Dual-core system performance in common business scenarios

For Intel Corporation

Executive summary

Intel Corporation (Intel) commissioned Principled Technologies (PT) to compare the performance in common business scenarios of systems with a dual-core (DC) Pentium 4 processor with systems using other types of Pentium 4 processors. We tested the performance of a wide range of common business scenarios, each of which involved well-known applications and common business functions, on three different processor/platform combinations:

- New business DC platform: Intel Pentium D 820 (2.8-GHz DC processor) on the Intel D955XBK
- Representative 2004 professional platform: Intel Pentium 4 540 (3.2 GHz with HT Technology) on the Intel D915GAV
- Representative roughly three-year-old platform: Intel Pentium 4 2.0 GHz on the Intel D845GVFN

Key findings

- Intel's dual-core technology delivered performance improvements when running multiple applications, even with a lower-speed processor than some of the comparison units.
- Dual-core technology resulted in even greater responsiveness wins than HT Technology.
- Frequently, the improvements were in both responsiveness (foreground) and throughput (total work completed) times.
- The performance improvements over older computers, such as one using a Pentium 4 2.0 GHz processor, were even more dramatic.

These systems represent, respectively, three business platforms: today's premium Intel Dual-Core platform, Intel's mainstream platform from last year, and one typical of the installed base of three-year-old systems.

We selected four scenarios and created scripts from them that allowed us to automate performance testing. We varied the number, type, and computational intensity of the tasks in those scenarios to match a range of user experiences.

We found that the DC platform, even though its processor was running at a lower speed (2.8 GHz) than the fastest other system (3.2 GHz), can yield significant performance advantages for users running multiple applications simultaneously. Our test scripts showed performance increases for the Pentium D 820 system over the Pentium 4 540 system from 9 percent to 53 percent in different applications. These improvements affected both response times (generally in the foreground task, the one the user is currently waiting on) and in throughput (total work completed, which is often largely a function of the background task or tasks). Foreground task improvements ranged from 16 percent to 53 percent. Background task improvements ranged from 9 percent to 35 percent. (We should note that the discrete graphics adapter in the Pentium D 820 system has the potential to affect overall system performance, which this paper is discussing.)

When we compared the Pentium D 820 system to the older Pentium 4 2.0GHz system, the improvements were even more dramatic: as high as almost 800 percent for foreground tasks, and over 400 percent for background ones. Some of the dramatic wins clearly exceed what one would expect from the combination of the increase in processor core clock speed and the addition of a second core. The basic reason for these huge wins is that the older system became so swamped with work in some of these scenarios that it could not effectively run all the tasks in parallel. Instead, it ended up running some of the tasks serially. So, for example, a foreground task that started shortly after a background task might not actually finish until the background had completed. These delays, which are ones real users would also experience, vanish with the ability of the DC and HT Technology processors to handle more simultaneous work. Many of the foreground and background task improvements translated into multi-second time savings that would be obvious to users. In one case, a foreground task that completed in 13.4 seconds on the dual-core system took over 119 seconds on the Pentium 4 2.0 GHz system. This big difference is due to the fact that, on the PC with the Pentium 4 2.0

GHz processor, the foreground and background tasks completed at about the same time, while on the PCs with DC and HT Technology processors the foreground task completed well before the background tasks.

We used a variety of common enterprise business applications in our testing and the resulting application scenarios. Enterprise applications we used in this study include the following (in alphabetical order):

- Adobe Acrobat 7.0.3 Standard
- Microsoft AntiSpyware (Beta 1)
- Microsoft Office Access 2003 (Service Pack 1)
- Microsoft Office Excel 2003 (Service Pack 1)
- Microsoft Office PowerPoint 2003 (Service Pack 1)
- Microsoft Office Word 2003 (Service Pack 1)
- Microsoft Office Outlook 2003 (Service Pack 1)
- Microsoft Visual C++ .NET 2003
- Network Associates' McAfee VirusScan Enterprise version 8.0i
- Skype Technologies Skype Version 1.3.0.60

Intel specified the general system types and supplied two of the motherboards. We purchased the older motherboards and the processors (as well as such common components as RAM, hard disks, and optical drives), and we built the systems. Each test system had the following basic components (see Appendix A for full configuration details):

- 512MB of the fastest RAM its motherboard supported
- 80GB hard disk with an 8MB buffer (SATA or EIDE depending on what the system could support)
- DVD-RW optical drive
- Windows XP Professional with SP2

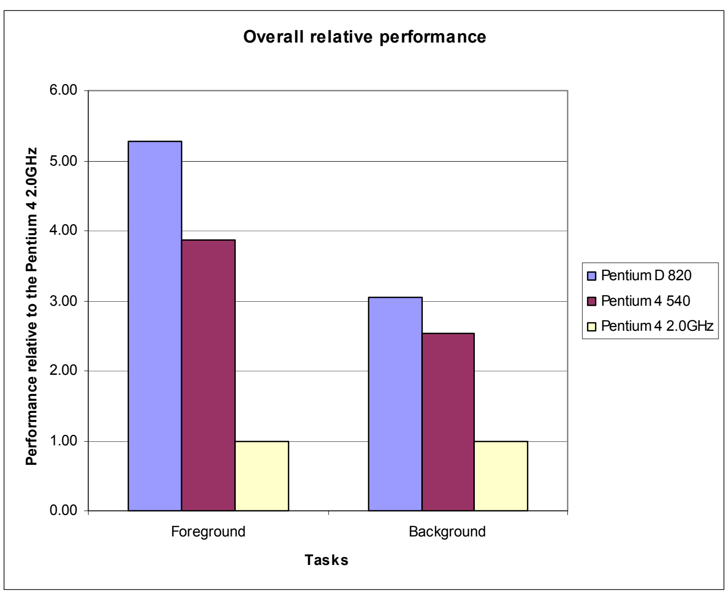


Figure 1: Overall performance, relative to that of the PC with a Pentium 4 2.0 GHz processor, on all of the scenarios' foreground and background tasks for all three systems. Bigger is better.

The application scenarios we tested on these systems include such common functions as opening messages in Outlook, scanning for viruses, converting documents to PDF format, conducting VoIP conference calls, and compiling C++ programs. To make it easier to test our scenarios repeatedly and accurately, we automated the hand-timed application functions using scripts we wrote in IBM's Rational Visual Test 6.5. In our analyses we use the times from our tests of those scripts. Figure 1 shows the overall response time win (foreground task) and the win in background task completion time, relative to the PC with a Pentium 4 2.0 GHz processor, across all the scenarios for each processor. These amalgamated results show how dramatically dual-core technology and HT Technology can improve performance over older processors. Figure 2 uses the same data but focuses solely on the systems with the Pentium D 820 and the Pentium 4 540 processors; those results are relative to those of the system with the Pentium 4 540 processor. As this chart shows, even though the DC Pentium D 820 processor is running at a lower clock speed its DC technology provided substantial performance boosts over the higher clock speed, HT technology system.

In the following sections we look more closely at our test application scenarios (Application scenarios), provide some useful basics about DC technology (Dual-core technology background), examine the results of our tests (Test results and analysis), and give detailed information about how we actually tested (Test methodology). In the appendices we present the configurations of the test systems, explain how to manually execute the application functions in our scenarios, and discuss some issues in the development of the test scripts.

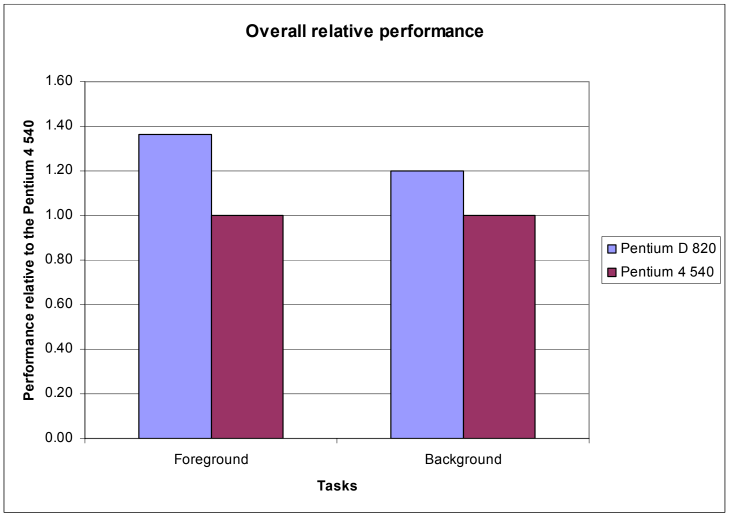


Figure 2: Overall performance, relative to that of the PC with the Pentium 4 540 processor, of the PCs with the Pentium D 820 and the Pentium 4 540 processors, on all of the scenarios' foreground and background tasks. Bigger is better.

Application scenarios

To analyze the types of performance improvements a DC processor can yield, we developed a set of common business application scenarios using typical applications and operations. We concentrated on typical office productivity applications, such as those in Microsoft Office 2003, but we also included a small range of other applications, such as voice conferencing and software development. We looked primarily at operations that are processor-intensive, because those are the ones on which users are typically likely to be waiting and thus appreciate performance improvements.

After looking at a variety of scenarios, we decided on four we felt represented a range of common user activities involving common business functions.

In the first scenario, Una, a business manager of a small software company, uses Microsoft Excel to prepare forecasting reports. She needs to prepare her monthly forecast before leaving for the weekend. She is running late, and the weekly scheduled virus scan starts. While the scan is running she makes a change to her forecast and does a recalculation on a formula-rich Excel forecast spreadsheet.

The second scenario involves Duane, a sales associate who has created a sales presentation in PowerPoint and an accompanying document in Word. He launches Word and starts a conversion of his report into PDF format for distribution to his client. He also needs to make a last-minute change to one of his slides before his lunch presentation. Just as he is about to launch PowerPoint, his system starts running an AntiSpyware scan. While the scan runs and the PDF conversion continues, he opens his presentation and waits for PowerPoint to display the thumbnail images of his slides before selecting the one he wants to edit.

The third scenario involves Kendra, a marketing manager. She has a Skype VoIP conference call scheduled with her three-person marketing team. After the call starts, Kendra realizes that she needs to get some historical competitive sales data from an Access database and make summaries available as an XML document for the team to share. While she is getting that data ready with Access, she opens an important email message that one of the other team members sent her earlier.

In the fourth scenario, Trey, a lead developer, arranges a VoIP conference call with his development team to discuss the state of the current project. While the rest of his team is getting ready for the call, Trey starts Visual C++ builds of the two main parts of the project. While waiting for the builds to finish, he compares the current version of the project's functional specification with the one from the previous meeting and merges the two using Word's compare and merge capabilities. He then begins the meeting in earnest.

For more details on how we executed and measured these scenarios, our specific test functions, and the files the scenarios involved, see Appendix B.

Dual-core technology

Before we discuss and analyze our results in detail, some basic information about dual-core processors is useful. The fundamental concept behind DC processors is relatively simple to state: Put two processors (or, more precisely, two complete processor cores) on a single chip. The resulting chip gives a system two processors available to do work. (In Intel's current DC architecture, these two processors share system RAM but have separate L2 caches.)

Most individual business applications today are not designed to take regular advantage of multiple processors. So, typical business users will primarily see performance gains from DC processors when they're running multiple applications simultaneously. In our experience, most users switch among multiple open applications, so the opportunities for performance improvements are many. Adding to those opportunities are the increasing needs to have such demanding applications as anti-virus programs, spyware blockers, and backup systems running as background programs during times users are trying to get their work done. As more and more users run PC-based communication applications, such as voice conferencing, and conduct meetings with those applications, the opportunity for performance improvements from DC processors also grows. Finally, as DC processors become more common, we can also reasonably expect to see more and more applications take advantage of their capabilities.

It's tempting to see DC technology as the next step after Hyper-Threading Technology: a move from two logical processors on a chip (HT Technology) to two physical processors (DC Technology). These two technologies, however, are not necessarily competing; they can be complementary. As Intel's Extreme Edition DC processors show, you can have a system with a DC processor, each of whose cores has two logical processors. Put a different way, though two physical processors are usually better than two logical ones, four logical processors with two underlying physical ones are likely to be even better. Intel's Pentium D processors, such as the one we used in this paper, do not support HT technology, though as we noted above, the Pentium Extreme Edition does.

When you're running multiple applications, DC processors have the ability to improve both system throughput—the amount of work the system performs in a given unit of time—and system response time—how quickly the system responds to user commands in applications the user is currently running. Overall system throughput can improve, as our results demonstrate, because the two processor cores have the potential to accomplish more work per unit of time than a single core. Response time can improve, as our results also demonstrate, because the task on which the user is waiting can have as much as a full processor core available to service it. Our application scenarios illustrate a variety of combinations of tasks that show real performance gains when running on DC systems.

Finally, we must note that DC processors will not, of course, always improve system performance. When each of the application tasks running concurrently does not fully subscribe the processor, a single processor, particularly one with HT Technology, may perform on par with a DC processor.

Test results and analysis

In this section we examine the results of tests with the application scripts we created. Each of the times we present in this section and in the corresponding graph for each scenario is the median, in seconds, of five runs. The following table shows the median times on each scenario for each of the test systems.

System processor and platform	Intel Pentium D 820 (2.8GHz Dual core) on Intel D955XBK	Intel Pentium 4 540 (3.2GHz) on Intel D915GAV	Intel Pentium 4 2.0GHz on Intel D845GVFN
Scenario 1			
Excel recalc (FG)	58.500	89.375	163.000
McAfee VirusScan (BG)	92.063	113.469	223.797
Scenario 2			
Open PowerPoint (FG)	13.360	20.406	119.609
Microsoft AntiSpyware (BG)	33.672	39.390	119.875
Word to PDF conversion (BG)	65.781	71.734	98.813
Scenario 3			
Outlook open (FG)	11.203	14.204	95.750
Access/XML report (BG)	57.484	63.062	148.922
Scenario 4			
Word/XML compare (FG)	40.359	47.016	146.968
Visual C++ compile #1 (BG)	28.812	38.765	120.235
Visual C++ compile #2 (BG)	19.984	25.406	112.406

Figure 3: Median results (of five runs) for all three test systems on all tasks of all four scenarios. (FG) indicates a foreground task and (BG) indicates a background task.

In the following sub-sections we explore further the results of the tests using each scenario. To make comparisons easier, in these discussions we always show normalized results in which we compare each system's time to the time the system with the Pentium 4 2.0 GHz processor took to do the task at hand. The result for the PC with the Pentium 4 2.0 GHz processor is thus always 1.00, because it's the comparison basis. Results higher than 1.00 indicate how much faster a system is than the Pentium 4 2.0 GHz system. Because of the normalization, larger results are better. So, for example, a result of 1.80 would mean the system in question was 80 percent faster than the Pentium 4 2.0 GHz system on the scenario in question.

For ease of reference, we also show comparisons between the results of the DC system (the Pentium D 820) and the HT system (the Pentium 4 540). In these comparisons, we normalize to the results of the Pentium 4 540, so that system's score is always 1.00. Again, larger results are better.

As part of our discussion of each scenario, we show some processor utilization curves that we captured with Windows XP's Perfmon utility. Our goal for each of these curves is to illustrate how demanding each task would be if run alone on a modern system. In multitasking scenarios, the system faces each of these same demands, but all at the same time from all the tasks that are running simultaneously. When all the simultaneous tasks have minimal processor requirements, keeping them all executing smoothly in parallel may not be difficult. When, by contrast, one or more of the tasks has significant processor demands, meeting those demands at the same time can be extremely difficult for a processor without the added execution ability of an HT Technology or DC processor. We chose to capture each task's processor-demand curve on the Pentium 4 540 system with HT Technology turned off. We chose the Pentium 4 540 system because it was a modern platform but not the main focus of this test; had we captured the processor utilization on the oldest system in this report we might have given the impression of trying to overstate the processor demands of the tasks. We chose to turn off HT Technology to make it easy to see the processor demands; had we captured the processor utilization with HT Technology enabled, Perfmon would have capped each logical core at 50 percent utilization. In all the processor utilization charts, we use a blue line for foreground tasks and a red line for backgrounds.

Application scenario 1

This scenario uses a formula-rich 445KB Microsoft Excel spreadsheet. It recalculates those formulas (the foreground task) while Network Associates' McAfee VirusScan does a scan on a large (86.8MB) Windows XP directory (the background task). The two tasks in this scenario are both very demanding ones that will use most if not all of the processor. (See Figures 6 and 7.)

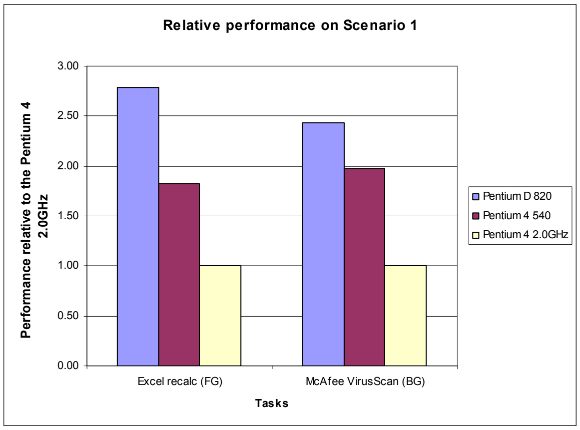


Figure 4: Performance , relative to the Pentium 4 2.0 GHz system, of each of the three systems on a pair of concurrent tasks. Bigger is better.

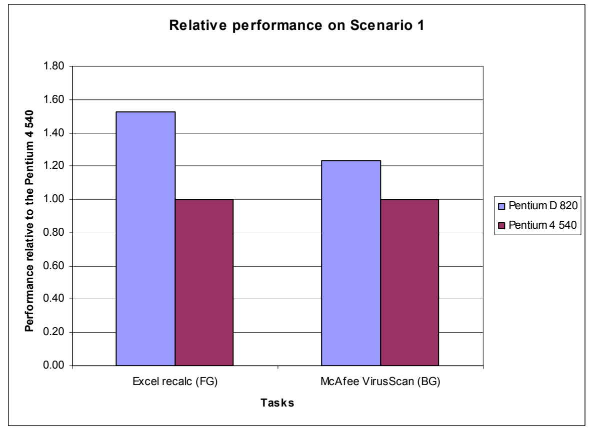


Figure 5: Performance, relative to the Pentium 4 540 system, of the Pentium D 820 and the Pentium 4 540 on a pair of concurrent tasks. Bigger is better.

Figure 4 shows the improvement in performance of both the foreground and background tasks over the Pentium 4 2.0GHz system. These results show that the DC system delivered very large performance improvements of 179 percent and 143 percent, respectively, for the foreground and background tasks. Not only are these large percentage improvements, they also represent big absolute time improvements: from 163.0 to 58.5 seconds on the foreground task, and from 223.8 to 92.1 seconds on the background task.

Figure 5 shows the same results, but this time comparing the DC Pentium D 820 to the Pentium 4 540. This view of the results highlights the performance gain a DC system can yield over current systems. Here, the performance improvements remain impressive: 53 percent and 23 percent, respectively, for the foreground and background tasks. The difference in the time to complete these tasks improves from 89.4 to 58.5 seconds on the foreground and from 113.5 to 92.1 seconds on the background. These are large time savings any user would appreciate.

The processor utilization curves in Figures 6 and 7 help illustrate the reason for these time savings. As Figures 6 and 7 show, both Scenario 1 tasks are able to take advantage of all the power of a processor, so the second core in the Pentium D 820 system provided a huge benefit. The HT Technology processors can

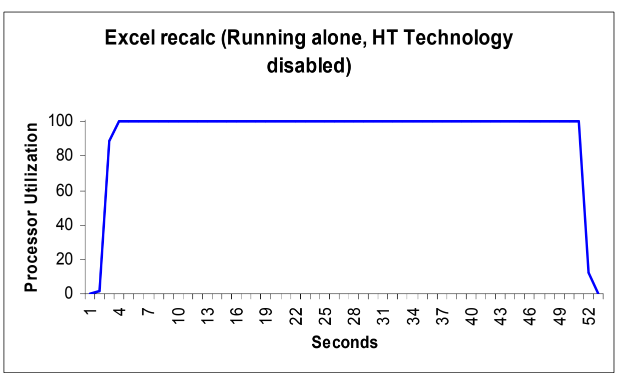


Figure 6: System processor utilization of the Scenario 1 foreground task (an Excel recalc) running alone.

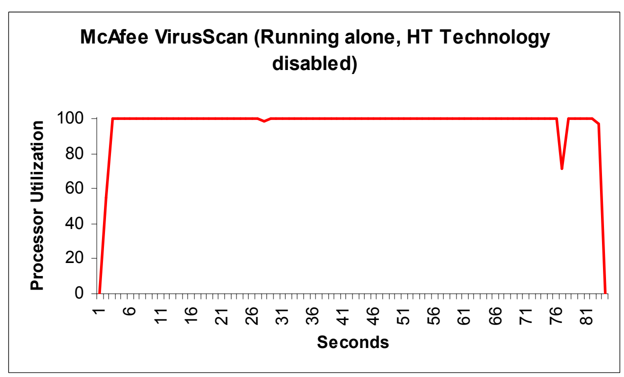


Figure 7: System processor utilization of the Scenario 1 background task (a virus scan using McAfee VirusScan) running alone.

also help somewhat with this kind of workload, as you can see from the significant improvement they yielded over the non-HT-Technology Pentium 4 2.0GHz system.

Application scenario 2

This scenario opens a 33.4MB PowerPoint presentation (the foreground task) while Microsoft AntiSpyware does a scan on a large Windows XP directory (the first background task) and Adobe Acrobat converts a 3.7MB document to PDF format (the second background task). None of the tasks in this scenario is quite as demanding as those in Scenario 1. The two background tasks, however, are fairly demanding.

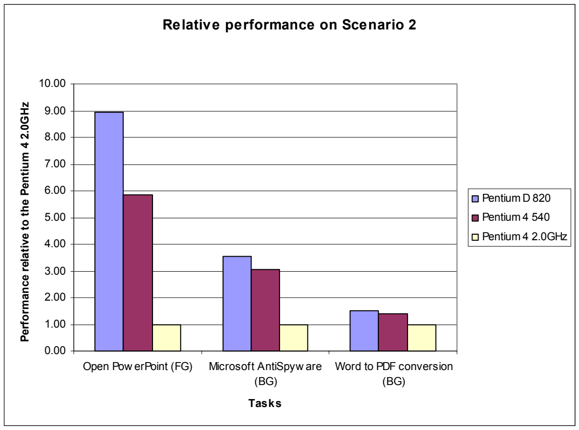


Figure 8: Performance, relative to the Pentium 4 2.0GHz system, of each of the four systems on three concurrent tasks. Bigger is better.

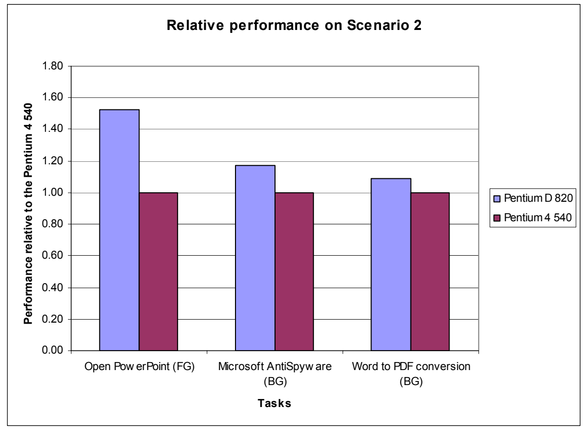


Figure 9: Performance, relative to the Pentium 4 540 system, of the Pentium D 820 and the Pentium 4 540 on three concurrent tasks. Bigger is better.

Figure 8 shows the improvement in performance of both the foreground and background tasks over the Pentium 4 2.0 GHz system. The foreground improvement is very large, with the Pentium 4 2.0 GHz taking significantly longer to complete the task: the Pentium D 820 system completed the PowerPoint open in 13.4 seconds, while the Pentium 4 2.0GHz took 119.6 seconds. As we mentioned in the Executive summary, improvements of this magnitude typically come from the older system being unable to run the tasks in parallel and so having the foreground task unable to complete until the background task finishes. That was the case here. The figures also show two background task improvements of 256 percent (AntiSpyware) and 50 percent (Word to PDF conversion).

As Figure 9 shows, the DC Pentium D 820 opens the PowerPoint presentation noticeably faster (13.4 seconds) than the Pentium 4 540 (20.4 seconds). On the background tasks, the results are not as dramatic, but still show the advantage of the DC Pentium D 820. The AntiSpyware task runs in 33.7 seconds rather than 39.4 seconds, while the Word to PDF conversion improves from 71.7 seconds to 65.8 seconds on the DC system.

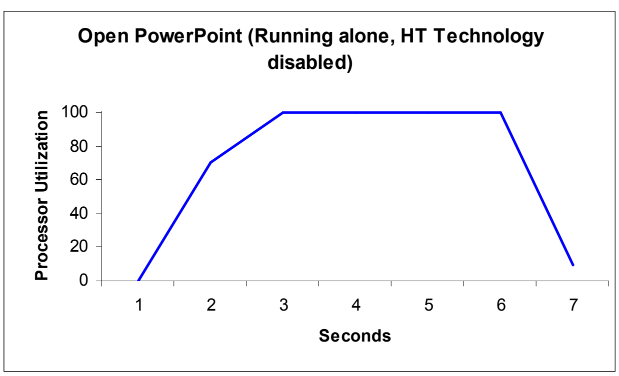


Figure 10: System processor utilization of the Scenario 2 foreground task (opening a PowerPoint presentation) running alone.

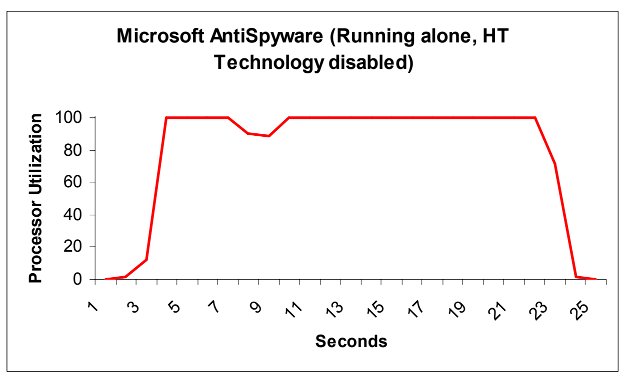


Figure 11: System processor utilization of one of the Scenario 2 background tasks (Microsoft AntiSpyware scanning for spyware) running alone.

The processor utilization curves in Figures 10, 11, and 12 help explain the performance results. The two background tasks here demand a full processor most, but not all, of the time. This usage level probably allows

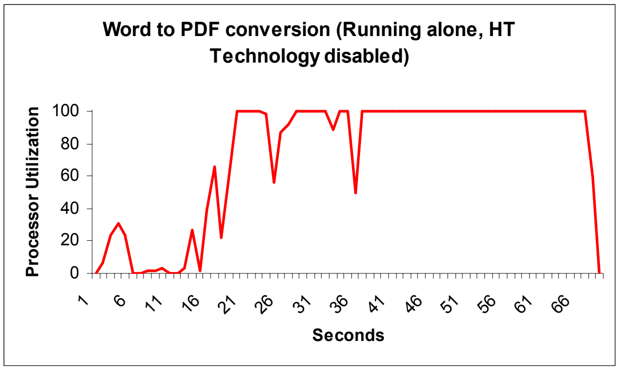


Figure 12: System processor utilization of one of the Scenario 2 background tasks (converting a Word document to PDF format) running alone.

the processors with HT Technology to do a decent job of servicing those two tasks. The HT Technology processors cannot, however, also provide good service for the PowerPoint open foreground task. The non-HT-Technology Pentium 4 2.0 GHz system is unable to cope with this level of load and ends up not running the foreground until one of the backgrounds has finished, which means it takes much longer to complete all the work. This kind of workload shows both how HT Technology can be of benefit and how a DC system can provide even greater performance improvements as the amount of simultaneous work increases.

Application scenario 3

This scenario opens Outlook and then displays an encrypted message within Outlook (the two actions combined are the foreground task we timed) while Microsoft Access produces an XML report on an 11.9MB database (first background) with a Skype VoIP conference call running at the same time (second background). Like all other tasks, the Skype workload was exactly the same on all the systems. We set up a multi-person conference call, initiated that call from the system under test, and then had Skype running, but with no audio load. Even though this no-audio call is obviously the least demanding Skype test possible, and though real people would be talking and thus increasing the load on the system, this setup both showed the performance differences in the system and let us ensure a perfectly consistent workload on all the test systems. Because Skype is running constantly during the test, we did not time it.

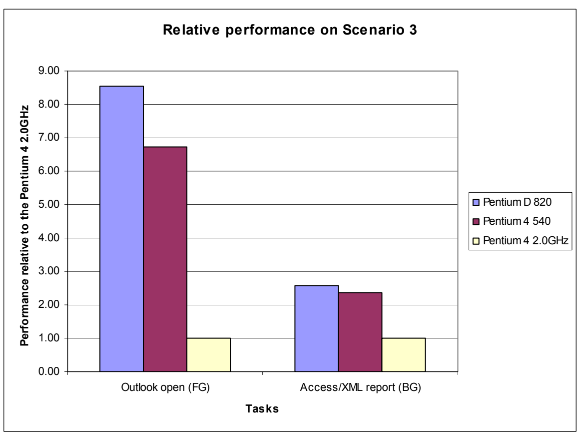


Figure 13: Performance, relative to the Pentium 4 2.0GHz system, of each of the four systems on a pair of concurrent tasks. Bigger is better.

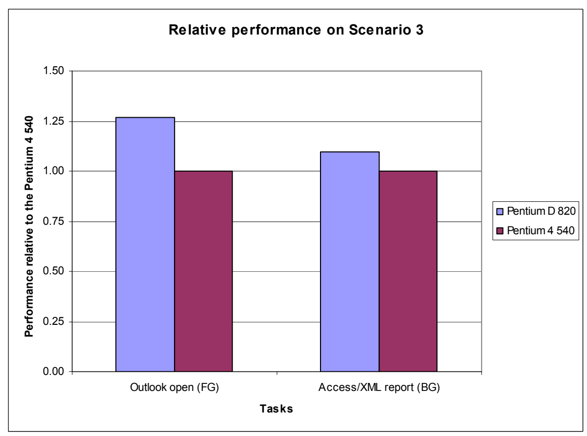


Figure 14: Performance, relative to the Pentium 4 640 system, of the Pentium D 820 and the Pentium 4 640 on a pair of concurrent tasks. Bigger is better.

Even more than in Scenario 2, these tasks vary a great deal in their demands on the processor; only the Access/XML report task really demands the whole processor.

Figure 13 shows the performance improvement of both the foreground and background tasks over the Pentium 4 2.0 GHz system. These results show very large performance improvements—over 700 percent and 150 percent, respectively—for the foreground and background tasks. These large percentage improvements represent absolute time improvements from 95.8 to 11.2 seconds on the foreground task and from 148.9 to 57.5 seconds on the background one. As we noted in the Executive summary, improvements of this magnitude in foreground tasks typically reflect the inability of older system to run the tasks in parallel. Thus, on those systems the foreground task cannot finish until the background task is complete or approaching completion. Figure 14 shows the same results but compares only the DC Pentium D 820 and the Pentium 4 540. This view of the results highlights the performance gain a dual-core processor system can have versus current systems. Here, the performance improvements are about 27 percent and 10 percent, respectively, for the foreground and background tasks. The time to complete the foreground task drops from 14.2 (Pentium 4 540) to 11.2 (Pentium D 820) seconds, and 63.1 to 57.5 seconds on the background.

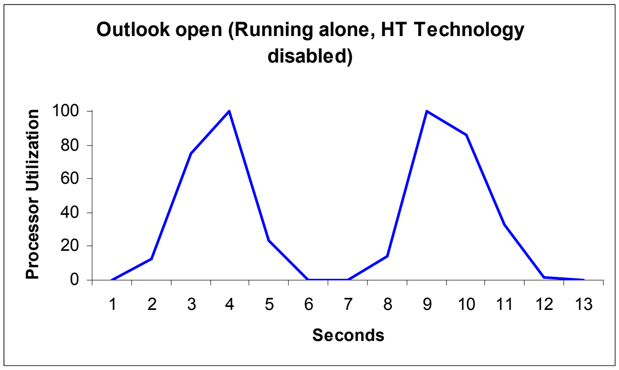


Figure 15: System processor utilization of the Scenario 3 foreground task (opening Outlook and opening a message) running alone.

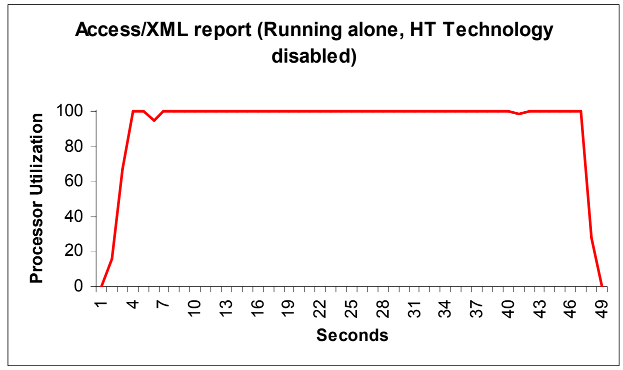


Figure 16: System processor utilization of one of the Scenario 3 background tasks (generating an XML report from Access) running alone.

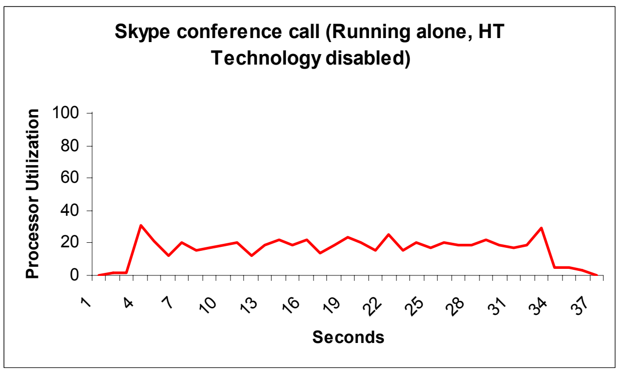


Figure 17: System processor utilization of one of the Scenario 3 background tasks (hosting a Skype conference call) running alone.

The processor utilization curves in Figures 15, 16, and 17 illustrate the processing demands the test systems faced. In this scenario, the Skype VoIP conference call requires a noticeable amount of the processor simply to keep the conference call open and to have the codec ready to process voices. The Access/XML report demands as much of the processor as it can get. Between those two demands, there is little capacity left on a non-DC system for the relatively mild processor requirements of the Outlook foreground task. The systems with processors with HT Technology handled this foreground task much better than the non-HT-Technology Pentium 4 2.0 GHz system. The need to run all three tasks (even though only one of them has a very high processor requirement) at the same time is what allows the second core in the Pentium D 820 system to show an improvement over the higher-clock speed, single-core, HT Technology processors.

Application scenario 4

In this scenario we open a 1.3MB XML document in Word and then compare and merge that document with an 11.2MB XML document (the foreground task), while a pair of Visual C++ project builds are executing (first and second backgrounds) with a Skype VoIP conference call running at the same time (third background). Like all other tasks, the Skype workload was exactly the same on all the systems. We set up a multi-person conference call, initiated that call from the system under test, and then had Skype running, but with no audio load. Even though this no-audio call is obviously the least demanding Skype test possible, and though real people would be talking and thus increasing the load on the system, this setup both showed the performance differences in the system and let us ensure a perfectly consistent workload on all systems. Because Skype is running constantly during the test, we did not time it.

As in Scenario 3, the tasks in this scenario put varying loads on the server. This scenario, however, adds an additional background task.

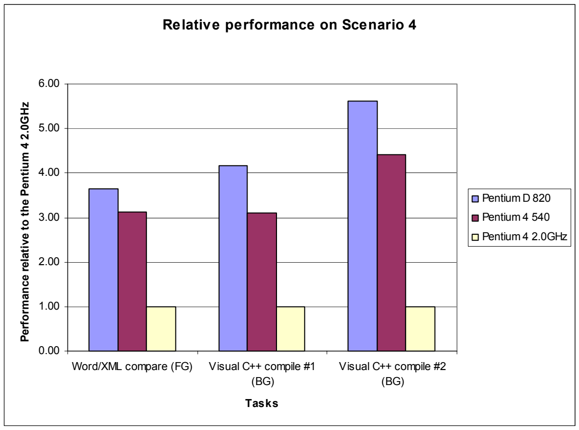


Figure 18: Performance, relative to the Pentium 4 2.0GHz system, of each of the three systems on three concurrent tasks. Bigger is better.

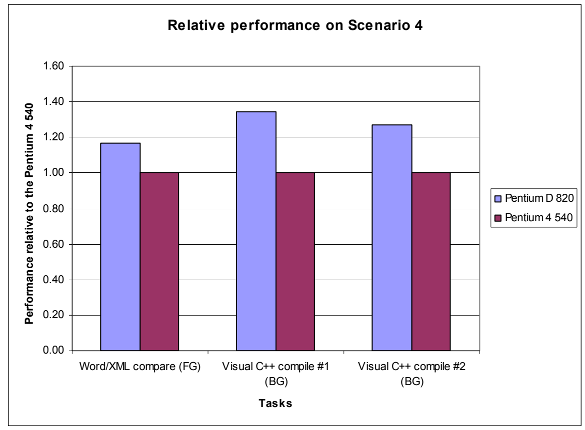


Figure 19: Performance, relative to the Pentium 4 540 system, of the Pentium D 820 and the Pentium 4 540 on three concurrent tasks. Bigger is better.

Figure 18 shows the improvement in performance of both the foreground and background tasks over the Pentium 4 2.0 GHz system. The improvements are obviously very large: 264 percent, 317 percent, and 462 percent, respectively, for the foreground, first background, and second background tasks. These large percentage improvements also represent big absolute time improvements: from 147.0 to 40.4 seconds on the foreground task, from 120.2 to 28.8 seconds on the first background, and from 112.4 to 20.0 seconds on the second background. Figure 19 shows the same results, but this time comparing only the DC Pentium D 820 system and the Pentium 4 540 unit. The performance improvements are still significant: 16 percent, 35 percent, and 27 percent, respectively, for the foreground, first background, and second background tasks. The time to complete these tasks improves from 47.0 to 40.4 seconds on the foreground, 38.8 to 28.8 seconds on the first background, and 25.4 to 20.0 seconds on the second background.

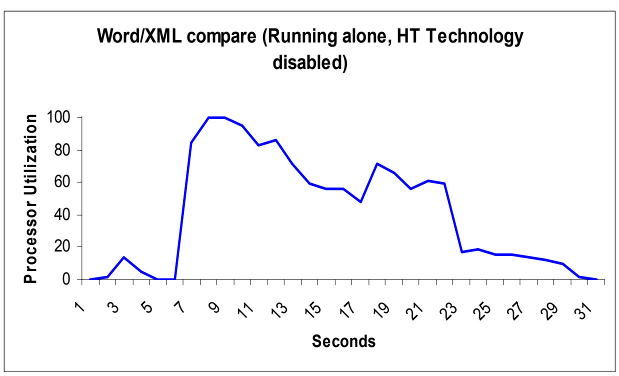


Figure 20: System processor utilization, without HT Technology, of the foreground task (Word/XML compare) running alone.

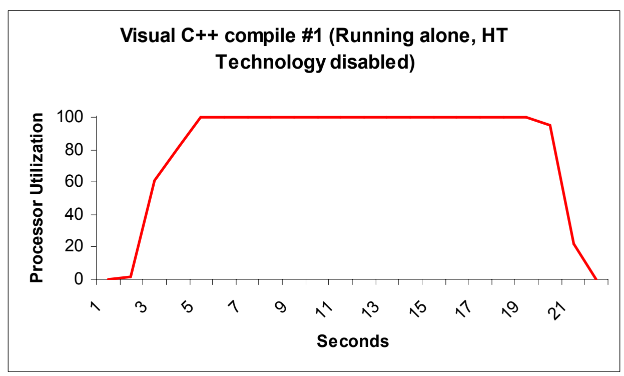


Figure 21: System processor utilization of one of the Scenario 4 background tasks (compiling a Visual C++ project) running alone.

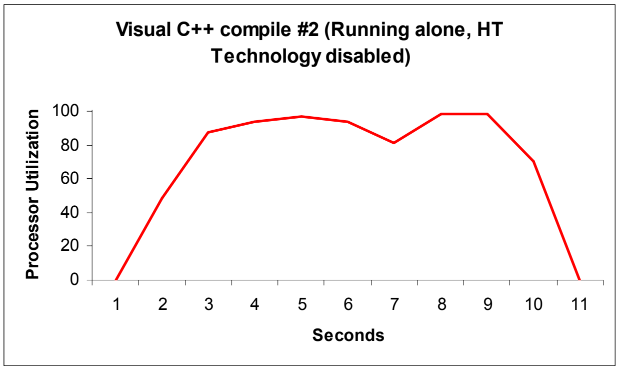


Figure 22: System processor utilization of one of Scenario 4 background tasks (compiling a second Visual C++ project) running alone.

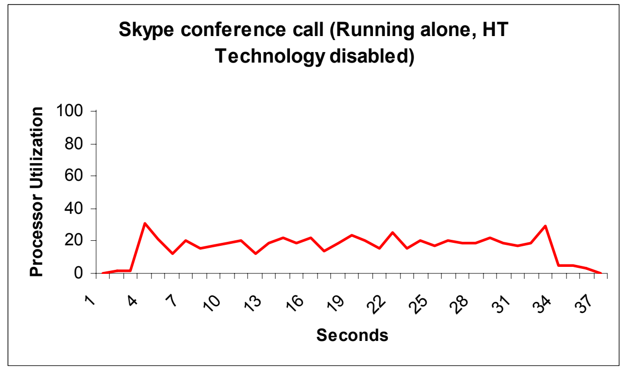


Figure 23: System processor utilization of one of the Scenario 4 background tasks (hosting a Skype conference call) running alone.

The processor utilization curves in Figures 20, 21, 22, and 23 give a general idea of what is happening. As in Scenario 3 and for the same reasons, the Skype VoIP conference call (Figure 23) requires a noticeable amount of the processor. The two background Visual C++ builds (Figures 21 and 22) show different processor requirements. The foreground Word operations (Figure 20) also vary over time in their processor demands. Put all of these tasks on the same system at the same time, however, and they can slow dramatically, as they did on the non-HT-Technology Pentium 4 2.0 GHz system. HT Technology helped the Pentium 4 540 do a reasonable job of handling this set of tasks with relatively uneven processor demands, but again, as in the other scenarios, the second core in the Pentium D 820 system provided even greater benefit.

These application scenario results show that DC technology is likely to yield performance improvements to a wide range of business users, particularly those on typical three-year-old systems, when multiple processor-intensive tasks are running at the same time.

Test methodology

We evaluated the performance of each of the application scenarios (see "Application scenarios") both by hand and with automated test scripts, which we developed with IBM's Visual Test 6.5. Appendix B details the steps we followed when we hand-timed the scenarios. In this paper, we concentrate most of our discussions on the results of the automated scripts, because those results are generally more repeatable than hand timings.

We collected results for five runs of each script in each system configuration. We refer in this paper only to the median results of those runs on each system/processor configuration. The scripts produce times (in milliseconds), where lower times to perform a given function are better.

Appendix A provides detailed configuration information on the three systems. We set up each of those systems using the following process:

- Install Microsoft Windows XP Professional.
- Using the standard Microsoft Windows Update Web site, apply all current (as of August 19, 2005) Windows XP critical updates, including Windows XP SP2, except optional updates, such as Windows Media Player 10 or Windows Media Connect, that were totally unrelated to the goal of this paper.
- Install Office 2003.
- Using the standard Microsoft Office Update Web site, apply all current (as of August 19, 2005) Office 2003 updates.
- Turn off Windows Service Pack 2 Security Center Pop-up Alerts. Control Panel->Security Center->Click "Change the way Security Center Alerts me" on the left->Uncheck Firewall, Automatic

Updates, and Virus Protection. This prevents issues with such alerts occurring during testing and affecting results.

- Turn off Windows Automatic Updates: Control Panel->Automatic Updates->Select “Turn off Automatic Updates”. This prevents updates from occurring during testing and affecting results.
- To ensure as consistent a starting point as possible for our performance measurements, defragment the disk of each system.
- Using Symantec’s Ghost utility, make an image of each system’s hard disk (so we could return to that clean image whenever necessary in our testing).
- Install the necessary additional software (versions current as of August 19, 2005) for each of the four scenarios:
    - Scenario 1
        - Network Associates’ McAfee VirusScan Enterprise version 8.0i
    - Scenario 2
        - Adobe Acrobat 7.0.3 Standard
        - Microsoft AntiSpyware (Beta 1)
    - Scenario 3
        - Skype Technologies Skype Version 1.3.0.60
    - Scenario 4
        - Microsoft Visual C++ .NET 2003
        - Skype Technologies Skype Version 1.3.0.60
- Using Symantec’s Ghost utility, make an additional image of each system’s hard disk for each scenario (so we could if necessary return to that clean image with the installed Office and additional applications).

To obtain the results for each application scenario test result we cite here, we followed the same basic process:

1. Reboot the system.
2. Wait ten seconds after the Windows hourglass disappears and Windows XP has completed its startup sequence. We wait to make sure the system is in a consistent starting state.
3. Run the test script (or hand-timed application functions, as appropriate).
4. Record the result.
5. Repeat the above three-step process five times. (If any test or script failed, we discarded that test’s results and ran the test again.)
6. Take the median result of the five iterations as the representative of the group.

Appendix A: Test system configuration information

In this appendix, we provide in-depth configuration information for the three test systems.

We installed the applications necessary to run the scenarios on each system using the same steps. Because IT organizations often employ standard configurations and to make the systems as comparable as reasonably possible, we installed on each system only those applications we needed for our testing. We applied all Windows critical updates, including SP2, available as of August 19, 2005, except those optional updates unrelated to enterprise systems, such as Windows Media Player 10 or Windows Media Connect. We applied all the application updates, such as those for Microsoft Office 2003, that were available as of the same date. We then defragmented the hard disk of each system and saved the resulting disk image. We performed all tests on these "clean" disk images.

This table provides detailed configuration information about each of the three test systems.

System	Intel Pentium D 820 (2.8GHz Dual core) on Intel D955XBK	Intel Pentium 4 540 (3.2GHz) on Intel D915GAV	Intel Pentium 4 2.0GHz on Intel D845GVFN
General			
Processor and OS kernel: (physical, core, logical) / (UP, MP)	1P2C2L / MP	1P1C2L / MP	1P1C1L / UP
Number of physical processors	1	1	1
Single/Dual Core processor(s)	Dual	Single	Single
Processor Hyper-Threading Technology Status	N/A	On	N/A
System Power Management Policy (AC/AlwaysOn, AC/Adaptive, DC/AlwaysOn, DC/Adaptive, Other)	AC/AlwaysOn	AC/AlwaysOn	AC/AlwaysOn
CPU			
Segment (Server, Desktop, Mobile)	Desktop	Desktop	Desktop
Vendor	Intel	Intel	Intel
Name	Pentium D	Pentium 4	Pentium 4
Model number	Pentium D 820	Pentium 4 540	Pentium 4
Stepping	4	1	9
Socket type and number of pins	Socket T/ LGA775	Socket T/ LGA775	Socket 478/ mPGA-478
Core frequency	2800	3200	2000
Front-side bus frequency	800	800	400
L1 Cache	16KB + 12KB (1)	16KB + 12KB	8KB + 12KB
L2 Cache	1MB (1)	1MB	512KB
Platform			
Vendor and model number	Intel D955XBK	Intel D915GAV	Intel D845GVFN
Motherboard model number	D955XBK	D915GAV	D845GVFN
Motherboard chipset	i955XE	i915G	i845GV
Motherboard revision number	AAC96732-405	AAC64137-300	AAC77646-105
Motherboard serial number	ABBK52000110	BQAV41800231	FCFN44613282
BIOS name and version	Intel BK95510J.86A.1784.2005.0715.1529	Intel EV91510A.86A.0444.2005.0429.2108	American Megatrends FN84510A.86A.0018.P05.0501301720
BIOS settings	Setup Default	Setup Default	Setup Default
Chipset INF driver	7.0.0.1025	7.0.0.1025	4.04.1007
Memory module(s)			

Vendor and model number	Micron Technology CM32M6408U26AD2F -3VC	Viking VI4CU326428DTNL1	Spectek s80032vmctl
Type	PC5300 DDR2-SDRAM	PC3200 DDR-SDRAM	PC2700 DDR-SDRAM
Speed in the system currently running @ (MHz)	533	400	333
Timing/Latency (tCL-tRCD-tRP-tRASmin)	4-4-4-12	3-4-4-8	2.5-3-3-7
Size	512MB	512MB	512MB
Number of sticks	2 x 256MB	2 x 256MB	2 x 256MB
Chip organization (Single-sided, Double-sided)	Single-sided	Double-sided	Single-sided
Channel (Single/Dual)	Dual	Dual	Dual
Hard disk			
Vendor and model number	Western Digital WDC WD800JD-75JNA0	Western Digital WDC WD800JD-75JNA0	Western Digital WDC WD800JB-00JAA0
Size	80GB	80GB	80GB
Buffer Size	8MB	8MB	8MB
RPM	7200	7200	7200
Type	Serial ATA 1.5	Serial ATA 1.5	ATA-100
Controller	Intel 82801GB (ICH7)	Intel 82801FB (ICH6)	Intel 82801DB (ICH4)
Windows XP Microsoft hard disk driver version number	5.1.2535.0	5.1.2535.0	5.1.2535.0
Operating system			
Name	Microsoft Windows XP Professional	Microsoft Windows XP Professional	Microsoft Windows XP Professional
Build number	2600	2600	2600
Service pack	SP2	SP2	SP2
File system	NTFS	NTFS	NTFS
Kernel	ACPI Multiprocessor PC	ACPI Multiprocessor PC	ACPI Uniprocessor PC
Language	English	English	English
Microsoft DirectX version	9.0c	9.0c	9.0c
Graphics			
Vendor and model number	NVIDIA GeForce 6600	Intel 915G	Intel 82845GV
Chipset	GeForce 6600	Intel 82915G Express	Intel 82915GV Express
BIOS version	5.43.02.16.00	Intel Video BIOS 1214	Intel Video BIOS 2759
Type (PCI Express, AGP 8X, Integrated)	PCI Express	Integrated	Integrated
Memory size	256MB	128MB shared	64MB shared
Resolution	1024 x 768 x 32-bit color	1024 x 768 x 32-bit color	1024 x 768 x 32-bit color
Driver	nVidia 7.7.7.7 7/20/2005	Intel 6.14.10.4363	Intel 6.14.10.3762
Sound card/subsystem			
Vendor and model number	SigmaTel High Definition Audio CODEC	Realtek High Definition Audio	Realtek AC'97 Audio
Type (PCI, Integrated)	Integrated	Integrated	Integrated
Driver	SigmaTel 5.10.4647.0	Realtek 5.10.0.5136	Realtek 5.10.0.5870

Network card/subsystem			
Vendor and model number	Intel PRO/1000 PM Network Adapter	Intel PRO/100 VE Network Adapter	Intel Pro/100 VE Network Adapter
Type (PCI, Integrated)	Integrated	Integrated	Integrated
Driver	Intel 9.0.15.0	Intel 7.1.12.0	Intel 6.4.14.0
Optical drive			
Vendor and model number	Lite-On DVDRW SHOW-1673S (CD 48X Rd, 48X Wr) (DVD 6X Rd, 6X Wr)	Lite-On DVDRW SHOW-16735 (CD 48X Rd, 48X Wr) (DVD 6X Rd, 6X Wr)	Lite-On DVDRW SHOW-16735 (CD 48X Rd, 48X Wr) (DVD 6X Rd, 6X Wr)
Type (CDROM, CDRW, DVD-ROM, DVD-W)	DVD-W	DVD-W	DVD-W
Interface (Internal, External)	Internal	Internal	Internal
USB ports			
# of ports	5	5	4
Type of ports (USB1.1, USB2)	4 - 1.10 and 1 - 2.00	4 - 1.10 and 1 - 2.00	3 - 1.10 and 1 - 2.00
Monitor			
CRT/LCD type (Plug & Play, TFT, Reflective)	Plug & Play	Plug & Play	Plug & Play
CRT/LCD refresh rate	75Hz	75Hz	75Hz

Figure 24: System configuration information for all three test systems.

Notes:
(1) These cache sizes are per core for each of the two cores.

Appendix B: Instructions for running the application scenarios

This appendix explains how we manually tested and timed each of the application scenarios. Though the vast majority of our discussions in this White Paper focus on the results of the automated tests, we verified that manually performing the same functions showed the same type of results as those the scripts produced.

As the instructions below reflect, to get the most consistent possible timings and to make our hand-timed actions more like the ones the automated scripts perform, we sometimes chose to follow procedures for launching applications that were different from those typical users would follow. (See Appendix C for additional information on scripting issues.) When we made such choices we also independently verified that the typical user procedures would still show similar results.

We consequently are confident that the benefits DC processor technology delivered in these (and other) scenarios are ones users would realize in real work situations and are not artifacts of the measurement or scripting technology.

We ran all application scenarios five times on each of the three systems under test, and we reported the median of those runs.

Scenario 1

This scenario uses two files or directories:

- Excel2Min.xls, a formula-rich 445KB Excel file.
- C:\Windows\Driver Cache\i386. To make the VirusScan task a manageable length and to ensure it always scanned the same files, we had it scan this Windows XP directory. The directory is 86.8 MB in size and contains nine files, including two compressed .cab files.

The scenario requires two applications:

- Microsoft Office Excel 2003 (Service Pack 1)
- Network Associates' McAfee VirusScan Enterprise version 8.0i

We used the following process in our manual tests of this scenario.

First, we set up the system to run the scenario by doing the following steps, which are necessary only once:

1. Start Excel.
2. Put Excel into windowed mode, if necessary, by clicking the Restore Down button in the upper right.
3. Set Excel's macro security level to low by doing the following. Go to Tools->Macros->Security, and open the Security Level tab. Click on Low.
4. Exit Excel.
5. Start McAfee VirusScan Console from Start Menu.
6. Select Task/New On-demand scan task.
7. Name the task "Cabfiles" (minus the quotes).
8. The VirusScan On-Demand Scan Properties – Cabfiles dialog appears. If it does not, right-click on the Cabfiles task name.
9. Click Remove twice to remove the two item names listed in the list box on that dialog, and click Yes on the confirm dialog that displays after each deletion.
10. Click Add, and scroll down to Drive or Folder.
11. Click Browse, and browse to C:\WINDOWS\Driver Cache\i386.
12. Click OK to close the Browse for Folder dialog, and again to close the Add Scan Item dialog.
13. Click the Detection tab, and check Scan inside archives (for example .ZIP).
14. Click OK.
15. Exit VirusScan Console.

16. Create a desktop shortcut to the Excel2min.xls file.
17. Create a desktop shortcut to the VirusScan Console.

To run each timed test, we then did the following:

1. Reboot the system.
2. Wait ten seconds after the Windows hourglass disappears and Windows XP has completed its startup sequence.
3. Open the Excel2min.xls document using its shortcut.
4. Open the VirusScan console using its shortcut.
5. Select the Cabfiles scan task.
6. Click Start.
7. Move the Scan Progress window out of the way of the Excel window and wait until the scan time is at seven seconds (the time displays in the status bar at bottom of the Scan Progress window).
8. Click on the Excel window, and press F9 to start the recalc.
9. Start the stopwatch when you press F9.
10. Stop the stopwatch when Excel shows that the recalc is done; it will do this by putting Ready in the status bar. Record this time as the foreground time.
11. The status bar in VirusScan gives you the elapsed time for the background task. Record this time as the background time.
12. Exit Excel.
13. Exit VirusScan Console.

<u>Scenario 2</u>

This scenario uses three files or directories:

- Content.ppt, a 33.4MB PowerPoint presentation with uncompressed pictures.
- DigitalHome1.Doc, a 3.7MB Word document with uncompressed pictures.
- C:\Windows\Driver Cache\. To make the AntiSpyware task a manageable length and to ensure it always scanned the same files, we scanned this directory.

The scenario requires four applications:

- Adobe Acrobat 7.0.3 Standard
- Microsoft AntiSpyware (Beta 1)
- Microsoft Office PowerPoint 2003 (Service Pack 1)
- Microsoft Office Word 2003 (Service Pack 1)

We used the following process in our manual tests of this scenario.

First, we set up the system to run the scenario by doing the following steps, which are necessary only once:

1. Start PowerPoint.
2. Put PowerPoint into windowed mode, if necessary, by clicking the Restore Down button in the upper right.
3. Exit PowerPoint.
4. Start Word.
5. Put Word into windowed mode, if necessary, by clicking the Restore Down button in the upper right.
6. Exit Word.
7. Start Microsoft AntiSpyware, and go through the setup assistant screens:
    a. On Step 1 of 3 screen: Select No to Would you like to enable AutoUpdater?
    b. On Step 2 of 3 screen: Select Yes to Would you like Real-time Security Agent protection?
    c. On Step 3 of 3 screen: Select No to Would you like to join the SpyNet community?

   d. On the Final Step screen: Uncheck Run a spyware scan every night, and click the Run Scan later link (located just under the Run Quick Scan Now button).

8. AntiSpyware opens after the install.
9. Click on the Spyware Scan target icon on the right side of screen.
10. Click the Manage schedule link at the bottom of the Spyware Scan screen.
11. Under Scheduled Scan Options, check Disable schedule.
12. Click Update Schedule.
13. Click OK on the Schedule Updated dialog.
14. Click on the Scan Options link below the Run Quick Scan Now button.
15. Under Select Your Scan Settings header, check Run a full system scan.
16. Click the Select… link next to the Scan selected drives/folders checkbox.
17. Select Clear all selected drives/folders, and check OK (this makes sure the program will scan only the files you select in the next few steps).
18. If a popup asks if you want to default to scan C:\drive, select No.
19. Check Scan select drives/folders, and click the Select… link again.
20. In the Select Drives or Folders to Scan dialog, browse to C:\Windows, and check Driver Cache as the folder to scan.
21. Click OK.
22. To the right of the Select… link, the following text should display: (C:\Windows\Driver Cache).
23. Make sure all four check boxes on the screen are checked.
24. Click Run Scan Now. (If you do not, AntiSpyware will not save all the setting selections you just made).
25. A message box, Your Scan Might be Incomplete!, appears. Select No in answer to its question about whether you would like to include the C:\ drive in your scan.
26. An AntiSpyware Spyware Scan Results screen will display. It should show 3839 files scanned.
27. Close the AntiSpyware Spyware Scan Results screen.
28. Exit AntiSpyware.
29. Start Adobe Acrobat.
30. Put Acrobat into windowed mode, if necessary, by clicking the Restore Down button in the upper right.
31. When the Registration screen displays, select I prefer not to register online.
32. Select Edit/Preferences. Highlight the Updates category, and check Do not automatically check for critical updates.
33. Select Help/Check for Updates.
34. On the Updates screen, highlight the Adobe Acrobat 7.0.3 and Reader 7.0.3 update and select Add.
35. Select Update.
36. The Adobe Acrobat Update Manager screen appears.
37. Click on Install Now.
38. Select Quit on the Adobe Acrobat Update Manager screen.
39. Exit Acrobat.
40. Create a desktop shortcut to content.ppt.
41. Open content.ppt.
42. Move it to the right side of screen.
43. Stretch it vertically until it displays 22 slides.
44. Create a desktop shortcut to Digitalhome1.doc.
45. Create a desktop shortcut to Microsoft AntiSpyware if there is not one already.

To time this test requires either two stopwatches or that you time only the foreground or background task on each run. The latter will require you to run the test twice for each complete set of timings you want. To run each timed test, we did the following:

1. Delete the Digitalhome.pdf file in the content folder after each run.
2. Reboot the system.
3. Wait ten seconds after the Windows hourglass disappears and Windows XP has completed its startup sequence.
4. Open Microsoft AntiSpyware using its desktop icon.

5. Select Scan Options.
6. In the Spyware Scan window, click Run Scan Now.
7. A warning pops up asking if you want to scan C:\. Do not click the No option yet; leave the window open.
8. Double-click on Digitalhome1.doc on the Desktop.
9. Click on the Convert to Adobe PDF button or menu item.
10. Click Save on the Save Adobe PDF File As dialog.
11. Start the Background stopwatch.
12. Wait three seconds.
13. In the AntiSpyware warning window you encountered earlier, click No.
14. Wait three seconds.
15. Double-click on the content.ppt file on the Desktop.
16. Start the Foreground stopwatch.
17. Wait until the thumbnails on the left display for the first 22 slides.
18. Stop the Foreground stopwatch.
19. The Acrobat task is done when it opens and displays the document.
20. Stop the Background stopwatch.
21. The scan progress window displays the time for the AntiSpyware task in minutes and seconds. Convert this time to seconds and record that result as the AntiSpyware background time.
22. Exit AntiSpyware, Acrobat, and Word.

<u>Scenario 3</u>

This scenario uses two files:

- CompetitiveSalesData.mdb, an 11.9MB Access database that contains sales data.
- Digital Homex2.doc, a 1.8MB Word document with pictures and graphics.

The scenario requires three applications:

- Microsoft Office Access 2003 (Service Pack 1)
- Microsoft Office Outlook 2003 (Service Pack 1)
- Skype Technologies Skype Version 1.3.0.60

We used the following process in our manual tests of this scenario.

First, we set up the system to run the scenario by doing the following steps, which are necessary only once:

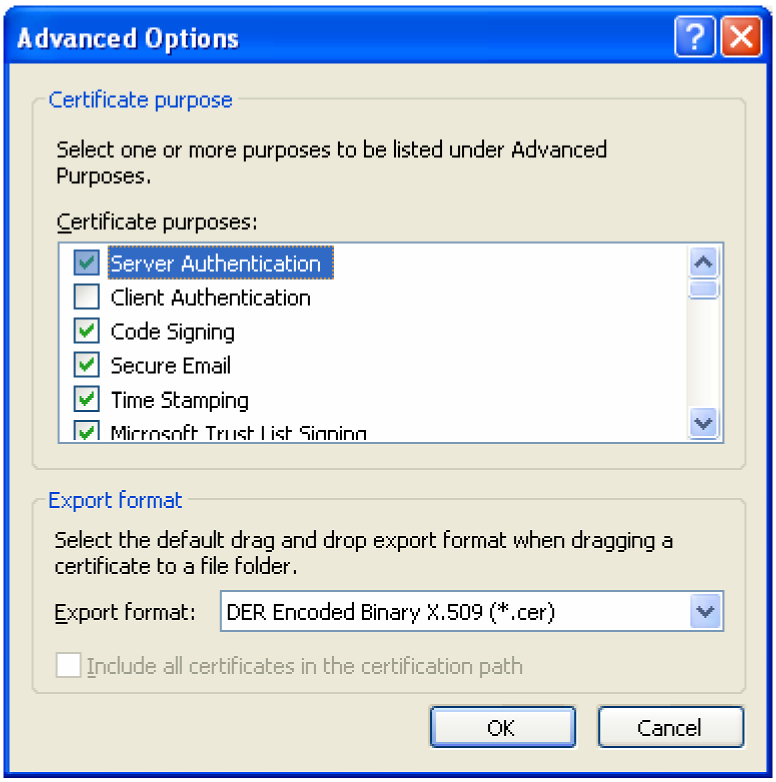
1. Start Skype.
2. Create a user account (one on each of four systems--the other two systems in this test plus a third 3.2GHz Pentium 4 640 processor system), and enter all the required information on the account screen.
3. Set up each client to automatically answer, and set the Away and Not Available icons to the highest option for each. To change these settings select, Options from the Tools drop down menu.
4. In the options window, the general options should be visible. Change the Show me as 'Away' when inactive for and Show me as 'Not Available' when inactive for settings each to 999 minutes.
5. Select Advanced in the left-hand window, then select Automatically answer call by placing a check in the box.
6. Select save, and close the options menu.
7. Add the Skype accounts of the other three systems as contacts on each system.
8. Exit Skype.
9. This scenario requires encryption as well, so you will need to get a digital certificate. To do this, you must connect your test machine to the Internet and you must have a valid e-mail address that you can use. Once you have both of those, do the following:
   A. Launch Internet Explorer and go to the Web page http://www.globalsign.net/digital_certificate/personalsign/index.cfm.

B. Select PersonalSignDemo -> GetYoursNowFree.

C. The site will now walk you through eight fairly simple steps you must take to get the certificate. Because the exact steps are beyond our control and may change, we did not document them here. Please note that at two points in the process the site will email, to the address you provide, a link for the next step. To continue the process at each of these points, simply click the link in the email message, which you must be viewing from the test machine.

D. At the end of these steps, you will have installed the certificate on the machine from which you are browsing, which is why you must take the steps from the test machine.

10. Start Internet Explorer.
11. Select Tools->Internet Options.
12. Select the Content tab.
13. Click Certificates.
14. Click on the certificate you just received
15. Select Advanced.
16. Make sure a check appears next to Secure Email, as it does in the screen shot below. If not, check the box next to it.



17. Start Outlook.
18. Create a fake POP3 email account in Outlook. The exact parameters do not matter, because Outlook should never actually try to connect to the server.
19. Set Tools->Customize->Always show full menus.
20. Select Tools->Options
    a. On the Preferences tab select the Email Options Button. On the Email Options dialog, select the Advanced Email Options Button. Make sure that “Suggest names while completing To, CC, BCC” is unchecked. Select OK to close the E-mail options windows.
    b. On Options->Mail setup tab, Uncheck Send immediately when connected.
    c. Click on the Send Receive Button, and set the Outlook send/receive settings to disable scheduled send/receive. Click Close to close the Send/Receive group’s window.
    d. On Options->Mail Format tab, select Mail format HTML, and check Use Microsoft Office 2003 to edit messages.

e. On the Options->Other tab, select Advanced Options, then check to see if the Startup in this folder: field shows the Inbox. Browse to the Inbox if a different folder appears. Select OK to close the Advanced Options.
f. On the Options->Security tab, select Encrypt contents and attachments for outgoing messages.
g. Click Setting… The address you used to order the Globalsign certificate should appear. Press OK to close this Change Security Settings.
h. On the Options->Other tab, select Advanced Options, then check to see if the Startup in this folder: field shows the Inbox. Browse to the Inbox if a different folder appears. Select OK to close the Advanced Options.
i. Press OK to close the Options dialog.

21. Create a shortcut to Outlook on the desktop.
22. Double-click on the Outlook shortcut.
23. Click on New to open the new message window.
24. Click on the attachment button, and attach the file Digital Homex2.doc.
25. Make sure the encryption symbol is selected on the right hand side of the menu bar.
26. Select File>Save.
27. The message should now be in the Draft folder in Outlook.
28. Do not enter any text into the message; just close the message.
29. Click on the draft folder so the message is now visible in the right-hand window pane. Drag the message to the Inbox folder.
30. Exit Outlook.
31. Open Access.
32. Put Access into windowed mode, if necessary, by clicking the Restore Down button in the upper right.
33. Go to Tools/Macros/Security, and select low for the security level.
34. Create a desktop shortcut to CompetitiveSalesData.mdb.

To time this test requires either two stopwatches or that you time only the foreground or background task on each run. The latter will require you to run the test twice for each complete set of timings you want. To run each timed test, we did the following:

1. Reboot the system.
2. Wait ten seconds after the Windows hourglass disappears and Windows XP has completed its startup sequence.
3. Open the Skype console window.
4. From the Skype window select Tools>Create conference…
5. In the conference call window select the contacts for the other three systems, and click Add.
6. Click on Start to begin the conference call.
7. Double-click on the CompetitiveSalesData.mdb shortcut.
8. Click on the macros menu option.
9. Select the export XML macro.
10. Click Run.
11. Start the Background stopwatch.
12. Wait 10 seconds.
13. Double-click on the Outlook desktop shortcut.
14. Start the Foreground stopwatch.
15. Pause the Foreground stopwatch when Outlook opens into the Inbox and displays the item count in the status bar at the bottom of the Outlook window.
16. Double-click on the first message in the Outlook inbox.
17. Resume the Foreground stopwatch.
18. Stop the Foreground stopwatch when Word displays the message.
19. Stop the Background stopwatch when the Access status bar says Export Complete.
20. Exit Access.
21. Exit Outlook.

Scenario 4

This scenario uses four files or sets of files:

- DigitalHomex1.XML (1.3MB) and DigitalHomex2.XML (11.2MB). These are two XML documents that contain pictures and graphics.
- ThreadPooling and SMC projects. Both are C++ source code samples from Microsoft's Web site.

The scenario requires three applications:

- Microsoft Visual C++ .NET 2003
- Skype Technologies Skype Version 1.3.0.60
- Microsoft Office Word 2003 (Service Pack 1)

We used the following process in our manual tests of this scenario.

First, we set up the system to run the scenario by doing the following steps, which are necessary only once:

1. Start Skype.
2. Create a user account (one on each of four systems--the other two systems in this test plus a third 3.2GHz Pentium 4 640 processor system), and enter all the required information on the account screen.
3. Set up each client to automatically answer, and set the Away and Not Available icons to the highest option for each. To change these settings select, Options from the Tools drop down menu.
4. In the options window, the general options should be visible. Change the Show me as 'Away' when inactive for and Show me as 'Not Available' when inactive for settings each to 999 minutes.
5. Select Advanced in the left-hand window, then select Automatically answer call by placing a check in the box.
6. Select save, and close the options menu.
7. Add the Skype accounts of the other three systems as contacts on each system.
8. Exit Skype.
9. Create a desktop shortcut for vc.sln (the Threadpooling project).
10. Create a desktop shortcut for smc.sln (the SMC project).
11. Create a desktop shortcut for Digital Homex1.xml.
12. Create a desktop shortcut for Digital Homex2.xml.
13. Put those shortcuts along the top right of screen.
14. Double-click on the vc.sln shortcut to open the console window.
15. Select tools>options to open the options window.
16. Select projects in the left-hand side of the options window.
17. Change Build Timing to Yes. Doing this will cause the build to show the elapsed time in the output window.
18. Double-click on the smc.sln shortcut, and repeat the step above to show the elapsed time in the output window.
19. Exit Visual Studio.

To run each timed test, we then did the following:

1. Reboot the system.
2. Wait ten seconds after the Windows hourglass disappears and Windows XP has completed its startup sequence.
3. Open the Skype console window.
4. From the Skype window select Tools>Create conference…
5. In the conference call window select the contacts for the other three systems, and click Add.
6. Click on Start to begin the conference call.
7. Double-click on the desktop shortcut for vc.sln.
8. Double-click on the desktop shortcut for smc.sln.

9. Move vc.sln to occupy roughly the left third of the screen, and move smc.sln so it occupies roughly the right third of the screen.
10. On each of these, select Build/Batch Build. A debug and release build should be visible, with just the release build checked.
11. Wait three seconds.
12. Double-click on the desktop shortcut for Digital Homex1.xml.
13. Start the Foreground stopwatch.
14. Pause the stopwatch when the file displays and the page counter at the bottom shows 1/42.
15. Select tools > compare and merge documents.
16. When the “Look in” window appears, click on the desktop icon, and select the Digital Homex2.xml shortcut.
17. Click Merge to begin the task.
18. Resume the Foreground stopwatch.
19. The Digital Homex2.xml document will open showing the status at the bottom of the window.
20. When the book symbol shows with a red X on it, the task has completed.
21. Stop the stopwatch.
22. The builds are done when the status bar at the bottom shows that the build succeeded. The elapsed time appears in the output window. You may need to scroll up a little to see the time results.
23. Exit Visual Studio and Word.

Appendix C: Issues in script development

To the best of our knowledge, despite its age IBM's Visual Test 6.5 remains the tool most widely used today for constructing application-based benchmarks and performance tests for PCs running various versions of Microsoft Windows. We have used this product (and previous versions of it) for many years to build performance tests. The tool does, however, have some stated limitations that unavoidably affect the way you develop performance tests with it.

First, the product's own documentation notes that its primary goal is to be a tool for automating application testing, not a benchmark development system. Consequently, the granularity of some of its functions and the way some of its functions behave are not ideal for benchmark development.

IBM also does not officially support Visual Test 6.5 for the Windows XP operating system. Because Windows XP is the leading and most current desktop version of Windows today, we nonetheless felt it was essential to use that operating system in our tests.

The presence of any scripting tool has the potential to affect the performance of a system. The tool unavoidably must, for example, occupy some memory and consume some processing power. Consequently, developing a performance-measurement script with such a tool involves maintaining a delicate balance between using the tool to automate typical real user behavior and minimizing the effects of the tool on system performance. To make sure the results of our scripts were accurate, we also, as we noted earlier, hand-timed each of the functions we scripted.

To minimize these limitations and problems we sometimes had to use scripting techniques that would achieve the same results as typical user behavior but not exactly mirror that behavior. Such techniques include inserting delays to mimic user think time and launching applications with a click on the OK button of a pre-filled Run command line. The hand timing instructions we provide in Appendix B reflect those techniques, so following those instructions will yield results similar to those the scripts produce. Whenever we had to use one of these alternative techniques we manually verified that doing so did not materially alter the way the system behaved and that real users performing the same actions in more typical ways would see the type of benefits from the DC processor technology that we describe.

The timings the scripts produce also inevitably contain some variability. This variability is a result of the combination of the tool's limitations and the generally asynchronous nature of the many processes Windows XP and other modern operating systems have running at any given time.

Finally, though one of the goals of this effort was to produce reliable scripts, we were not trying to build bulletproof benchmarks for wide distribution and use. We developed the scripts to mimic user behavior on our specific test systems; on different systems the scripts might show different levels of benefit from a DC processor or even fail to work. So, though the scripts are as reliable, self-contained, and free of system dependencies as we could reasonably achieve within the project's timeframe, they do sometimes fail or encounter problems. Should a problem occur, rebooting the system and running the script again will generally yield a good result.

Principled Technologies®

Principled Technologies, Inc.
4813 Emperor Blvd., Suite 100
Durham, NC 27703
www.principledtechnologies.com
info@principledtechnologies.com