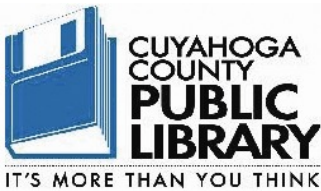




TScale Adds Volume for Cuyahoga County Public Library Application Users



Cuyahoga County Library System (CCPL) is the sixth largest library system in the country. It supports a system of 28 branches located through out the Cleveland, Ohio suburbs.

CCPL's office automation systems consist of four Citrix MetaFrame Servers, each typically supporting more than 50 users. The most heavily used application is Microsoft™ Outlook 2000, and with each additional user, it was becoming dangerously starved for memory.

Jim Kenzig, CCPL's network manager, said, "Memory is almost always one of the most constrained resources in our Terminal Server environment." The library system needed the ability to provide more capacity on existing servers for users and to ensure memory availability for peak periods.

Benefits of Implementing TScale

The results of implementing TScale speak volumes for the library system. "TScale reduced the size of our critical applications," said Kenzig, "freeing up valuable headroom in our servers."

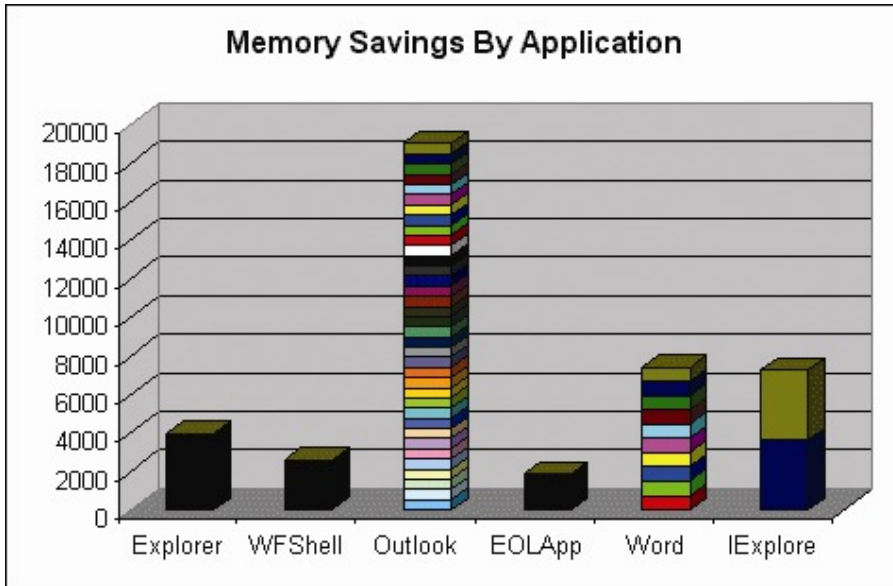
Memory saved by TScale fell into two broad categories. System services like the Citrix Resource Manager (AMGRSvce) and McAfee (MCSHIELD), were reduced in memory footprint significantly. However, since these services are only loaded once on each Terminal Server, their savings are realized only one time.

Savings for applications were another story altogether. For frequently used applications, the number of users at each point in time multiplies the savings for each application.

- Microsoft Outlook 2000 memory utilization was reduced by more than 500K per user session.
- Memory savings of more than 19MB for one server from just Outlook 2000.
- Total saving across all applications was more than 14MB.

Would You like to Evaluate TScale Advanced?

If you would like to evaluate TScale Advanced and help your Terminal Server systems improve utilization of memory, email RTO at sales@rto.com or call 678-987-4302.



The graph shows memory savings for each process in multiple components. Other processes and applications had also been reduced in size, adding to the overall effect of TScale on the system.

Memory Savings by Application

Impact of TScale across sets of applications and number of users.

| Application | Average User | Average Savings (KB) | Total Savings (KB) |
|----------------------|--------------|----------------------|--------------------|
| Amgrsvce | 1 | 872 | 872 |
| MCSHield | 1 | 1,032 | 1,032 |
| Explorer | 53 | 76 | 3,952 |
| Outlook | 36 | 528 | 19,008 |
| WFSHshell | 52 | 50 | 2,600 |
| Eolapp | 26 | 71 | 1,846 |
| Word | 10 | 742 | 7,420 |
| Iexplorer | 2 | 3,636 | 7,272 |
| Total Savings | | | 44,002 |



System Environment

Applications are published and accessed using Citrix ICA Passthrough client and Emergent's AppPortal. Servers are Compaq DL 380s with quad 500 MHz processors and 2 gigabytes of ram.

The production Office Automation environment:

- NT 4.0 TSE SP6
- Citrix Metaframe 1.8 SP3 FR1
- ICA browser hotfix Q269214
- Office 2000 suite
- Outlook 2000
- Project 2000
- Emergent Online AppPortal
- ICA Watchdog

Testing & Implementation

Before implementing TScale, CCPL measured the number of users typically using the most popular applications, creating a real-life benchmark against which to assess the total impact of TScale on the system.

RTO and CCPL tested and evaluated TScale using the following steps:

1. Implemented user interface as an MMC console. Downloaded MMC from Microsoft and installed it on CCLP NT servers. Since MMC require reboot after install, this was done one day prior to the TScale install so that the nightly reboot of the terminal servers would complete the process.
2. TScale installation package (only 581 K) was delivered via email.
3. Installed TScale on a production server, without impacting the production server.
4. Analysis of potential memory improvements.
5. Took measurements of both the number of Outlook sessions running during peak utilization period, as well as the average amount of memory that each instance was consuming.
6. Scheduled early morning optimization after the nightly reboot, so users would start working with optimized applications.
7. Ran TScale for several days, each day finding new combinations of applications and their component DLLs to optimize. (Continued nightly scheduled optimization) After 5 days, the server was fully optimized and since CCPL rarely changes server software, TScale analysis service was turned off.
8. Took memory measurements again to compare to the pre-established CCPL baseline.