client device.
Cs (Compute Server) is the total processing time (seconds) required by the server(s).
F The threshold which defines the boundary between tolerating and frustrated performance zones.
Frustrated Zone Any application response time above F in which the user is very negatively affected by response time.
Payload is information content (bytes) that must be delivered to/from the user's device.
Performance Zone A range of time (between two time values) that a user waits for an application to respond during which his/her perception of the application's responsiveness does not change.
Response Time is defined as the elapsed time (seconds) between a user action (e.g. mouse click, enter, return) and the system response (client, network, server), so the user can proceed with the process. The aggregation of these individual task completion waiting periods defines the "responsiveness" of the application to the user.
RTT is the round-trip-time (seconds) between the user and the application server.
Satisfied Zone The range of application response times between 0 and T in which the user is not affected by the response time.
T The target threshold which defines the boundary between satisfied and tolerating performance zones.
Task Each user-application interface interaction that requires a user entry and an application response.
Tolerating Zone The range of application response times between T and F in which the user is negatively affected by response time.
Turns are the application client-server software interactions (turn count) needed to generate a user-level system response or task (see above). The user is not aware of turns.

NetForecast helps enterprises and vendors understand and improve the performance of networked applications.

Additional information is available at: www.netforecast.com

Peter Sevcik is President of NetForecast and is a leading authority on Internet traffic, performance, and technology. Peter has contributed to the design of more than 100 networks, including the Internet, and holds the patent on application response-time prediction. He can be reached at peter@netforecast.com.

Rebecca Wetzel is an Associate of NetForecast and a 20-year veteran of the data networking industry, with unparalleled inside knowledge of the Internet service and product markets. She works with network product vendors and service providers to develop and implement strategies for new IP-based offerings. She can be reached at rebecca@netforecast.com.