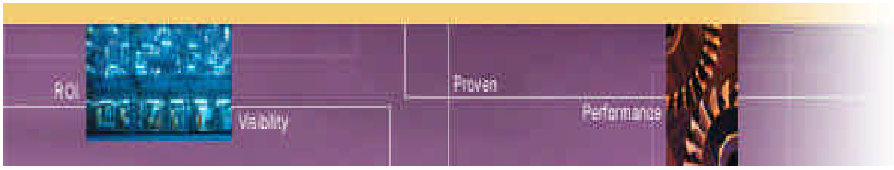


OLAP Benchmarking in Supply Chain Management with Solid State Devices

by Paul Bertucci

April 15, 2002

Table of Contents

The Value Chain Realm

i2 Technologies carves out the overall Value Chain software business into three distinct areas:

- Supplier Relationship Management (SRM)
- Supply Chain Management (SCM)
- Customer Relationship Management (CRM)

In order to guarantee successful implementations to their customers in each of these realms, i2 Technologies must not only provide the software leadership, but also be able to deploy their products on hardware platforms that meet the performance and scalability requirements of their customers.

It is for this reason that we will target one of the primary areas within i2 Technologies offerings for an advanced benchmark in order to answer many scalability and performance questions that have not been addressed in the industry today. It should be noted that many of the same techniques discussed in this paper can be applied to other SCM, CRM or SRM products with the same results and to many other product offerings other than one's from i2 Technologies. For i2 Technologies, the flagship SCM product that is at the heart of many of their sales is the Demand Management product suite (Demand Planner). As you have no doubt heard by now, moving to a demand driven SCM model is the goal of most organizations. In addition, most organizations that purchase SCM software nearly always implement the Demand Management products first because of the immediate ROI potential when done correctly.

The current Demand Management product (Demand Planner) is built on a proprietary OLAP multi-dimensional database cube. This OLAP environment offers huge benefits to the users since most demand planning requirements fit so naturally into a multi-dimensional model. In addition, OLAP users require extensive analytic methods and techniques when developing Demand Forecasting models that have drill-down capabilities within dimensional hierarchies (Product dimensional hierarchies, Geography/Customer dimensional hierarchies and Time dimensional hierarchies – as seen in figure 1).

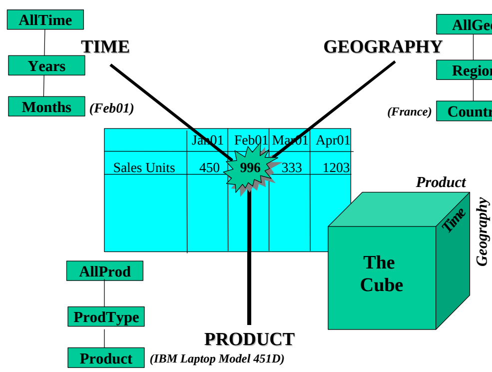


Figure 1
Multi-dimensional OLAP database cube for Demand Planning

Hardware Upgrade Transparency Approach

Many software implementations can benefit from some type of performance improvement that provides a path to higher number of users and system scalability. For some *software* architectures it is not always possible to increase performance by just turning certain handles. This benchmark will take the approach that the customer can build the SCM (Demand Planner) implementation in the typical way without having to change internal system configurations. We will then test to see if performance can be improved by a couple of different hardware-upgrade approaches that will be predominantly transparent to this software implementation. This hardware transparency approach greatly simplifies the whole picture for the vendor (i2) and the customers and can also be done extremely quickly.

The first hardware upgrade will be to squeeze performance and scalability out of the existing server (1-way CPU) by using a Solid State device for all "hot" files in the OLAP cube. Hot files are those files that receive heavy I/O activity in the course of typical processing.

The second hardware upgrade will be a straight server upgrade from our baseline server (1-way CPU) to a much more powerful (5-way CPU) server without any Solid State Devices.

The third hardware upgrade will be to achieve the maximum performance and scalability possible from the upgraded (5-way CPU) server by using a Solid State device for all of the "hot" files in the OLAP cube.

Benchmark Goal

A basic goal of achieving certain minimum system performance should be defined early on in the design cycle. Achieving these goals must be met as a measure of success for the entire implementation. The benchmark example being used here is that of a typical i2 Technologies Demand Planning implementation for a Fortune 500 manufacturing company. For every hour of processing time of this currently executing system (which we will call our Baseline) we will want to bring that down to near 20 minutes (at least a 67% (3x) improvement ~ or be faster by at least 3 times) in order to fulfill our performance and scalability goal <u>and</u> with minimum impact to the software implementation.   An example would be a data load into the cube that takes 18 minutes must come in at around 6 minutes for it to be worth doing at all. Anything less is not acceptable.

Benchmark Findings Summary

As you will see from the details that will follow, the overall performance improvements ranged from a low of 29. 75% (1.3 times faster than before) for certain OLAP DB maintenance operations to improvements as high as 75.22% (4 times faster than before) for massive OLAP Data Measure update operations. Overall, the areas that are most visible to the end-user experienced the best improvements (Data Measure updates, Data Loads and Data Exports). This will no doubt make certain upgrade paths much more feasible because of the transparency of the upgrade and the direct affect this has on the end-users of the software (making them much more productive and providing them with much more enabling data so they can do their jobs better and more accurately).

Hardware/Software Environment in Benchmark

BASELINE SYSTEM

Hardware: HP NetServer LC2000 (1-Way/Intel Pentium III – 600 MHz)
Raid Disk Array (Symbios Logic 896, 22910 PCI SCSI Adapter)
HP 9.10GB A 80-S1A0 SCSI disk devices
Memory - 512 MB RAM - SDRAM

Software: Windows 2000 Advanced Server ver 5.0.2195
i2 Technologies Demand Planner (current release)
Microsoft SQL Server 2000 (legacy data system)

SERVER UPGRADE SYSTEM

Hardware: HP NetServer LH6000 (5-Way/Intel Pentium III – 700 MHz)
Raid Disk Array (Symbios Logic 896,22910 PCI SCSI Adapter)
HP 18.2GB 10K Ultra3 SCSI disk devices
Memory – 3 x 256 MB RAM – ECC SDRAM

Software: Windows 2000 Advanced Server ver 5.0.2195
i2 Technologies Demand Planner (current release)
Microsoft SQL Server 2000

SOLID STATE DEVICE UPGRADE

Hardware: Solid Data Excellerator 1000 - 5 GB – ULTRA HVD SCSI SSD unit
Adaptec 2944UW Controller

Benchmark Description

The specific area's that will be benchmarked are centered on the typical processing done in this SCM product environment. These are:

1. Data loading/data integration processing – daily processing
2. Data Accesses/OLAP cube processing – real-time/dynamic accesses
3. And certain Data Maintenance processing - daily or weekend processing

1. Data Loading/Data Integration processing

Typically, data loading/data integration processing can take multiple hours each day. This puts significant pressure on the already tight batch data processing window of most company's. In addition, when failures occur, the likelihood of fixing issues and re-running these jobs are, remote at best (due to this limited batch processing window). This usually results in the Demand Planning cube not getting its needed incremental updates which, in turn, has a direct affect on the usability and accuracy of the data cube itself. In addition, this batch processing window has typically dictated that only certain amounts (data volumes) could be done within the limitations of that window. This, in turn (from a design point of view) has limited what was available in the data cube itself (you can only put into a cube what can be processed in the available batch processing window). If this could somehow became a non-factor through increasing the amount of data that can get loaded (easily) into the cube within your batch window, then you would have the possibility of better (more accurate) demand forecasts being created, and so on. So, this translates into two areas of significant ROI; one in the data quality/data volume area (more data/better forecasts) and the other in the data latency/data timeliness area (most current data available in the cube). It will be here that a direct ROI can be calculated.

The figure 2 describes the typical data loading/data integration configuration:

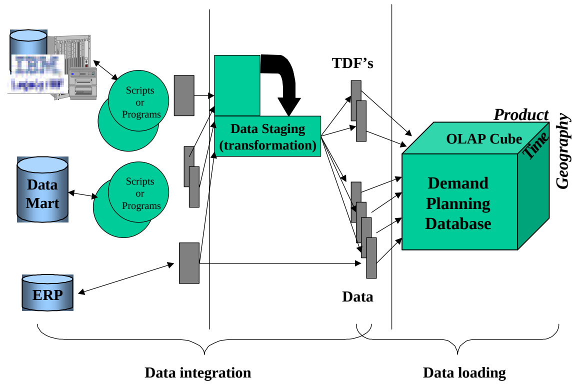


Figure 2
Data integration/Data loading implementation

For **data integration**, many parts of the data staging area and the intermediate flat files would be candidates for performance improvements. Any gains here would have a direct impact on reducing the batch processing time requirements as indicated earlier.

Our benchmark legacy system consists of the following:

- Inventory table – 3,100 rows
- Product structure table – 1,330 rows
- Sales Orders table – 3,650,000 rows
- Sales Order Details table – 12,767,867 rows
- Customer table – 1,910 rows
- Market/Geography table – 958 rows
- Average Selling Price table – 5,660 rows

Staging Data (used during transformation/data integration) will create numerous tables and flat file representations during conversion, aggregation and initialization of OLAP cube loading. It will be necessary to create concatenated multi-dimensional keys (Time + Geography + Product keys) for every piece of data (data measures data) that will be loaded into the cube.

For **data loading**, any improvements that surround accessing the data load files (flat files) and the Table Description Files (TDFs) and pumping them into the OLAP cube faster will directly impact the ability to re-run processing as well as do much more data volume processing. This would significantly improve the value of the resulting OLAP cube itself.

2. Data Accesses/OLAP cube processing

On the front-end tool and OLAP cube processing side, i2 Demand Planner currently supports a multi-user environment with typical lock management. When updates occur in the OLAP cube, i2 has chosen to lock significant portions of the hierarchy for the duration of the update. Sometimes these updates take several minutes to hours depending on how high up in the hierarchies the updates occur. And, in general, the bigger the cube and the more end-users connected simultaneously the slower the overall process is.

Improving the **Data Accesses** can greatly affect the overall usage and productivity of the end-users. For this reason, a huge amount of weight will be placed on the benchmark results from this area.

If the overall update processing can be greatly reduced, many more possible "what if" data forecasts, allocations, or value updates can be attempted (more because it is so fast!). All have direct results in more end-user usage of the tool, greater satisfaction with the tool, and better resulting demand forecasts. This is also directly translatable into ROI (measured via forecast accuracy and forecast cycle time).

Below are the data measures that were defined in the OLAP cube and used during this benchmark:

- Sales Units data measure
- Average Selling Price data measure
- Inventory data measure
- Sales Forecast Demand Signal data measure

The typical database update operations that were used during this benchmark were DBCalc's, Middle-out Forecasts, Allocations, and standard updates in a variety of different ways:

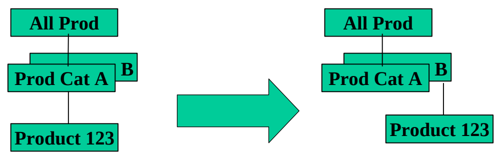
1. Single data measure Updates - lowest level in hierarchy
2. Single data measure Updates – highest level in hierarchy
3. Multiple data measure Updates – lowest level in hierarchy
4. Multiple data measure Updates – Highest level in hierarchy

These were then summarized into an "Updates" value for benchmark reporting.

3. Data Maintenance processing

There are a couple specific data maintenance processes that are extremely time-consuming and cost companies much money and time. They are doing hierarchical data movement (e.g. when a product is moved to a different product group in the cube). This is very time-consuming because essentially the product's location and all related data must be physically moved in the cube. Typical run times for a single hierarchical movement (called **realignments**) range between 1 to 2 hours each. Companies very often move dozens or hundreds of products around in the cube due to categorization errors or product release strategies. An ample ROI can be gained if this time intensive processing can be sped up resulting in a more up to date and accurate cube at all times, not to mention the increased usage of realignment (that has been used sparingly due to it's processing time limitations).

Figure 3 shows an example of a physical "realignment" of a product in an OLAP cube:



Product 123 within Prod Category A
realigned to be in Prod Category B

Figure 3
Data Realignment in an OLAP cube

In addition, checking the OLAP cube's **synchronization** (checksync) at all levels and for all data measures is extremely time consuming but very valuable from a data integrity point of view. Any improvements in these areas will be very valuable as well. The exact set of DB maintenance operations performed in the benchmark was:

1. Realignments (middle level)
2. Realignments (lowest level)
3. Reindexing (all)
4. CheckSync (1 dbf in Cube – all data measures in dbf)
5. CheckSync (all dbf's in cube – all data measures in dbf's)

Benchmark Findings

The OLAP Cube was populated to be about 5 GB in size (in total) and it was found that all operations execution times increased at the same rate as the physical database size increased. In other words, the same operation execution times took roughly 8 times longer on an OLAP cube that was 40 GB in size.

All benchmark measurements are being reported in average time (elapsed time of an operation).

During each execution, Performance Monitor was also turned on and the following **counters** were activated:

- Memory\Page Faults/sec
- Memory\Page Reads/sec
- Memory\Page Writes/sec
- PhysicalDisk(_Total)\% Disk Read Time
- PhysicalDisk(_Total)\% Disk Time
- PhysicalDisk(_Total)\% Disk Write Time
- PhysicalDisk(_Total)\Avg. Disk Queue Length
- PhysicalDisk(_Total)\Disk Read Bytes/sec
- PhysicalDisk(_Total)\Disk Write Bytes/sec
- Process(_Total)\Working Set
- Processor(_Total)\% Processor Time
- System\Processor Queue Length

Relevant values from these PerfMon measurements will be mentioned where appropriate. Particularly important were the Processor Queue Lengths as this shows whether your system is CPU bound our not and Physical Disk\Disk Queue Length as this shows whether you are I/O bound or not.

The benchmark major functions that were performed are summarized below:

Benchmark Function/Processing	
OLAP Cube Data Population	
	1. Initial Data Loads (avg)
	2. Subsequent Data Loads (avg)
OLAP Cube Data Exports	
	1. Single data measure exports (avg)
	2. Multiple data measure exports (avg)
OLAP Update (DB calc's, Middle outs, Allocations, DB Updates) - Online	
	1. Single data measure Updates - lowest level (avg)
	2. Single data measure Updates - highest level(avg)
	3. Multiple data measure Updates (avg)
OLAP Cube DB Maintenance	
	1. Realignments (middle level) (avg)
	2. Realignments (lowest level) (avg)
	3. Reindexing (all) (avg)
	4. CheckSync (1 dbf in Cube) (avg)
	5. CheckSync (all dbf's in cube) (avg)
Legacy to OLAP Extract Processing	
	1. Overall Extract/transformation processing (MS SQL Server 2000) (avg)

Figure 4 shows the physical file configuration of an i2 Technologies Demand Planning OLAP cube.

Benchmark Findings cont.

There are two main types of files; binaries (dbf) and indexes (cdx). An OLAP cube consists of many of these for varying purposes. In general these are: System/Control dbf's (binaries) and Physical data files dbf's (binaries) and cdx's (indexes).

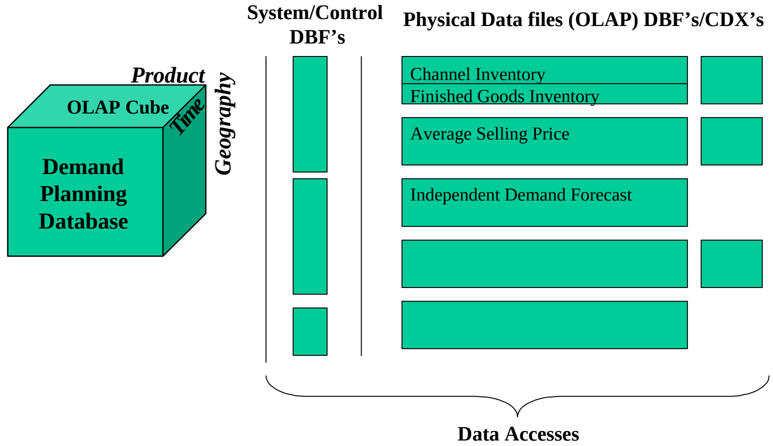


Figure 4
I2 Demand Planner Physical File implementation

Any of these files are candidates for identification as being "Hot" depending on what is being executed. A thorough technical evaluation should be done on any SCM implementation to determine which are potential "hot" files. This benchmark chose to consider all of the files as "hot" to maximize performance and to minimize implementation/configuration changes for transparency reasons.

Baseline Benchmark times (1-way CPU)

The benchmark functions/processes were first run on the HP LC2000 NetServer. This contained our current production i2 Technologies Demand Planning OLAP cube and the legacy database tables (MS SQL Server 2000). The following is the complete summary of these baseline executions:

Benchmark Function/Processing		HP/LC2000 NetServer (in seconds)
OLAP Cube Data Population		
	1. Initial Data Loads (avg)	750
	2. Subsequent Data Loads (avg)	679
OLAP Cube Data Exports		
	1. Single data measure exports (avg)	73
	2. Multiple data measure exports (avg)	158
OLAP Update (DB calc's, Middle outs, Allocations, DB Updates) - Online		
	1. Single data measure Updates - lowest level (avg)	922
	2. Single data measure Updates - highest level(avg)	1052
	3. Multiple data measure Updates (avg)	2292
OLAP Cube DB Maintenance		
	1. Realignments (middle level) (avg)	437
	2. Realignments (lowest level) (avg)	308
	3. Reindexing (all) (avg)	121
	4. CheckSync (1 dbf in Cube) (avg)	1368
	5. CheckSync (all dbf's in cube) (avg)	2755
Legacy to OLAP Extract Processing		
	1. Overall Extract/transformation processing (MS SQL Server 2000) (avg)	1790

During these baseline executions CPU Processor Queue Lengths increased up to as much as 14 (averaged about 6) for most of the data loading and updating operations, System\Processor Time% averaged 99.89%, and the Physical Disk\Disk Queue Lengths averaged between 7 indicating both an I/O bound and CPU bound system.

SSD Device Upgrade Benchmark times (1-way CPU)

The SSD device was then installed on the original HP LC2000 NetServer (1-way CPU) to see if this could help out in any way. The entire Application (both i2 Technology Demand Planner environment and MS SQL Server 2000 legacy database environment) were then physically moved over to this SSD device without much issue at all (very easy to do – and keeping with our hardware transparency approach). The results were fairly good overall. The results yielded an improved throughput up to as much as the CPU could cope with.

Specifically, the CPU Processor Queue Lengths increased to as high as 25 (averaged about 11) with a slight decrease in Physical Disk\Disk Queue Lengths (because it is now a SSD device). Improvements occurred across the board, but the major limitations of not enough CPU cycles created a brick wall (can't process the commands fast enough to realize full benefit from faster I/O). In affect it was severely CPU bound. Based on the type of processing that were part of this benchmark, an addition of between 3 or 4 CPU's would be the starting place for processing power upgrades (if you were not satisfied with the improvements gained with just the SSD device upgrade). It could very well be that these gains more than satisfy your needs.

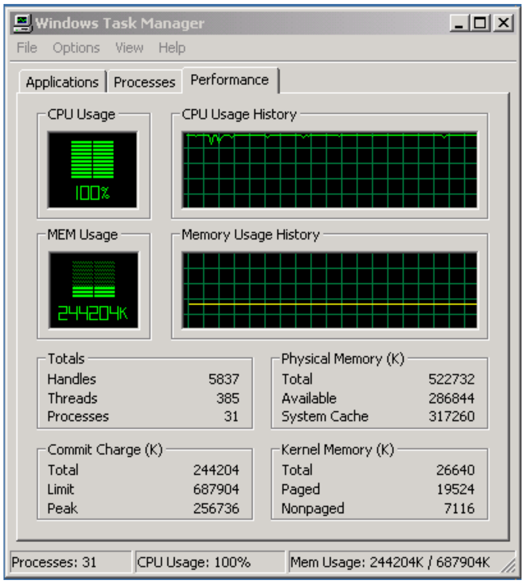


Figure 5
Performance tab – Task Manager – CPU saturated to 100% - with SSD

The following is the complete summary of the SSD upgrade executions along with the percentage improvement from the original baselines:

		HP/LC2000 (seconds)	HP/LC2000 + SSD (seconds)	Upgrade (LC2000:SSD) (% Improvement)
Benchmark Function/Processing				
OLAP Cube Data Population				
	1. Initial Data Loads (avg)	**750**	**390**	**48.00% (1.92x)**
	2. Subsequent Data Loads (avg)	**679**	**375**	**44.78% (1.81x)**
OLAP Cube Data Exports				
	1. Single data measure exports (avg)	73	67	8.22% (1.09x)
	2. Multiple data measure exports (avg)	158	90	43.04% (1.75x)
OLAP Update (DB calc's, Middle outs, Allocations, DB Updates) - Online				
	1. Single data measure Updates - lowest level (avg)	**922**	**370**	**59.87% (2.49x)**
	2. Single data measure Updates - highest level(avg)	**1052**	**650**	**38.22% (1.62x)**
	3. Multiple data measure Updates (avg)	**2292**	**1375**	**40.01% (1.66x)**
OLAP Cube DB Maintenance				
	1. Realignments (middle level) (avg)	437	284	35.02% (1.54x)
	2. Realignments (lowest level) (avg)	308	200	35.07% (1.54x)
	3. Reindexing (all) (avg)	121	102	15.70% (1.18x)
	4. CheckSync (1 dbf in Cube) (avg)	1368	1245	8.99% (1.09x)
	5. CheckSync (all dbf's in cube) (avg)	2755	2341	15.02% (1.17x)
Legacy to OLAP Extract Processing				
	1. Overall Extract/transformation processing (MS SQL Server 2000) (avg)	1790	1741	2.74% (1.03x)

Strong showing in the **load** and **update** processing. Elsewhere, only fair improvements.

Server Upgrade Benchmark times (5-way CPU)

The entire Application (both i2 Technology Demand Planner environment and MS SQL Server 2000 legacy database environment) were then physically moved over to the new HP LH6000 NetServer. We will be interested in how much a more powerful server will increase performance (at the I/O and CPU levels) since we are going from a 1-way CPU to a 5-Way CPU.

Going to a 5-way LH6000 NetServer is an excellent performance upgrade to consider and is one that has minimal impact on the i2 Technologies configuration and implementation (completely transparent to the implementation).

Figure 6 shows the huge load being put on the server during update processing executions with this 5-way CPU server.

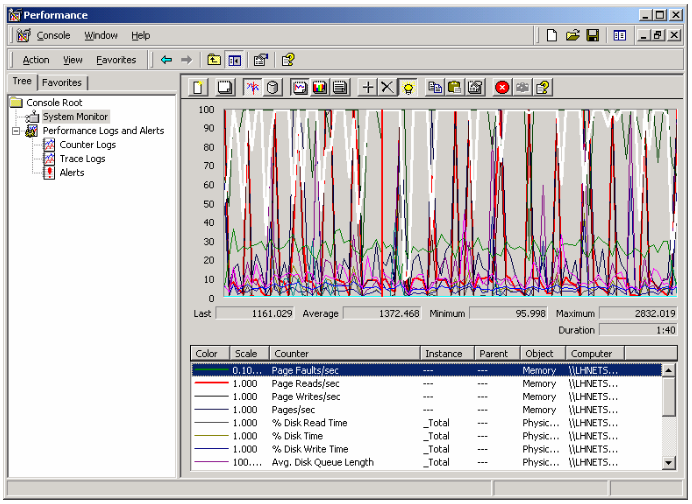


Figure 6
PerfMon – System Monitor – counters during update executions

HP Server Upgrade Benchmark times cont.

The following is the complete summary of the server upgrade executions along with the percentage improvement from the original baselines:

	HP/LC2000 (seconds)	HP/LH6000 (seconds)	Upgrade (LC2000:LH6000) (% Improvement)
Benchmark Function/Processing			
OLAP Cube Data Population			
1. Initial Data Loads (avg)	**750**	**305**	**59.33% (2.46x)**
2. Subsequent Data Loads (avg)	**679**	**295**	**56.55% (2.30x)**
OLAP Cube Data Exports			
1. Single data measure exports (avg)	73	44	39.73% (1.66x)
2. Multiple data measure exports (avg)	158	116	26.58% (1.36x)
OLAP Update (DB calc's, Middle outs, Allocations, DB Updates) - Online			
1. Single data measure Updates - lowest level (avg)	**922**	**373**	**59.54% (2.47x)**
2. Single data measure Updates - highest level(avg)	**1052**	**444**	**57.79% (2.37x)**
3. Multiple data measure Updates (avg)	**2292**	**885**	**61.39% (2.59x)**
OLAP Cube DB Maintenance			
1. Realignments (middle level) (avg)	**437**	**280**	**35.93% (1.56x)**
2. Realignments (lowest level) (avg)	**308**	**150**	**51.30% (2.05x)**
3. Reindexing (all) (avg)	121	46	61.98% (2.63x)
4. CheckSync (1 dbf in Cube) (avg)	1368	1164	14.91% (1.17x)
5. CheckSync (all dbf's in cube) (avg)	2755	2217	19.53% (1.24x)
Legacy to OLAP Extract Processing			
1. Overall Extract/transformation processing (MS SQL Server 2000) (avg)	**1790**	**1228**	**31.40% (1.46x)**

As you can see from the **bolded** line entries above, just by making this server upgrade, we sharply improved most of the major (critical) type of processing.

HP Server Upgrade Benchmark times cont.

However, we have still not achieved the targeted goals yet (of more near 67% improvement – a 3x improvement). We did see the CPU Processor Queue Length go down to ZERO, %Processor Time average 20.1875%, but the Physical Disk/Disk Queue Length still averages between 2 to 3 indicating a system that is NOT CPU bound but now only I/O bound. Figure 7 shows a quick shot of the Performance tab of the Task Manager during one of the multi-row update operations:

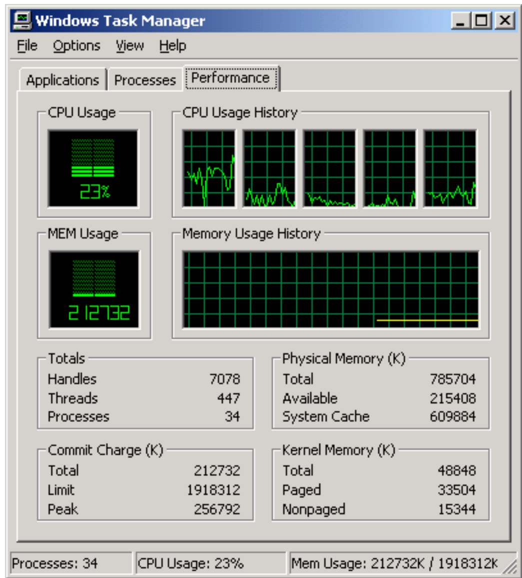


Figure 7
Performance tab: Task Manager 5-way CPU Utilization – low

SSD Device Upgrade Benchmark times (5-way CPU)

We are still I/O bound and it has become apparent that the only way to squeeze more out of this OLAP environment is to isolate the physical data files (binaries and cdx's) onto some type of Solid State Device in order to achieve the goals that we stated earlier. We want to eliminate the physical disk queuing as much as possible. As we saw earlier in **figure 4**, there are many physical files that could be candidates for this isolation. In general, the hottest files (the one's that are accessed the most) would be the highest candidates for isolation. However, in order to make this as transparent as possible, we decided to isolate the entire OLAP cube onto an SSD device. Minimal configuration/implementation changes were needed. We also configured this SSD Disk Device to use compression in order to allow for maximum OLAP Cube size. With compression, a 4.72GB OLAP cube is physically stored on only 514MB of space (all handled at the NTFS level). This allows us to grow our OLAP cube to a much larger size without hesitation and to use a smaller SSD device that is more cost effective to our budget. With our eyes on the Physical Disk counters, we ran the benchmark again with the SSD device. The results are as follows:

		HP LH6000 (sec)	HP LH6000 +SSD (sec)	Upgrade (LH6000:SSD) % Improvement	Overall Improvement %
Benchmark Function/Processing					
OLAP Cube Data Population					
	1. Initial Data Loads (avg)	**305**	**205**	**32.79% (1.49x)**	**72.67% (3.66x)**
	2. Subsequent Data Loads (avg)	**295**	**200**	**32.20% (1.47x)**	**70.54% (3.39x)**
OLAP Cube Data Exports					
	1. Single data measure exports (avg)	44	18	59.09% (2.66x)	75.34% (4.05x)
	2. Multiple data measure exports (avg)	116	95	18.10% (1.22x)	39.87% (1.66x)
OLAP Update (DB calc's, Middle outs, Allocations, DB Updates) - Online					
	1. Single data measure Updates - lowest level (avg)	**373**	**246**	**34.05% (1.52x)**	**73.32% (3.75x)**
	2. Single data measure Updates - highest level(avg)	**444**	**319**	**28.15% (1.39x)**	**69.68% (3.30x)**
	3. Multiple data measure Updates (avg)	**885**	**568**	**35.82% (1.56x)**	**75.22% (4.03x)**
OLAP Cube DB Maintenance					
	1. Realignments (middle level) (avg)	**280**	**225**	**19.64% (1.24x)**	**48.51% (1.94x)**
	2. Realignments (lowest level) (avg)	**150**	**82**	**45.33% (1.83x)**	**73.38% (3.76x)**
	3. Reindexing (all) (avg)	46	28	39.13% (1.64x)	76.86% (4.32x)
	4. CheckSync (1 dbf in Cube) (avg)	1164	961	17.44% (1.21x)	29.75% (1.42x)
	5. CheckSync (all dbf's in cube) (avg)	2217	1760	20.61% (1.26x)	36.12% (1.56x)
Legacy to OLAP Extract Processing					
	1. Overall Extract/transformation processing (MS SQL Server 2000) avg	**1228**	**905**	**26.30% (1.36x)**	**49.44% (1.98x)**

The Physical Disk Queue Lengths came down to near ZERO (0.089925 average) as we had hoped. CPU stayed about the same and we gained from 17.44% (1.21x) to 59.09% (2.66x) improvement across the board in execution times with the Solid State Device in place. Total improvements from our original baselines coming in well into the desire goals we stated up front with the critical ones of Updates, and Loads well above our 67% (3x) goal. As an added bonus, by using the SSD device, we automatically gained an implementation on a fail-over architecture (it comes built in).

Benchmark Conclusions

Our intention was to make performance and scalability changes that were as transparent as possible to our software implementation using primarily a hardware approach. We initially gained good improvements by simply putting the SCM OLAP implementation on an SSD device with the original 1-way CPU server. This didn't reach our goals but may be enough for folks that don't want to do massive server upgrades. We gained improvements by throwing many more CPU's at the application, but again couldn't quite meet our improvement goals because of the nature of OLAP physical architectures (highly I/O bound). By adding in the SSD device, we were able to achieve the performance we desired and didn't have to modify the software configuration or implementation. This is extremely important due to the complexity of SCM OLAP implementations (they are very messy, at best, and there aren't very many performance handles that can be directly referenced). The cost effectiveness of this approach can easily be justified. The short amount of time that the entire system was migrated to the new server and the SSD device was extremely small.

It was possible to achieve the declared performance goals of at least 67% (3x) improvements to gain that next level of performance and scalability for your OLAP (or other Value Chain) software implementations. This will be possible by orchestrating a physical system upgrade to more CPU power along with well placed SSD devices and do it with minimal impact on the software architecture itself (perhaps the most attractive part).

The same underlying approach can be taken with most OLAP implementations regardless of if they are i2 Technologies OLAP, Microsoft Analytic Services OLAP, or others. Each can benefit from the hot file isolation onto SSD Devices and the transparent hardware upgrade approach used here. The ease, simplicity, and rapid deployment of this type of approach makes it an extremely viable path to higher performance and scalability.

Benchmark Participants

The benchmark was created and implemented by Paul Bertucci, Chief Technology Advisor to 4Points Technologies, Inc., (www.4pointstech.com) and Managing Principal of Database Architechs, (www.dbarchitechs.com). Testing was performed in conjunction with 4Points Technologies and hardware and operating system setup help from Raja Sekhar Botchu of 4Points. Mr. Bertucci is a recognized expert in Database Design and Performance & Tuning. He is the author of Sybase's "Physical Database Design" and "Performance & Tuning" courses, author of "Microsoft SQL Server 2000 Unleashed" and "ADO.NET in 24 Hours" books from SAMS Publishing, and an expert in Supply Chain Management implementations. He has worked extensively with Oracle, Sybase, MS SQL Server, DB2, OLAP and other DBMS's for 20+ years for Fortune 500 companies around the globe.

Hardware Providers

Hewlett Packard provided the LH6000 NetServer (www.hp.com)
Solid Data Systems provided the SSD Devices (www.soliddata.com)