Checking Reality Again:
The Case for Full Disclosure, Mr. Lincoln

By Bruce Weiner

October 6, 1998
Updated: October 7, 1998
(PDF version 84 KB)

What's Going On

Mindcraft tested the file-server performance of Microsoft Windows NT Server 4.0 and Novell NetWare 5 over the native TCP/IP stack for each operating system. We tested with the Ziff-Davis Benchmark Operation NetBench 5.01 benchmark and published the report at our Web site. Novell published what they call a "Reality Check" in response to our report. In it, Novell impugns our honesty, professionalism, technical competence, and independence as well as accuses us of trying to fool you. We cannot let Novell's unreal and incorrect characterization of Mindcraft and our report go unchecked. In addition, we have received many e-mails about our report; some asking clarifying questions and many emotional diatribes disparaging Mindcraft, our employees, and our work. This rebuttal will address those e-mails also.

To begin with, the Novell heading "You can Fool Some of the People Some of the Time…" is a misquotation from President Abraham Lincoln. He actually said, "If you once forfeit the confidence of your fellow citizens, you can never regain their respect and esteem. You may fool all of the people some of the time; you can even fool some of the people all the time; but you can't fool all of the people all of the time." We will not forfeit the confidence, respect, and esteem our clients and others have in Mindcraft and our reports as Novell would have us do. Therefore, we ask you to read this rebuttal of Novell's "reality check" with an open mind and to decide for yourself what the truth is.

At Mindcraft, we pride ourselves on not fooling any of the people any of the time. We call things the way we measure them and see them. Can we make mistakes? Sure. And when we make one, we admit it. In this case, we believe that we did not make a mistake. We'll start by addressing the issues Novell raised.

Dubious Origin

Novell did ask Mindcraft who sponsored the testing for our report. We declined to answer that question at the time because our client had not authorize us to disclose their name. We have now been authorized to disclose that Mindcraft was commissioned by Microsoft Corporation to produce an independent and unbiased assessment of the file server performance of Windows NT Server 4.0 and NetWare 5 using NetBench 5.01. Our clients tell us it is essential that we maintain their confidential information, including their identity. So you can see that we were not previously in a position to release the name of the client who sponsored our work.

We have worked with many hardware and software vendors and user organizations over the more than 13 years we have been in business. The reason our clients keep returning to have us do more work for them is that the work we do is fair, accurate, and technically competent. Contrary to Novell's implication, who our client is makes no difference as to how we run a test. We are an independent test lab whose clients pay us to obtain fair and accurate information about products whether they are their own, a competitor's, or ones our client wants to buy.

I'm sure that's why Novell hired us to produce a performance report on BorderManager. I'd be willing to bet that they think that our BorderManager report is fair, accurate and unbiased. This dubious origin claim is a little too disingenuous. It's a case of casting doubt and aspersions if you don't have any thing better to address the truth.

Unprofessional Methods

(1) Standard Benchmarking Practices Were Violated. Novell claimed that "For a NetBench benchmark to be considered publishable in a Ziff-Davis publication, there are two requirements that are absent in the Mindcraft results: (1) standard test configurations must be used, and (2) three test runs must be averaged to get the results." You should note that they are referring to Ziff-Davis's publication procedures, not any NetBench run rules or license requirements. While Novell is trying to find fault with Mindcraft's procedures when they claim we didn't use standard test configurations, we want you to know that had we used the standard NetBench NBDM_60.TST test suite, we never would have seen the peak performance of NetWare 5 on the server we tested! As you can see by looking at Mindcraft's report, the peak performance for NetWare 5 occurs when there are 96 clients, that's over 50% more clients than the NBDM_60.TST test suite uses. You should also know that there are no standard NetBench configurations for servers, client systems, or networks. The configuration we used is published in our report in full detail as required by the NetBench license. It's much more detail than Novell published in their response to Microsoft, which violates the NetBench license because it is missing configuration details the license requires. It's even more detail than Novell published in the NetBench results at the end of their "reality check."

In the response to (2) below, we'll address Novell's criticism that using iterations instead of time to control the test duration produces results that are skewed to favor one product over the other.

As far as averaging three runs to get published data, we found that when we did back-to-back runs we had less than a 5% difference between runs with the same configuration parameters. This may be because we are careful to recreate the same initial conditions when we start a NetBench test. We begin by reformatting the server's data disk and rebooting the server. This way we avoid disk fragmentation that could degrade performance as we do multiple benchmark runs. Then we reboot the client systems and the test controller to be sure that they don't have any cached data. Then we do a fresh install of NetBench on the server. After all of this, we run the benchmark. We don't know if Novell follows similar procedures because they did not disclose them.

By the way, any performance differences between benchmark runs that are 5% or less should be considered a run variation and are insignificant. This applies not only to NetBench but also to all standardized benchmarks where disk latency, network traffic, and similar timing dependencies affect the results.

(2) The benchmark test parameters used were known to give faulty results. Novell's accusation that "Mindcraft's use of this parameter [iterations instead of time to determine how long to test] implies either a lack of professional testing procedures or a deliberate attempt to influence the results" couldn't be more incorrect. Mindcraft had several discussions with two Novell representatives about the best way to configure NetWare 5 to maximize its performance as measured by NetBench. Both were very friendly, helpful, and quite competent. It turned out that they are the same Novell technical experts who supported PC Magazine for their First Look article about NetWare 5, so we assume they are highly competent to tune NetWare 5 to get the best NetBench performance results. We used all of the NetWare 5 server and client tuning parameters they recommended, except one which we'll discuss in (4) below, and did not change any other tuning parameters from the NetWare 5 defaults for the server or clients.

The Novell experts did raise the question about using iterations instead of time to control the test duration and did mention that Microsoft had reported a bug with iterations. However, they did not know what the nature of the bug was. So we contacted Microsoft to get the details. Microsoft told us the problem is that NetBench incorrectly adds the measurements made during ramp-up and ramp-down to those made during the period when it should be reporting measurements.

In order to understand this issue better, you need to understand more about NetBench. When a NetBench test duration is specified in iterations, each NetBench client runs through its entire test script for the specified number of iterations and then stops. Since each client in a given mix uses the same test script, they will report the same number of bytes transferred. However, they will take a different amount of time to complete the specified number of iterations yielding a different throughput rate for each client.

When NetBench test duration is timed, each mix ends approximately after the number of seconds specified. The NetBench client detail report for timed duration tests shows for each client the bytes transferred, the mix duration in seconds, and the number of iterations through the test script (expressed only as an integer). Again, throughput is computed for each client as the number of bytes transferred divided by the duration in seconds. In timed mode, NetBench may terminate a client when it is in the middle of an iteration of its test script. So each client will report a different number of bytes transferred and a different number of iterations elapsed. In addition, each client will run for a different amount of time. We have seen differences of almost three seconds in the duration and two elapsed iterations between clients in a given mix. While these numbers may seem small, they can add up to over a 30% difference in the amount of work clients do in the same mix as measured by the bytes transferred. This means for timed duration tests the total amount of work done will vary for different servers. In other words, each server will do a different number of file open, close, read, write, etc. operations.

The simplest way to think of the difference between time duration and iteration duration is to think of a race between two runners. Using time duration is like a race that sees how far the two runners can go in a given amount of time. One may cover 100 yards while the other covers only 90 yards. On the other hand, using iteration duration is like a race where each runner must cover 100 yards.

So when should you use a test duration based on time or iterations? You should use a timed duration when you don't care whether or not each client does the same amount of work. This also implies that you should use a timed duration when you don't care whether or not the server being tested does the same amount of work as another one to which you are comparing it. On the other hand, you should use an iteration duration when you want each server you are comparing to do the same amount of work. Thus, we contend that when you want to compare servers using NetBench the only fair way to do it is using iterations for duration because you are guaranteed that the throughput produced is based on executing the same number of each type of file operation on each server.

Now, getting back to how to deal with the iteration reporting bug. Microsoft told us to divide the bytes transferred that NetBench reports by three, since there were three total iterations and only the middle one was supposed to be measured. But how could we validate that Microsoft was telling us the truth about the bug? That turned out to be simple. Here are the steps we took:

1. We divide the number of bytes transferred that NetBench reports for a benchmark based on iterations by three (again, the three only applies to the case where there is one iteration each of ramp-up, test, ramp-down; if there were one iteration each for ramp-up and ramp-down and two iterations for test, we would divide by four because four iterations were measured). This gives us what should be an invariant as to the number of bytes transferred per iteration which we'll call *xfer*.
2. Next, we look at any NetBench report produced using a time duration. We take the bytes transferred reported for each client and divide it by *xfer* which should yield the number of iteration reported for an time-based test if Microsoft is right.
3. In every case, we found that the number of iterations computed in step 2 matched the number of iterations NetBench reports for every client in every mix we looked at for tests that used time-based duration. We verified this iteration calculation for several NetBench tests we did on systems with Windows NT Server 4.0 and Solaris 2.6.

So we have given you a way to prove that iteration-based NetBench testing can produce accurate results.

One might question whether one ramp-up iteration is enough to properly warm-up the server's cache under the test scenario we used. A more detailed look at how NetBench works will shed some light on this. Each client has its own set of files that it uses. Each client creates these files the first time it is included in a mix. The client reuses the files when it runs in subsequent mixes. So if a client is used in five mixes then the files will have been used four times by the time the last mix starts. As more mixes and thus more clients are added during the progress of the benchmark, the percentage of files that have only been through one iteration goes down. For example, in the second mix 15 new clients are added so their workspace has been accessed only for the one ramp-up iteration before measurements begin while the workspace for the first client has been accessed for a total of four iterations (the three times during the execution of the first mix and one iteration during the ramp-up of the second iteration). This means that for the second mix, 15/16ths, or 93.8%, of the workspaces have been accessed only once. By the seventh mix, which is where we measured the peak performance for NetWare 5 and the two-processor configuration for Windows NT Server 4.0, the number of workspaces that are accessed only

once before measurement starts represent 16/96ths, or 16.7%, of the workspaces. To put it another way, 83.3% of the workspaces had been used for at least four iterations, certainly enough iterations that the server's cache should be considered warmed up.

As for the NetBench documentation reference that Novell cites to justify their dismissal of iteration-based tests, careful scrutiny will yield a very different impression than Novell wants you to have. The manual that comes with NetBench, *Understanding and Using NetBench 5.01*, on page 154 states:

> "By executing a test mix in this way [using a ramp-up and ramp-down period], NetBench is better able to measure your server only when it is in a steady state... The server is no longer thrashing as a result of starting the tests, and it has flushed out any residual caching effects that occurred as a result of creating test data files. In addition, all clients are executing the tests, so the server does not have a light load. (A light load occurs when only a few clients are running tests, such as during the Ramp up or Ramp down periods when not all clients have started the tests yet or some of the clients have ended the tests.)"

In other words, the documentation claims that during the period when NetBench is recording information from the test all of the clients should be running. On page 155, however, the NetBench documentation contradicts the above steady state claim in a highly illogical way when it states:

> "We recommend you use seconds instead of iterations because you can maintain more control during the tests. If you use iterations, you can run into a situation where the load on the server is not balanced. For example, you might have 20 clients, each of which is to perform 20 iterations, but the server can only handle 10 clients at once. You may find that 10 clients do all their iterations and stop. Then the next 10 do all their iterations. In other words, one group of clients ran after the other group of clients. As a result, you only had 10 clients running on the server at one time instead of having all the clients using resources at the same time..."

What does the NetBench documentation mean when it states "the server can only handle 10 clients at once?" If it is a reference to a license limitation on the server that restricts how many clients can be supported simultaneously, then the same situation would apply to tests using a time duration. The license limitation certainly does not apply to the tests we ran since the servers were licensed to support the number of clients we used. If this statement is referring to some inherent limitation of the server operating system, we doubt that any NetWare 5 or Windows NT Server 4.0 user or system administrator would stand for the operating system acting in such a sequential mode. If this statement is referring to some inherent limitation of the server hardware, we did not observe this sequential behavior on the 400 MHz Pentium II-based ProLiant 1850R we used when running NetBench.

Let's look at the statement in the documentation more carefully. The concept that 10 clients will complete all of their iterations before any of the remaining 10 clients get to perform any of their file operations implies one of the following:

- Some wait synchronization between the clients;
- The server is telling the clients which ones it will listen to; or
- The server does not queue file system requests in a fair manner.

There is no NetBench documentation describing any of the above nor is there a configuration parameter for any of the above behaviors. If you believe the above scenario that one group of clients can complete their execution before another group starts, why wouldn't the same scenario apply to the same test when the duration is specified in terms of time instead of iterations? There is nothing about NetBench that would prevent it. In other words, the above rationale for not using iterations is wrong.

Could there be another rationale for not using iterations? Sure. When we asked a Ziff-Davis NetBench developer, he replied that Microsoft had reported the bug we discussed above. He also questioned how many iterations would be enough to warm up today's fast file servers. But he did not say that iterations do not work when we asked him if there were any other technical problems besides the iteration reporting bug.

The Novell technical experts did tell us that they saw lower performance for NetWare 5 when they used iterations for test duration instead of time. They also told us that they saw improved performance for NetWare 5 when they used time duration instead of iterations. To report the other side of this issue, Microsoft told us that they see at most a 2% to 3% difference between using iterations and time when testing Windows NT Server 4.0.

What can account for the differences that Novell saw? To begin with, Novell wasn't specific as to how much performance difference they saw, just as they weren't in their Reality Check Web page. But we'll take a stab at a rationale for why there could be a significant difference between tests run with iteration and time durations. The most obvious reason is a client effect. As we pointed out above, with time duration some clients may do over 30% less work than others. Also, Novell seems to like using opportunistic locking when it publishes NetBench results. Because opportunistic locking means that each client must also fill its own cache in addition to that on the server, it may take more ramp-up iterations for them to do it. So it may not be that using iterations deflated the NetWare 5 results, but rather, the amount of work done for the iteration-based test was different than that done for the time-based test.

It would clearly help to do a detailed study on what happens when using iteration- and time-based duration with NetBench.

As for Novell's specious claim that Mindcraft used unprofessional test procedures or made a deliberate attempt to influence the results, you can see from the detail and information we provided above that we approached the testing most professionally and used the test method that lets you compare the performance of NetWare 5 and Windows NT Server 4.0 fairly because each server did the same number of each type of file operation.

(3) The server hardware configuration was contrived to put NT in it's best light. The Novell technical experts we talked to clearly misunderstood our discussion about tuning the NetFlex 3 Network Interface Card. We were discussing Web server performance we have seen on Compaq systems when using the NetFlex 3. If you look at published SPECweb96 reports, you will see that Compaq systems are routinely configured with a MaxReceives parameter. Compaq also does this for their NetBench tests. It is a normal, recommended tune to apply to a system that has heavy network traffic. In fact, it is similar to the tuning values Novell recommend we set for the minimum and maximum packet receive buffers. Novell claimed that NetWare 5 would run below its best performance level if we didn't set these packet receive buffer parameters. So what's wrong with the NetFlex 3 tuning? Nothing! Remember, we used the same NetFlex 3 and similar tuning for both NetWare 5 and Windows NT Server 4.0.

Novell's reference to "blue screens" (system crashes on Windows NT Server) is also a misdirection. We told them what would happen with versions of Web servers that are no longer shipping if we didn't tune the NetFlex 3 card. Despite Novell's implication, we have **never** seen a "blue screen" on Windows NT Server 4.0 when using NetBench.

Novell's contention that we fixed the disk configuration to favor Windows NT Server 4.0 over NetWare 5 is totally without merit. Here are the facts about the disk subsystem we used for our tests:

- NetBench performance is highly sensitive to the performance of the disk subsystem on the server. So we configured the disk subsystem to be the fastest possible for both operating systems.
- We used only one disk for the operating system, whether it was NetWare 5 or Windows NT Server 4.0. The OS disk was connected to the RAID controller in the server because that is the highest performance configuration. No I/O bandwidth preference was given to the OS disk as Novell claimed and, as you'll see in the last bullet below, that would be irrelevant anyway.
- We used nine disks to hold the data on one logical drive configured as RAID 0. By spreading the data across nine disks we could achieve the best performance possible given that we had only that many disk bays available in which to put drives.
- Novell's claim that Windows NT Server 4.0 accessed the system disk during the test is wrong. We did several runs using the standard Windows NT performance monitor tool, perfmon, to see how the various parts of the system performed. We did look at physical disk accesses during the test and found that there were no accesses to the Windows NT Server 4.0 system disk.

Novell claimed that "This [disk] configuration, while possible in a test environment, does not represent a typical business use." Yet, there is nothing atypical about the disk subsystem configuration we used. We used the same one for both the NetWare 5 and the Windows NT Server 4.0 tests. Where is the bias Novell is looking for? Like beauty, in the eyes of the beholder.

(4) Major operational characteristics were applied unevenly. Novell's discussion of opportunistic locking is a very clever act of misdirection. Windows NT Server 4.0 does ship with support for opportunistic locking turned on by default. And so does NetWare 5 according to the Novell technical experts that provided us tuning parameters. But opportunistic locking also requires support from the clients. Windows 95, Windows 98, and Windows NT Workstation all ship with client-side caching (opportunistic locking) turned on. And Microsoft recommends that users keep it that way. In fact, they make it difficult to turn opportunistic locking off on Windows NT Server 4.0 and their client operating systems.

Novell, on the other hand, currently recommends against turning opportunistic locking on in their client even though they support it by default in their server. In fact, Novell makes it hard to turn on opportunistic locking in their client. Unlike other tuning parameters set via the Advanced Settings option of the NetWare 5 client, the Novell technical experts told us we needed to insert a new, undocumented key into the registry on each client system to turn on opportunistic locking.

So what's all this mean? We thought about how each system typically would be used in a business (remember Novell's complaint at the end of point (3) above) and concluded that most users would go with each vendor's recommendation. So that's how we tested each system, we used the default settings for opportunistic locking.

Our approach for dealing with opportunistic locking was contrary to what Novell and Microsoft wanted us to do for the benchmark. Novell wanted us to go against their own recommended configuration and do something they don't want their customers to do. Microsoft wanted us to turn on a feature they call TurboMode which Microsoft says does additional caching on the clients safely and improves NetBench results. They both wanted us to use special settings on the clients that are not enabled by default and that would require setting registry keys on every client system connected to the server. We thought that would mislead our readers, so we didn't do it.

Novell claimed that "...Mindcraft reused the same files thus maximizing the positive impact of client-side caching..." This shows a lack of understanding of how NetBench works. We have no control over how NetBench uses files as Novell implies. In fact, NetBench does reuse the same workspace from mix to mix unless you explicitly configure it not to. But that's consistent with the NetBench documentation and is also how Novell runs NetBench. More smoke in your eyes!

Obviously Biased Results

The points below continue with the same numbering Novell used in their "reality check."

(1) The benchmark results are contrary to other published results. We have seen many of the published results to which Novell refers that show NetWare outperforming Windows NT Server. If the referenced reports are true, why then does the First Look article show that Windows NT Server 4.0 outperforms NetWare 4.11 when tested with NetBench?

(2) The Price/Performance metric cited is extremely misleading.

Let's look at Novell's complaints bullet-by-bullet:

- A price/performance metric is affected by both of its components. It seems disingenuous for Novell to claim we used contract pricing for Windows NT Server 4.0 and list pricing for NetWare 5 when we fully disclosed that we used pricing for both operating systems based on a quotation from a VAR whom we asked to give us prices for the same discount level and licensing circumstances.
- Of course, Novell will think that the price/performance metric we computed is wrong because they disagree with the performance we measured. As we showed above our measurements were unbiased.
- The last three bullet items under this heading in Novell's "reality check" address Total Cost of Ownership, something we did not address in our report. Therefore, we have no comment on their statements.

Comments on Novell's Benchmark Tests

Novell attached their own benchmark results to their "reality check" without discussing what they show or providing enough detail to reproduce them or to evaluate how fair they are. For example, what NetBench testing parameters did they use? What kind of networking was used? There are several version of drivers for the Intel NICs in the server, which one did they use? What were the client systems? What was their configuration? Were switches or hubs used for the networking? Which models were used? Was the networking full- or half-duplex? The list can go on.

The NetBench performance results Novell published for a single- and dual-processor system with opportunistic locking show very little performance difference between the two configurations. Why? Some explanation would help you and us understand why NetWare 5 shows very little scaling, contrary to Novell's scalability claim.

Summary

This rebuttal shows in detail that the attacks and criticisms that Novell made in its "reality check" are unfounded. We wish that Novell had dealt directly with us before putting out such an embarrassingly biased and inaccurate response to our file sever comparison. Given the professional manner that Novell's technical experts dealt with Mindcraft when we were performing the tests, it's a shame that someone in their organization had to resort to name calling and libelous remarks as they attacked our report with unsupported generalities.

Mindcraft also would welcome the opportunity to work with Novell to resolve any of the issues raised by our testing or theirs and to produce another honest and impartial product comparison.

For more information, contact us at: info@mindcraft.com
Phone: +1 (408) 364-2860
Fax: +1 (408) 364-2862