Monash
Information Services

M

Memory-Centric Data Management

A Monash Information Services White Paper

by

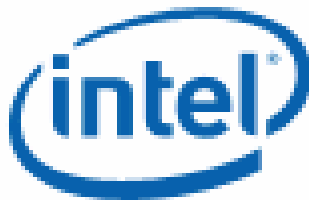**Curt A. Monash, Ph.D.**

May, 2006

Version 1.01

Sponsored by:

Table of Contents

About this Paper

This is a sponsored yet independent white paper on the subject of memory-centric data management technology. It was subject to review but not to approval from its five sponsors: Applix, Intel, Progress Software (Real Time Division), SAP, and Solid Information Technology. The principal author and editor of this paper is Curt A. Monash, Ph.D., president of Monash Information Services. However, the appendices reflect the words and opinions of the sponsors themselves. The paper can be downloaded from www.monash.com/MCDM.pdf. Information about the author can be found at www.monash.com/curtbio.html.

Defining Memory-Centric Data Management

The term "memory-centric data management" was recently coined by Curt Monash, to cover a group of related products or technologies with two common characteristics:

- They manage data primarily in RAM, rather than on disk.

- They have a rich data manipulation language, just as DBMS do.

Typically, data is loaded from a persistent disk-based data store, and may have changes logged onto that same store. However, the performance characteristics, optimizations, and bottlenecks of a memory-centric data management product are primarily related to semiconductor RAM (Random Access Memory), not to disk.

Executive Summary

Memory-centric data management products improve performance/cost.

Two of IT's most consistent mandates are lower costs and more speed. To meet this demand for cheap speed, standard disk-based DBMS need help. Increasingly, this help is coming in the form of memory-centric data management technology.

They break through the disk speed bottleneck.

Conventional DBMS are designed to get data on and off disks as safely, quickly, and flexibly as possible. Much of their optimization is focused on one key bottleneck – the slow speed with which a random byte of data can be found on disk, up to 1 million times as long as it might take to find the same byte in RAM. But the optimizations and access methods designed to address this bottleneck don't do as well once the data is safely in main memory. Memory-centric data management tools, using access methods that would be unsuitable for a disk-centric set-up, can perform vastly better.

Hardware advances have paved the way.

The rise of memory-centric technology is closely tied to the advent of 64-bit chips and highly-parallel architectures. A single 32-bit CPU can only manage 1-3 gigabytes of in-memory data. But servers with multiple 64-bit Intel CPUs routinely handle 100 gigabytes and more.

Memory-centric technology powers highly demanding apps.

As a result, memory-centric technology now powers some of the most demanding mission-critical systems; examples include program stock trading, telecommunications equipment, airline schedulers, and the Amazon.com bookstore. Memory-centric technology also provides blindingly fast analysis of databases up to half a terabyte or more in size.

Memory-centric technology's benefits are staggering.

The benefits of specialty data managers – as measured in cost, performance, and or functionality – can be enormous. This is particularly true in the case of memory-centric data management tools. On the OLTP side, many demanding memory-centric applications would just be impossible using disk-centric products. And the situation is similar for OLAP. Memory-centric technology permits real-time analysis for complex problems, even when disk-centric technology's performance is more like "come back after lunch."

Memory-centric data managers are specialized in specific areas.

Memory-centric data managers typically rely on unconventional data structures that are best suited for specific kinds of tasks. Accordingly, different kinds of application are best served by different memory-centric products. For example:

- Applix's TM1 supports real-time interactive multidimensional OLAP analysis, with much more speed and computational flexibility than disk-centric OLAP alternatives.
- SAP's BI Accelerator (available as a part of NetWeaver) offers highly

interactive hybrid OLAP analysis, with much faster and more consistent performance than is offered by conventional RDBMS.

- Solid Information Technology's BoostEngine is optimized for high-speed OLTP of telecommunication network data. In particular, it allows cost-effective diskless replication of relational OLTP databases across many nodes of a network.

- Progress Software's ObjectStore serves up complex in-memory objects, allowing OLTP in scenarios that could choke even the most powerful relational systems.

- Progress Software's Apama event stream processing products allow the filtering of stock tickers and other high-speed information streams. No product that requires staging the data to disk can handle these real-time tasks.

It's OK to use specialty data stores.

Large DBMS vendors often argue against specialty data management technologies, claiming that there are a variety of technical, administrative, and cost benefits when enterprises consolidate databases and DBMS brands. However, we find these arguments to be unconvincing -- indeed, they are contradicted by those vendors' own strategies. Each of the leading DBMS providers offers a variety of fundamentally different data stores, and most are increasing the number of their offerings, not decreasing them. Meanwhile, virtually every large enterprise has long managed multiple data stores, and will surely do so for many years into the future.

Memory-centric data management is appropriate for many enterprises.

The simplest way to judge whether memory-centric technology is suitable for your enterprise is to start with this question: How much would faster data management performance be worth to you? If the answer is "Not much," you probably should stick with conventional DBMS technology. But if new functionality, reduced user wait times, or simple price/performance have high potential value, you should investigate memory-centric data management technology.

Introduction

Conventional DBMS don't always perform adequately.	Ideally, IT managers would never need to think about the details of data management technology. Market-leading, general-purpose DBMS (DataBase Management Systems) would do a great job of meeting all information management needs. But we don't live in an ideal world. Even after decades of great technical advances, conventional DBMS still can't give your users all the information they need, when and where they need it, at acceptable cost. As a result, specialty data management products continue to be needed, filling the gaps where more general DBMS don't do an adequate job.
Memory-centric technology is a powerful alternative.	One category on the upswing is **memory-centric data management** technology. While conventional DBMS are designed to get data on and off disk quickly, memory-centric products (which may or may not be full DBMS) assume all the data is in RAM in the first place. The implications of this design choice can be profound. RAM access speeds are up to 1,000,000 times faster than random reads on disk. Consequently, whole new classes of data access methods can be used when the disk speed bottleneck is ignored. Sequential access is much faster in RAM, too, allowing yet another group of efficient data access approaches to be implemented.
It does things disk-based systems can't.	If you want to query a used-book database a million times a minute, that's hard to do in a standard relational DBMS. But Progress' ObjectStore gets it done for Amazon. If you want to recalculate a set of OLAP (OnLine Analytic Processing) cubes in real-time, don't look to a disk-based system of any kind. But Applix's TM1 can do just that. And if you want to stick DBMS instances on 99 nodes of a telecom network, all persisting data to a 100th node, a disk-centric system isn't your best choice – but Solid's BoostEngine should get the job done.
Memory-centric data managers fill the gap, in various guises.	Those products are some leading examples of a diverse group of specialist memory-centric data management products. Such products can be optimized for OLAP or OLTP (OnLine Transaction Processing) or event-stream processing. They may be positioned as DBMS, quasi-DBMS, BI (Business Intelligence) features, or some utterly new kind of middleware. They may come from top-tier software vendors or from the rawest of startups. But they all share a common design philosophy: Optimize the use of ever-faster semiconductors, rather than focusing on (relatively) slow-spinning disks.
They have a rich variety of benefits.	For any technology that radically improves price/performance (or any other measure of IT efficiency), the benefits can be found in three main categories:

- Doing the same things you did before, only more cheaply;
- Doing the same things you did before, only better and/or faster;
- Doing things that weren't technically or economically feasible before at all.

For memory-centric data management, the "things that you couldn't do before at all" are concentrated in areas that are highly real-time or that use non-relational data structures. Conversely, for many relational and/or OLTP apps, memory-centric technology is essentially a much cheaper/better/faster way of doing what you were already struggling through all along.

Memory-centric technology has many applications.

Through both OEM and direct purchases, many enterprises have already adopted memory-centric technology. For example:

- Financial services vendors use memory-centric data management throughout their trading systems.
- Telecom service vendors use memory-centric data management in multiple provisioning, billing, and routing applications.
- Memory-centric data management is used to accelerate web transactions, including in what may be the most demanding OLTP app of all -- Amazon.com's online bookstore.
- Memory-centric data management technology is OEMed in a variety of major enterprise network management products, including HP Openview.
- Memory-centric data management is used to accelerate analytics across a broad variety of industries, especially in such areas as planning, scenarios, customer analytics, and profitability analysis.

The memory-centric market is growing up.

Memory-centric data management has traditionally been the province of small vendors. But the market is growing up. Specialists such as Applix and Solid grow larger every year, and have built impressive customer lists. Progress entered more recently, followed by SAP and Oracle within the past year.*

*SAP actually has a long history of developing and using advanced memory management technology, specifically in LiveCache and also in some middle-tier OLTP caching that predated comparable uses of products like TimesTen. But BI Accelerator is SAP's first product with DBMS-like user accessibility, which is a key requirement of our product category definition.

Users want this technology because of its speed.

From a user standpoint, the biggest driver for memory-centric technology is the need-for-speed. If many thousands of people all want something RIGHT NOW, getting it off of disk can create intolerable bottlenecks. And if a handful of users all want to run complex analyses at the same time, bottlenecks can also ensue.

Vendors meet this desire because they can.

On the vendor side, much of the growth in memory-centric offerings can be tracked to hardware advances. 64-bit chips and massive parallelism make it possible to use much more RAM than before. And Moore's Law makes that theoretical possibility affordable in practice.

Many enterprises should use memory-centric technology.

Some smaller enterprises may come close to the single-database ideal, running one mixed-used RDBMS (Relational DBMS) for most or all of their transactional apps and data-based analytics alike. For those, the benefits of memory-centric technology may not be compelling. But at most large enterprises, there are one or more areas where memory-centric technology could be very appealing. Memory-centric OLAP allows better, more interactive analytics, and almost any organization can use that. And an in-memory cache or front-end can help ease the most demanding OLTP challenges.

The Memory-Centric Difference

Computing power grows exponentially.	By most measures, computing power doubles every couple of years. Whether you're looking at CPU (Central Processing Unit) speed, RAM (Random Access Memory) capacity, RAM capacity per unit of cost, disk storage density, network throughput, or some other similar metric – all of these are subject to some version of Moore's Law. That is, they improve by a factor of 2 every couple of years or so. For example, in a little over two decades, the standard size of a PC hard disk has increased from 10 megabytes to 80 or 160 gigabytes, for a total of 13 or 14 doublings.
	Note: PCs and servers use substantially similar components these days, so it's appropriate to use numbers from either class of machine.
Disk speed is the main exception.	But there's one huge exception to this trend. The rotational speed of disks is limited by their tendency to "go aerodynamic" – i.e., to literally fly off of the spindle. Hence this speed has grown only 12.5-fold in a half a century, from 1,200 revolutions per minute in 1956 to 15,000 RPM today.
So disk random access remains painfully slow.	The time to randomly access a disk is closely related to disk rotation speed. A 15,000 RPM disk makes half a rotation every two milliseconds (ms) – and such disks are commonly advertised as having 3.3-3.9 ms seek times. That's almost a million times longer than raw RAM seek times, which have declined to just a few nanoseconds. And it's over 1000 times slower than what may be the real in-memory speed bottleneck -- the 1 gigabyte/second interprocessor interconnect rate.
The disk speed bottleneck chokes DBMS design.	Conventional DBMS vendors are held hostage by their fear of truly random disk access. Data is moved around in large sequential blocks. Much development effort goes into clustering likely query results into the same region of a disk. And many of the cleverest data access algorithms are simply rejected, because they're not practical at such low access speeds.
Users suffer accordingly.	These design limitations have a great impact on user functionality. OLAP is rarely real-time, or if it is, its functionality is much more limited than that of Microsoft Excel. Object data structures – at least from conventional RDBMS vendors – are either very simple or very slow to use. Network event and other event-stream processing can be so slow as to render conventional DBMS useless for those purposes. Even relational data warehouse functionality is limited by the cost of providing acceptable performance.
Memory-centric data management clears it.	For applications suffering from such limitations, there can be a superior alternative to conventional disk-based data management. **Memory-centric data**

management blasts through the disk speed bottleneck. And with recent improvements in addressability, parallelization, and general chip price/performance, memory-centric technology is becoming feasible for an ever greater number of applications and use-case scenarios.

The key is exploiting RAM.

Memory-centric architectures are very different than disk-based ones. Disk-based ones look at blocks of memory. (They have to rely on data being physically stored in a predictable arrangement, so as to get it off disk quickly despite the slow rotation speed.) Memory-centric ones, however, go straight to the target, exploiting the R in Random.

Conventional DBMS caches can't keep up.

The most natural competitor for memory-centric technologies is the cache of a conventional disk-based system. However, systems designed to be memory-centric have several major advantages over those that simply use disk-oriented data structures. Specialized memory-centric systems often fit several times as much raw data into a given amount of RAM than a conventional DBMS can manage. Their memory-optimized data structures can also give advantages in performance and throughput. Finally, in the non-relational implementations, they have powerful functionality that disk-based systems simply don't match.

Hardware progress fuels memory-centric adoption.

The rise of memory-centric data management is closely related to advances in platform technology. Different applications stress different aspects of the hardware, but there are two consistent themes:

1. *More RAM.* 32-bit CPUs can only address 2-4 gigabytes of RAM, depending on the operating system. 64-bit processors blast through this barrier. It is currently possible to put 8 or 16 or even 64 gigabytes of RAM on a single board, and those numbers are growing fast. Even more important, it is practical to put up to 16 boards in the same box, and bigger systems yet can be created, limited only by high-end network speeds.

2. *More CPUs.* CPU proliferation isn't just crucial to memory-centric OLAP. It's also important on the OLTP side, although usually in a more loosely-coupled way. Whether acting as a cache for application servers or as a monitor replicated across a number of network nodes, memory-centric OLTP tools gain in value as systems get more powerful and more real-time.

Memory-centric technology takes a variety of forms.

Memory-centric technology takes different forms when applied to different kinds of datasets and application requirements. Notably:

100 gigabytes of RAM can hold a large data mart.

- Relational query processing is vastly faster if all the data is in RAM to begin with. The more complex the query is, the bigger the benefit. It is now practical to put 100 gigabytes of data in RAM – which may reflect

half a terabyte of raw data, or several terabytes of disk storage in a conventional DBMS architecture. That amounts to a good-sized data mart, or a small data warehouse, with much faster processing than is practical using disk-based technologies. SAP offers this technology in its *BI Accelerator.*

Memory-centric MOLAP is far superior to disk-based.

- For MOLAP (multidimensional OLAP) databases, memory-centric processing has long been superior to disk-based. Now that modern hardware has eliminated the previous database size limit (about 1-3 gigabytes), larger enterprises should take another look at the technology. Not only are performance and database/index size much superior, but greater functionality is possible as well, especially in the realm of complex business modeling. (Specifically, calculations are possible that refer to multiple cubes, with flexibility that precomputation-oriented disk-based systems can't match.) Applix offers this capability in its *TM1.*

Memory-centric OODBMS can have blazing OLTP performance.

- Object-oriented DBMSs provide a good match to OO programming languages such as Java. More important, they can provide high-performance infrastructure to applications whose complex data structures give RDBMS fits. But the most natural implementation for an OODBMS relies on chains of random-access pointers, and that architecture is only performant in a memory-centric implementation. Progress's *ObjectStore* boasts some extremely high-performance applications, notably the Amazon.com bookstore – a huge application distributed over a large number of application servers.

Memory-centric technology deserves broad consideration.

Indeed, it's hard to think of an area of computing where memory-centric technology shouldn't be considered. If you can circumvent the disk speed bottleneck altogether, you should do so. If you need transactional persistence, then hybrid or caching memory-centric solutions can provide major performance boosts. Really, the only two types of applications for which memory-centric technology shouldn't be considered are:

1. Applications small enough that the performance improvement isn't worth the investment.
2. Applications so big that the RAM required for memory-centric technology is unaffordable.

Memory-Centric Analytic Processing

Inherent Problems of Disk-Based OLAP

Disk-centric query performance is a challenge.

Reasonably flexible query and analysis technology was first introduced over 20 years ago. For all that time, it has caused IT managers one overriding fear: People might actually use it. No matter how carefully designed a database you have, there's always a potential "query from hell" that brings it to its knees.

Basic RDBMS storage is optimized for OLTP.

Getting good query performance out of relational databases is not easy. In its simplest form, RDBMS technology is optimized for OTLP much more than for analytic processing. For example, data is stored in the shortest rows that make sense. This reduces the cost of updates -- but analyzing the data can require expensive joining of many different short-row tables. Similarly, the central data access method for RDBMS is the b-tree – ideally suited for handling one disk-based record at a time, but not so great for finding whole ranges of results.

One way to boost query response is aggressive indexing.

The traditional way to boost DBMS performance on analytic queries is clever indexing. Essentially, database designers anticipate the types of queries likely to be asked, and prebuild indices that will deliver results quickly. Star schemas and their relatives are popular for this purpose, providing what in effect are precomputed joins. Many other techniques, such as bitmaps, are used too.

Another is preaggregated calculations.

Another major boost to analytic DBMS performance is a different kind of precalculation, namely precomputation of aggregated sums, averages, and the like. (Depending on the vendor, this may or may not be the principal meaning of "materialized view.") Indeed, many queries never get down into the finest-grained level of detail. For those, precalculating results can provide orders-of-magnitude performance improvements over more primitive approaches.

Both cause databases to balloon in size.

But all this precalculation and prebuilding of indices comes at a high cost – you have to store all that stuff. And you have to get it on and off of disk. Typically, the indices wind up being several times as large as the raw data itself. In MOLAP implementations, the "data explosion" can be by multiple orders of magnitude. Here's why this problem is very hard to avert.

OLAP usually starts with star schemas.

Most OLAP implementations are based on one primary concept – what is now called a "star schema" – and one of two primary implementation strategies: denormalized relational storage (ROLAP, for Relational OLAP), or arrays (MOLAP, for Multidimensional OLAP). Either way, the same thing is happening logically. For each tuple, every element except one is chosen from a pre-ordained list of possible values; these elements are called *dimensions*. Only one element – the *fact* – is truly free.

Note: More precisely, star schemas are the simplest case. But the same data explosion problem occurs (for example) in more complex snowflake schemas, for exactly the same reasons.

Explosion results, even with sparsity compression.

This simplifying assumption is very powerful, allowing fast answers to queries of exactly the sort people ask during many kinds of business analysis. But it can exact a large implementation-cost price. Logically, this model causes exponential explosion in the number of values in the database. Fortunately, most of these values are zero, so sparsity compression eliminates *much* of the problem. Much, but not *all* -- given the need to respect the disk speed bottleneck, disk-centric sparsity compression is very imperfect.

Precalculation makes the problem much worse.

What's worse, if one is aggressive about precalculating possible aggregate values, the set of aggregate results is still exponentially large, yet no longer all that sparse. This creates quite a dilemma. Thus, ROLAP implementations typically forgo both the costs and benefits of aggressive precalculation. But even so, they typically have indices several times as large as the raw data itself. And in MOLAP databases, where aggressive precalculation is indeed the norm, data can explode by multiple orders of magnitude.

High costs and low functionality result.

All this extra data has to be stored, potentially at high cost if the database is large. What's more, even though in some ways it boosts speed, its sheer volume is also a continual performance drag. Because of these costs, functionality tradeoffs are made. In the most common limitation, many complex relational queries simply can't be answered in real time. And if you want real-time access to your MOLAP models for what-if analyses, like you have with your spreadsheets, you're totally out of luck … unless you take advantage of memory-centric technology.

Memory-Centric Solutions

Memory-centric technologies avert these problems.

Some of OLAP's toughest problems can be solved by memory-centric technology. Ideally, a whole data mart is loaded into main memory. Any kind of analysis can be done against it at a high, consistent speed. Because of superior data access methods that are practical only when in RAM, the whole exercise consumes less silicon and disk – and hence costs less money – than one might think.

SAP and Applix offer examples.

Fortunately, this ideal is pretty close to practical reality. The two best examples we know of are SAP's BI Accelerator/NetWeaver, with hybrid relational/multidimensional OLAP features, and Applix's TM1, offering a purer in-memory MOLAP. Both are sold primarily with application product suites; but both (especially the Applix product) are also self-contained enough to use on a standalone basis. SAP's product was only recently released, and is targeted at a market (100 GB+ data mart sizes) that wouldn't even make sense without

recent hardware developments that make that quantity of RAM addressable. Applix's has been around longer, racking up a decent customer base despite having being previously confined to the 1-3 gigabyte data mart sizes that 32-bit technology allowed. We think it deserves a new look from larger enterprises now that larger memory sizes have become practical.

SAP's BI Accelerator (Memory-Centric HOLAP)

SAP's BI Accelerator provides hybrid memory-centric OLAP.

SAP's *BI Accelerator* starts from the same simplifying assumption as the disk-based solutions cited above: The data is in a star-schema-like format. Indeed, SAP has long arranged its analytic databases in InfoCubes, which are indeed star-schema (or snowflake) data marts. These are hosted on relational or hybrid* OLAP disk-based systems (and now in memory-centric hybrid OLAP too!) as the customer prefers.

*In our usage, "hybrid" OLAP (HOLAP) systems are ones that can be addressed by a multidimensional DML such as MDX, but even so have largely relational data structures under the covers. Microsoft Analysis Services, which shifts data transparently between relational and MOLAP servers, is the classic product defining this category.

It sticks a whole relational data mart into RAM.

The basic idea of BI Accelerator is to stick an entire InfoCube in RAM, then run blazingly (and consistently) fast full table scans against it to execute almost any possible query. Some serious optimization was needed to work around the bottleneck of "slow" 1 gigabyte/second interprocessor communication rates, but still the platform exploited is much faster than one that includes any kind of data transfer from disk.

Performance benefits are impressive.

The performance benefits are impressive. Query-from-hell performance can be improved by 100 times or more. Even queries that already ran quickly are commonly speeded up several-fold. Indeed, one of the first BI Accelerator customers reported an overall blended speedup in the two orders of magnitude range. Using BI Accelerator today requires having SAP's NetWeaver product stack (which most SAP customers of course already have), plus a highly parallel piece of hardware based on Intel processors, and shipped on hardware from a choice of IBM or Hewlett Packard. Even so -- where query speed and data warehouse expense matter, BI Accelerator's benefits can far exceed its costs.

Avoiding indices boosts cost effectiveness.

For the same raw data, BI Accelerator can easily have a 5:1 size advantage vs. the alternative of a conventional RDBMS tuned for data warehousing. This differential -- a key part of the product's cost effectiveness – has two main sources. One is the absence of data explosion, and indeed the near absence of indices whatsoever. With minor exceptions, BI Accelerator answers queries via what amounts to full table scans.

So does superior data compression.

In addition, BI Accelerator boasts better data compression than is practical in disk-based systems. The idea behind most database compression techniques is to turn tabular structures into lists of values – lists of distinct values (and where they occur), lists of non-zero values (the simplest case), and so on. Up to a point, these approaches can be used effectively on disk. But the decompression step of looking up data can involve extra, random disk accesses, and hence can get choked off at the disk-speed bottleneck. But memory-centric technology doesn't share the same limitation.

A column-centric architecture is good for query performance.

BI Accelerator's compression advantages are closely tied to a core aspect of its architecture – it is column-centric. Column-centric systems store and process data columns-first rather than rows-first. For update performance, this is not the best design. But for queries, which typically look at certain columns across a whole large set of records, it's often superior. BI Accelerator's compression techniques are closely related to the data structures that make column-centricity viable in the first place.

Note: "Bitmaps" are the simplest column-centric approach, although except in very low-cardinality cases pure bitmaps aren't viable. The best developed column-centric systems may well be text indexing engines, and BI Accelerator indeed grew out of SAP's TREX text indexing functionality.

Reliable, blazing performance is a very nice thing.

As SAP correctly points out, BI Accelerator isn't just about better performance. It's about getting fast response times consistently. Analysts can develop the justified expectation of quasi-real-time query response. Operational BI can operate reliably, with little concern about how complex the underlying queries may be. If you want to weave analytics into the fiber of your business, that kind of performance is highly desirable. And if you are indeed investing in quasi-real-time analytics performance, then memory-centric technology such as BI Accelerator is a compelling alternative.

Applix's TM1 (Memory-Centric MOLAP)

MOLAP remains a niche technology, for good reasons.

In 1993, the inventor and chief advocate of relational DBMS shocked the IT world. Dr. E. F. Codd proposed a new, non-relational technology, now called MOLAP, for non-transactional data analysis. MOLAP has never ascended beyond niche status, however, for what we regard as two primary reasons:

1. Relational technology has closed some of the gap with MOLAP's capabilities.
2. The data explosion problem makes MOLAP databases uneconomical, except for small and simple ones.

But memory-centric TM1 obviates the

Applix, however, offers a memory-centric MOLAP product called TM1 that neatly overcomes those objections. All calculations are done on-demand, in

main reason.

memory (except that results are of course cached and reused to answer repetitive queries). Thus, data explosion from preaggregation is completely eliminated. Also, as is usually the case with memory-centric technology, TM1's data access methods provide greater efficiency and compression than could be achieved simply by translating data access techniques from disk.

Complex models can be used in real time.

What's more -- beyond the fast and efficient use of resources, TM1 has another major advantage vs. disk-based technologies. It can actually do arbitrary computations in real time, via a spreadsheet interface. Other MOLAP systems can, by precalculating, simulate a reasonable number of real-time calculations, if the inputs are restricted to a single hypercube or array. But if you want to analyze your data in a substantially more flexible manner, memory-centric technology is the only practical way to go.

Performance results are impressive.

As reported in a forthcoming white paper by Empirix, an independent test showed that: a single Intel processor-based 4 CPU server running TM1 provided subsecond response times to the vast majority (85%) of nearly 24,000 queries generated by 500 concurrent users. The queries included a mix of read, write and complex recalculation requests.

TM1 is a powerful analytics engine.

Applix's TM1 is very well integrated with the world's single most popular analytics tool, Microsoft Excel, and offers access to and interoperability with data in leading ERP systems, including SAP. Even though TM1 isn't as widely integrated with market-leading analytics tools as BI Accelerator, which benefits from its conventional relational interface and all the connectivity of SAP Netweaver, it has computational features that relational systems can't match. So TM1 is well worth considering as a mainstream enterprise analytics engine.

Memory-Centric Transaction Processing

Disk-centric OLTP performance can also be problematic.

Analytic queries aren't the only challenge for traditional disk-centric RDBMS. OLTP queries can be problematic too. They can require just as many joins and just as much expensive processing as most OLAP queries – and with much more demanding response time requirements. Problems also arise in cases where the data never was on disk to begin with. Be they stock tickers, network events, telemetry readings, or perhaps soon also RFID (Radio Frequency IDentification) data, there are increasingly many data streams that demand high-speed, real-time filtering.

One answer: In-memory data structures

For decades, the answer to these problems has commonly been in-memory data structures. If disk storage is completely impractical, or the available DBMS of the era just don't get the job done, smart application developers build a whole in-memory processing system just for the specific task at hand. SAP, for example, has long supported complex business objects that don't map closely to relational structures; indeed, that's pretty much the core idea of its famed BAPI interface.

Memory-centric data managers help with these.

Increasingly, however, packaged memory-centric data management products have arisen, eliminating or at least much lessening the need for difficult custom systems software programming. Right now they are concentrated in three main areas:

- Data caching products such as Progress's ObjectStore.
- Event-stream processing products such as the Progress's Apama line.
- Replication-intensive memory-based DBMS such as Solid Information Technology's BoostEngine.

Middle-tier caching

Back-end DBMS caching does a lot of good.

Conventional RDBMS rely heavily on caches. Basically, if data is accessed, it's left in RAM until pushed out by more recently obtained data. Records and tables that get used repeatedly, therefore, can usually be found right in RAM, neatly circumventing the disk speed bottleneck. This is strictly an automatic systems issue; users and application programs don't have to know whether or not the data is in RAM.

Caching is also needed on the middle-tier.

But while such caching makes the DBMS more efficient, it doesn't help with application server/DBMS inter-tier communications. Thus, software vendors and users have built various kinds of custom app server caching capabilities. For example, SAP estimates that 80% of its transactional read accesses can be

satisfied by lookup tables maintained on the app server, which hold much-reused data such as tax rates and currency conversion factors. But for most users such custom development is too costly; they need off-the-shelf middle-tier caching technology.

Disk-centric DBMS can't handle it.

The ideal solution would extend transparent DBMS caching across server boundaries, covering both database and application servers. Unfortunately, no conventional DBMS vendor has yet found a way to make this strategy work. Oracle seemed to be heading that way for a while, but gave up. IBM, which is a leader in both DBMS and app servers, doesn't seem even to have tried.

Memory-centric data managers can.

So the best practical approach to middle-tier caching is memory-centric data managers. These products are sometimes called "in-memory DBMS," by those who think a DBMS doesn't have to actually store data in a persistent way. As in the example of SAP's custom technology cited above, such products can do a great job of speeding performance and reducing network loads.

Some have further advantages.

But those aren't their only virtues. Progress's ObjectStore, for example, provides complex query performance that wouldn't be realistic or affordable from relational systems, no matter what the platform configuration. Most notably, ObjectStore answers Amazon's million-plus queries per minute; it also is used in other traditionally demanding transaction environments such as airplane scheduling and hotel reservations.

Memory-optimized data structures are key.

ObjectStore's big difference vs. relational systems is that it directly manages and serves up complex objects. A single ObjectStore query can be the equivalent of dozens of relational joins. Data is accessed via direct pointers, the ultimate in random access – and exactly the data access method RAM is optimized to handle. On disk, this approach can be a performance nightmare. But in RAM it's blazingly fast.

There are two major scenarios for using this technology.

So when should you use memory-centric middle-tier caching? In two types of situations. First, and this is the main one, you should use the technology when performance needs mandate it. Second, if your best programmer productivity choice is to use a highly nonrelational structure (which usually means an object-oriented one), memory-centric data management can be a superior alternative to what are usually categorized as object-relational mapping tools.

Event-stream processing

Wall Street funds platform innovation.

For the past two decades, no application area has fueled more platform innovation than securities trading. Huge amounts of money are made and lost within a minute or less. And so enormous amounts of IT investment have been cost-justified by the need to make trading decisions ever faster and ever better.

Its data is primarily tabular.	Securities data lends itself naturally to tabular formats. It consists of short records. Data values are either numeric or chosen from fixed, clean lists of possible character strings (e.g., ticker symbols, brokerage firm identifiers). Transaction boundaries are well-defined. And so relational DBMS deservedly run Wall Street.* **Actually, we suspect Wall Street's use of XML is likely to explode. But for now, it's an essentially relational industry.*
But there's no time to get it onto disk.	But trading decisions – both about what to buy and also about how to execute the trade – are based on subtle inferences, teased out of huge streams of data. A pattern may be identified in half an hour's worth of ticker data, then exploited within three seconds. Disk-centric DBMS can't keep up with these needs. Exacerbating the problem, some of the analysis is based on precise timing and sequencing relationships between trade events, in ways that are awkward or inefficiently handled in conventional disk-based data structures. Therefore, the only realistic way to make these applications work is to first filter the data, then make split-second trading decisions based on it, and only store the information afterwards.
Other apps also must filter data before it is stored.	In other cases, data may be filtered, with only the filtered results ever getting stored at all. RFID tracking data may work that way, for instance, with most of the readings discarded and only movement between zones ever recorded on disk. The same goes for GPS or other vehicle tracking data. Other cases arise in real-time telecommunications analysis, whether for reasons of network security, other network operations, or intelligence/law enforcement.
Event stream processors *are the answer.*	A new class of products has emerged to meet these needs, in a category commonly called *event stream processing*. One of the leaders is Progress, which acquired the Apama algorithmic trading technology and is combining it with the same caching capabilities that underlie ObjectStore. For many stream-filtering applications, this kind of technology is the best alternative.

Hybrid memory-centric RDBMS

Some apps need memory-centric technology AND data persistence.	Sometimes an application requires the benefits of memory-centric technology, yet still needs to store transactional data on disk. This situation commonly arises in network management and telecommunications applications, which commonly share several characteristics:

- There are many nodes, physically distributed. It would be painfully expensive to put a disk at every node. It is also important to minimize the CPU and RAM requirements at each node.

- A lot of data needs to be stored and processed temporarily, e.g. for threat analysis or call routing purposes. However, only a small amount of subsetted or calculated data needs to be stored persistently, such as the raw material for a billing application (caller, starting time, ending time, etc.).
- No ongoing database administration is possible. The system needs to run untouched for the life of the application.

In essence this is event-stream processing, but with simpler data filtering, a more complex topology, and a requirement for data persistence.

The answer is hybrid DBMS with robust replication.

The best fit for these needs is a hybrid DBMS with three major elements:

- Efficient relational OLTP functionality (the data lends itself to a conventional tabular architecture).
- Memory-centric architecture to manage the data while in RAM.
- Robust replication to move data from diskless nodes to where it will actually be written to disk.

Fortunately, these requirements are compatible.

Solid's BoostEngine fits the bill.

Solid Information Technology's *BoostEngine* is such a product. It includes a memory-centric database engine add-on to Solid's disk-centric DBMS *EmbeddedEngine*. EmbeddedEngine is a well-established RDBMS, optimized for compactness and for embedded/unattended use, and sold mainly into the telecommunications and/or device manufacturer markets. In line with the ever more multi-node architectures used in these markets, it has had strong replication capabilities all along. BoostEngine was introduced to meet needs such as super-fast lookup table response, much more efficiently than is possible in disk-centric systems.

Hybrid RDBMS can be the best way to go.

All technologies have limitations, and this one is surely no exception. Its simplified OLTP capabilities assume a stable application environment, and might not do well running (just to say something) a 50-module ERP system. Nor is it optimized for data warehousing. But for the efficient, reliable handling of a few highly demanding applications, a hybrid memory-centric RDBMS such as Solid's is often the best way to go.

Technical Deep Dive: Memory-Centric Data Management vs. Conventional DBMS

Conventional DBMS do extra processing to minimize random accesses.

As we've noted above, disk-based systems are designed to optimize for constraints and requirements that memory-centric products simply don't face. For one thing, they need to make as few random disk accesses as possible. Thus, they want to recognize as few distinct memory addresses as possible. To ensure this, they handle data – including index data – in largish blocks, then process the data within a block to find exactly what they were looking for.

Memory-centric systems can use a broad variety of access methods.

Memory-centric systems, however, are free to retrieve exactly the record they want – or the exact pointer, tree node, small hypercube, and so on. This lets them use all sorts of data access methods that are well-known to computer scientists, but rarely practical for implementation in high-performance disk-based DBMS. And so a memory-centric system often performs much faster than a disk-based system, even when the disk-based system has already pulled all needed data into its in-memory cache.

Pointers are viable in-memory, but not on disk.

Often, these involve pointer and/or tree structures. For example, Solid has found that *tries* – a variant of tree normally thought of as a retrieval method mainly for free text – are more efficient in-memory than the b-trees that other OLTP RDMS rely on. Applix's TM1 is implemented via a tree of hypercubes that also would give questionable performance on disk. And ObjectStore's complex pointer structures can be blazingly fast when implemented in-memory, even though they face the usual challenges of object-oriented DBMS performance when implemented in a disk-centric mode.

And they don't deserve their bad rap.

In connection with this observation, we note that pointers have gotten somewhat of a bad rap in the data management world. Pointer-based data manipulation languages are indeed problematic, but pointers can be used under the covers without being reflected in the DML (Data Manipulation Language) at all. What's more, hierarchical DMLs aren't all bad. Both object-oriented programming languages and XML demonstrate that hierarchical views of data are appropriate for certain programming tasks, as long as – and this is the requirement that 1970s/1980s hierarchical systems didn't meet -- there's a well-performing tabular means of getting at the same data for future applications down the road.

Conventional DBMS also try to minimize I/O.

Memory-centric systems also benefit from going to the other extreme of the specificity spectrum. Even when they get disk reads lined up in near-perfect sequential order, disk-based DBMS have to be careful about the total amount of data they retrieve from the disk. Thus, the simple-minded query resolution

approach of a full table-scan is a very costly last resort.

That isn't necessary in memory-centric systems.

But if the data is already in memory, table scans aren't nearly as expensive. That's the key idea behind SAP's BI Accelerator, for example. It can afford to dispense with almost all indices – and shrink the overall dataset accordingly – because it can afford to do table scans on almost every query. Of course, the system tries to be as selective as possible. For one thing, it only looks at those columns relevant to a particular query. But the raw speed of a pure silicon solution allows the elimination of a lot of technical and DBA overhead associated with the sophisticated indexing used in disk-based data warehouses.

Columnar data structures have many advantages

One important group of memory-centric data structures is the *columnar* ones. Almost all relational DBMS store data in rows. Column-based data management, however, gives a natural advantage in query execution; the system only has to look at those columns which actually are relevant to the query. It also has the interesting feature that the distinction of index vs. raw data is largely obviated; a full columnar index, unlike a typical row-based set of relational indices, contains all the information in the underlying database.

But they're only practical in-memory.

The traditional drawback of columnar systems is that it's hard to update the indices without random accesses. But that's not a problem for memory-centric systems! Thus, columnar indexing is typically viable only if it's practical to keep the whole index continually in memory. In the case of text search engines, that's usually what happens, most famously in the case of Google. Indeed, the Google cache is just a reconstruction of the information in the Google columnar text index. Not coincidentally, SAP's BI Accelerator, a columnar technology, is based on technology originated in SAP's TREX text indexing functionality.

Some aspects of memory-centric technology have long pedigrees.

The columnar-storage example illustrates a key point: A lot of this memory-centric stuff isn't really new. Rather, it's a new use of old ideas. Computer science is full of algorithms that, for whatever reason, don't quite make it into important commercial product. Now some of those algorithms are finally getting industrial-strength use.

Appendix – Applix Q&A

Q. Please give a brief company overview of Applix.

A. Applix provides a complete performance management application for finance and operations. With Applix TM1, decision makers answer the hardest questions and perform what-if analysis against large data sets faster than with any other application. And TM1 is consistently ranked easiest to use and fastest to implement. Applix's 2,200 customers, including more than one-third of the Fortune 100, use TM1 for strategic planning, reporting and analytics for powering strategic analysis of financial, transactional, operational and other business data. Applix is a founder of the BPM Standards Group and has been recognized by numerous industry analyst groups for its technical leadership and vision in the marketplace, as well as high levels of customer satisfaction. TM1 applications are delivered by Applix and by a global network of partners.

Q. Please describe Applix's products in memory-centric data management and their target market(s)

A. TM1 is based on OLAP technology which enables users to quickly view and understand large sets of complex business data. TM1 does all the calculating in RAM, providing much faster response times compared to OLAP engines that store their data on a disk drive. TM1's memory-centric capability also allows users to develop complex models based on the business data to reflect various derived business performance measures including projections and forecasts of future business activities. Users can change an input value and immediately view the ripple effect throughout the model.

The target markets are analysts, managers – in all business functions -- and executives in companies of all sizes who need to perform planning, reporting, and analytics, particularly against complex business models.

Q. How has TM1 evolved over time?

A. TM1, originally developed in 1984, was acquired in 1996 by Applix. Over the years the product evolved from a single user, single threaded application in the world of PCs with 640K RAM to a multiuser, multithreaded server. In 1998 Applix released TM1 7.0, which provided improved levels of scalability and enterprise-wide deployment for its OLAP technology. In 2002, Applix TM1 8.0 increased performance, analytical functionality and manageability. In 2003, TM1 provided support for 64-bit platforms, and Applix released the first version of TM1 Web. In 2004 Applix released TM1 Planning Manager. In 2005 Applix continued to enhance TM1 with version 8.4. Dynamic views added to Excel's reporting capabilities for more sophisticated dashboarding and complex analysis. Applix also released TM1 Financial Consolidations and Financial Reporting capabilities.

Q. Where have TMI sales historically been concentrated, in terms of application areas and industry sectors?

A. In addition to planning, budgeting, forecasting, and reporting applications, TM1 handles complex business applications such as risk analysis, activity-based costing, inventory and merchandise planning,

supply & demand, customer/product/brand profitability and compensation planning. Financial services, insurance, healthcare, pharmaceutical, manufacturing, consumer goods, and telecommunications industries, among others, use TM1.

Q. How big do TMI models get?

A. From 3GB to 60 GB and beyond. Examples:

- One customer analyzes account information on 1.2 million customers, and their database is 12 GB.

- A Telco company's cube viewer shows 6 million members in a dimension and 44 GB of data.

- Another company analyzes a cube of 26 million members in a dimension and 60 GB of data.

Q. What are your supporting or companion products and functionality to the core TM1?

A. TM1 includes these capabilities or components:

- Planning – Enterprise planning to drive strategic goals

- Budgeting – Bottom-up and top-down enterprise budgeting

- Forecasting – Addresses customer demand and estimated sales revenues

- Consolidation – Rapidly streamlines data collection and reporting on financial and operational data from multiple business units, general ledgers, charts of accounts

- Web and Excel interfaces - Designed to work from within Excel

- Reporting and Analysis – Improves decision-making based on consistent, timely and accurate reports

- Extract, Transform, and Load capability – Enables a company to perform fast and precise data loads of disparate corporate data from ERP and other legacy systems (leveraging technologies such as ODBC and ODBO) as well as Excel and flat files.

Q. What is the relationship between TM1 and your application products?

A. The application products are based on the core TM1 OLAP engine capabilities.

Q. What exactly is the latest version called, and when was it released? When will the next version be available?

A. TM1, 9.0, released in December, 2005, delivers enterprise-class performance for simultaneous use by hundreds of financial analysts, line-of-business managers and senior executives for Web-based business performance management activities and reporting. Applix re-architected TM1 Web in Microsoft .NET to deliver extremely fast performance for large numbers of enterprise users.

The latest version, March, 2006, includes support for the Windows 2003 x64 platforms, providing greater physical and virtual memory for new levels of performance and scalability; and improved interoperability with SAP and SAP Business Information Warehouse (BW) data, enabling significant amounts of data to be transported rapidly from SAP BW to TM1.

The next release is scheduled for Summer, 2006.

Q. What are some typical configurations, and what do they cost?   What would be a typical hardware configuration and cost for each?

A. A typical (initial) installation of TM1 would include a server and 20-50 users. The prices are typically in the range of $100,000-$250,000.

Q. What architectural choices have you made in designing TM1 that you would not have made if it weren't memory-centric?

A. TM1 has had a calculation-on-demand approach from its inception, which necessitated a memory-centric approach. All subsequent design decisions have been based on this fundamental premise. So the short answer is "all of them".

Q. Besides being memory-centric, what are TM1's other major differentiating features?

A. TM1's memory centric approach makes possible pretty much all of its differentiating features: speed, modeling flexibility, quick response to changes, etc. The other major design features of TM1 have to do with simplicity, openness, standards, security, replication and synchronization, and:

- Modeling: A comprehensive and expressive modeling structure to handle complex business models. Users input data, pose what-ifs as many times as needed, immediately see the changes, and compare the responses with historical data or other projections. The rules language of TM1 was made practical by the in-memory model.

- Spreading: TM1's data spreading capabilities provide rapid data entry at any level of your plans, budgets or analyses. The capabilities include the ability to spread input data in proportion to other data elsewhere, such as previous year values.

Q. Why does it matter that TM1 has more flexible calculation than disk-centric MOLAP?
A. Modeling flexibility makes it possible to model the behavior of a business much closer to reality and thus reflect its performance accurately in the model. Without such flexibility, models tend to be overly simplistic and in some cases misleading.

Q. The better BI tools try to cache intelligently, of course. Where do you think they fall short of your offerings?

A. In spite of their ability to cache, most, if not all other BI products require pre-calculation in order to provide reasonable performance. Trouble is, that when a change is made, the pre-calculation has to be repeated, resulting in significant delays. TM1's approach is to calculate everything on demand, at very high speed, thus reflecting the result of changes immediately. In addition, the expressiveness of the language allows for the easy creation of very flexible and sophisticated models similar to those that would be implemented in spreadsheets (for smaller amounts of data).

Q. What future product directions would you care to disclose?
A.

- Ever increasing scalability to support larger companies with more ambitious applications.
- Integration with a broader suite of complementary products.

Q. Which other products do you compete against most often on customers' short lists? When you lose, why do you lose?
A. Cognos Planning, Hyperion Business Performance Management Suite. TM1 is not chosen when the prospect company makes its choice based upon branding hype and/or chooses not to engage in a proof of concept.

Appendix – Intel Q&A

Q. Please give a brief company overview of Intel.
A. For decades, Intel Corporation has developed technology that has enabled the computer and Internet revolution that has changed the world. Founded in 1968 to build semiconductor memory products, Intel introduced the world's first microprocessor in 1971. Today, Intel, one of the world's largest chip makers, is also a leading manufacturer of computer, networking, and communications products.

In partnership with the world's largest and most robust community of hardware manufacturers and software developers, Intel delivers innovative, highly compatible server platforms and software tools that help customers grow and compete with confidence. Intel's approach of providing balanced, complementary technology components ensures that customers get optimal performance, availability, manageability, and efficiency in their server infrastructure.

Since 1994, Intel and SAP have collaborated to lower total cost of ownership for their customers. As evidenced by over 74% of new SAP solution installations being deployed on Intel platforms, Intel and SAP deliver optimized and integrated solutions, from servers to mobile clients, which reduce risk and enable business growth. For additional information, please see www.intel.com

Q. Please describe your participation in the memory-centric data management market.
A. Intel collaborated with SAP in the development of the BI Accelerator (BIA) by using advanced Intel software and hardware technology optimizations to maximize performance and scalability in solutions based on enterprise services architecture on innovative 64-bit Intel® Xeon processor-based platforms. 64-bit and multi-core pre-configured systems powered by Intel® Xeon processors allow blade server configurations required for plug & play "appliance" delivery, available from OEMs such as HP and IBM, with very low administrative overhead.

More generally, Intel is the preferred platform for demanding enterprise applications, including those for memory-centric DBMS.

Q. What do you do in your hardware platforms that is particularly helpful to memory-centric processing? How do you expect these capabilities to evolve?
A. Memory-centric processing requires these continuing evolutions in hardware platform technologies:

1) Increases in the amount of memory available to processing threads. Support for vastly greater physical memory and virtual memory space enables new scenarios not possible before. Intel server platforms are now EM64T-enabled (http://www.intel.com/technology/64bitextensions/), allowing access to 64-bit virtual memory footprints. Going forward, all Intel platforms will support, larger, faster, cheaper memory (for instance, 4GB DIMMs, with 667MHz etc.), and add additional address lines to the platform to support even larger physical memory.

Moving to a 64-bit architecture increases the amount of virtual memory that can be directly addressed from 4 GB to 16 Exabytes. Of course, current applications don't require that large an address space, and so OS support and hardware support generally limit the virtual address space of actual OS processes to

~16TB (Terabytes). For instance, one limit is currently the size of addresses that pass from the CPU to the memory subsystem, and the OS can assume that limit to simplify/optimize some of its internal page translation routines. Currently, this limit is 40 bits in most high-volume Intel servers, allowing direct access to 1TB of RAM, with plans to increase this to at least 48 bits of address space in future platforms.

Beyond physical memory address space, however, is the very real increase recently in actual RAM that can be physically supported on a platform. It is not unusual for current servers to now have 32Gb - 64Gb of RAM. The ability to connect high-speed RAM into a single platform is limited not only by designed-in address lines limits, but also by physical factors such as electrical fan out, capacitance, clocking delays, etc. Intel is making continual improvements in these areas, including the use of Fully Buffered DIMMS.

2) Increases in the amount of data which can transfer from memory to the processors (memory bandwidth). When memory size increases, additional data must be transferred from memory to the processing engines. For instance, in current Intel platforms for the BI Accelerator, during heavy query processing, we achieve near our theoretical bus bandwidth. Platforms available from Intel in late 2006 for the BIA will more than double this bandwidth (10.66 Gb/sec).

3) Keeping the time to access memory (latency) low, even with more and more memory modules. A key Intel approach is to put large memory caches in the processor, thus limiting the need to access slower memory. For instance, an access to memory might take 100ns, while an access to a processor cache takes as little as 2 clock cycles (for a 2.0GHz processor, this would be an access time of 1ns, or .01X the access time to memory).

In addition, to improve hit rates in this cache, Intel implements advanced prediction algorithms to "pre-fetch" data into the cache before it is actually called for by the processor. On some memory-bound workloads, this can increase performance as much as 10-20%.

Additional improvements going forward include the ability to add more memory and access bandwidth to platforms while keeping latency in control using Fully Buffered DIMMS. (Please see www.intel.com/technology/magazine/computing/fully-buffered-dimm-0305.pdf for more details.)

4) Increasing the amount of thread-level parallelism. Intel's approach is to continue to apply advanced out-of-order processing in the CPU. With our next generation server CPU architecture in 2006, we expect the CPI (Cycles per Instruction) to decrease by ½ on BIA code. In addition, for the BIA advanced compiler techniques are being examined which in the past have given double digit performance improvements by providing better software pre-scheduling for the hardware.

5) Increasing the number of useful hardware threads that can be applied to the same data in memory. Currently, the BIA is delivered on single-core processors. By the end of the year, dual-core products will be available for the BIA. After that, platforms with more and more cores will become available. Intel is working directly with SAP development teams to ensure that BIA will effectively utilize these additional cores.

6) As single-platform limits are reached (processing power, memory, memory bandwidth, etc.), Intel high volume platforms directly allow cost effective scale-out approaches using low cost components (e.g., Intel-based blades). BI Accelerator was conceived with this scale-out design in mind so customers can quickly

add more capability as their data and query loads increase. During design and testing of the BI Accelerator product, SAP and Intel focused heavily on testing this "blade scaling" concept. We used real user workloads to prove that adding more blades into the configuration not only gives BIA the ability to scale with data size, but also cost-effectively give better query performance.

To summarize, Intel is quickly bringing additional improvements in these important platform technologies to market. As an example, to quickly compare the current Blades used for BIA to those that will be available in 2H of 2006:

Item	Current Platform	2H 2006 Platform
Platform	Nacona	Woodcrest+
Cache	1Mb-2Mb	2Mb+
Bus	800 MHz	1066 MHz
Memory	8Gb (DDR2 400Mhz)	16Gb (FBD DDR2-667MHz)
Performance for BIA	1x	> 2x

Q. What do you believe are the most important metrics that determine a chip/hardware platform's ability to support memory-centric data management efficiently?
A. As with most real computing tasks, there is no single metric that determines hardware platform effectiveness for memory-centric data management. All of the key items mentioned above (memory size, latency, bandwidth, processor micro architecture, processor speed, number of cores, etc.) must be considered.

Rather, it is the delicate balance among these competing design factors, including such "mundane" items like cost, market availability, platform form factor, platform power, platform manageability, etc. that make a hardware solution attractive for solving a real customer's business problem.. Intel's long history of innovation and research has led us to understand these issues and offer platforms which contain the proper right balance among these factors.

Q. What is the role of Intel with regards to the SAP BI accelerator?
A. Intel is collaborating with SAP to drive development and scaling of the Enterprise Services Architecture (ESA) and vision. ESA is SAP's implementation of a Service Oriented Architecture (SOA). SAP NetWeaver with the BI accelerator is a key enabler of a product that conforms to ESA. Intel has been working toward a vision of the Service-Oriented Enterprise (SOE) for some time, which extends far beyond simply making better use of data-center resources. It provides guidance for addressing many of the key challenges currently faced by enterprise architects, such as traversing firewalls, integrating third-party networks, and enabling mobile workers, while minimizing costs.

Q. Why does Intel believe it is uniquely positioned to address customers' issues in the context of this functionality?
A. Intel has innovative technology coupled with the insight into how companies are implementing ESA to help customers address the tough challenges of moving to these environments. Intel's deep technology expertise and advanced Intel software development tools maximize the performance and scalability of this technology. Intel also offers a broad range of open, flexible platforms, from 64-bit to multi-core and with

features for greater security, manageability and reliability, via the industry's largest ecosystem of platform providers. Intel platforms help improve performance and scalability and lower total cost of ownership. Currently, more than 74% of all new SAP installations are deployed on Intel platforms.

Q. What does Intel offer to solve IT challenges?
A. Intel-based servers help businesses worldwide reduce their risks and operate with great productivity at low cost. This is because Intel-based servers have a 20-year proven track record in delivering outstanding reliability and performance across the broadest choice of hardware, operating systems and applications. Large and small businesses alike depend on Intel-based servers. 8 out of every 10 servers shipping today are based on Intel architecture.

Intel also helps business solve their toughest IT challenges. In addition to increasing server processor frequencies and bus speeds for better performance, Intel delivers innovation beyond the CPU in chipsets, standards, storage, I/O, memory, server management, power management, and CPU virtualization. These innovations extend the value and capability of customers' servers.

Appendix – Progress Software (Real Time Division) Q&A

Q. Please give a brief company overview of Progress Software.
A. Progress Software Corporation (NASDAQ: PRGS) is a global industry leader providing application infrastructure software for all aspects of the development, deployment, integration and management of business applications through its operating units: Progress OpenEdge Division, Sonic Software, DataDirect Technologies, and Progress Real Time Division.

Users of information technology today demand software applications that are responsive, comprehensive, reliable and cost-effective. Progress Software products address their needs by:

- Boosting application developer productivity, reducing time to application deployment, and accelerating the realization of business benefits.
- Enabling highly distributed deployment of responsive applications across internal networks, the Internet, and disconnected users.
- Simplifying the connectivity and integration of applications and data across the enterprise and between enterprises.

Q. Please describe your products in memory-centric data management and their target market(s).
A. Progress Software's products accelerate the performance of existing databases through sophisticated data caching; manage and process complex data using the industry's leading object database; and support occasionally connected users requiring real-time access to enterprise applications through data replication and synchronization. Progress products also provide the ability to manage and analyze real-time event stream data for applications such as algorithmic trading and RFID.

Progress ObjectStore® is proven technology for developing high-performance object database management (ODBM) environments. It is an object database that connects to existing enterprise and relational database systems and then caches data to deliver performance to users at in-memory speed. Progress Software's DataXtend product line efficiently delivers data to distributed applications. Progress® DataXtend® CE (Cache Engine) provides a distributed, persistent data infrastructure for applications with real-time requirements. DataXtend CE tools offer a choice of model-driven or schema-driven development and expand an organization's ability to support high-speed concurrent access to relational data in whatever format is required by the application logic.

Progress® Apama® supports the processing of multiple streams of event data with the goal of identifying the meaningful events within those streams. Traditional software environments have been forced to respond to the world "after the fact" -- after events have happened. Apama provides event processing platform that allows businesses to monitor events as they happen, analyze the events to identify salient patterns, and act -in milliseconds.

Q. How have ObjectStore, the DataXtend product family, and Apama evolved over time?
A. Progress Software acquired the ObjectStore object database product in 2002 through the acquisition of eXcelon and has continually released upgrades based on market demand.

Through the acquisitions of Persistence Software in 2004 (which offered a product called EdgeXtend) and PeerDirect (an existing division of Progress Software), as well as some internal development, Progress launched the DataXtend product line in October 2005.

ObjectStore's flexibility provided an excellent basis as a high performance storage engine for Event Stream Processing applications, but we saw an opportunity to provide a higher level infrastructure including a programming language specifically designed for this need as well as a set of development tools to provide a more complete environment. This led to the acquisition of Apama in April of 2005 which, combined with our ObjectStore-based EventStore technology, is the basis of Progress Software's Event Stream Processing (ESP) platform.

Q. Where have your sales historically been concentrated, in terms of application areas, industry sectors, and metrics of both size and complexity?
A. Progress Software's memory-centric products are used across a range of industries and applications; however, there is high adoption in areas where high performance is critical. One example of this would be in telecommunications, where the technologies are used for applications such as intelligent networking, call routing and switching, and fraud detection. Another example would be in financial services, in particular capital markets, where performance improvements in the range of milliseconds can make millions of dollars of difference for applications such as algorithmic trading or risk management. Other areas include military applications, intelligence, energy trading and surveillance.

The overall data set size managed in these systems can be extremely large. Systems are in deployment which manage terabytes of data, and it is not uncommon for applications to be managing millions of objects in the store.

Q. What exactly are the latest versions called, and when were they released?
A. ObjectStore v6.3 was released in October 2005. DataXtend CE v3.1/v9.1 and DataXtend RE v7.5 were released in October 2005. Apama v2.4 was released in February 2006.

Q. What are typical deployment configurations for Progress Software memory-centric data management products?
A. The Progress products are deployed in a wide range of configurations, from embedded devices such as copiers and network switching gear, to farms of Linux boxes supporting high performance web sites, to very high end Solaris or HP-UX systems in data centers.

Q. What makes ObjectStore and the DataXtend products particularly suitable for memory-centric applications?
A. The ObjectStore Cache-Forward Architecture® delivers multiple benefits that make it ideal for memory-centric systems. First, its virtual memory mapping provides an application with the ability to model anything in the database that can be modeled in physical memory, providing enormous flexibility for optimization in applications that can define data structures that best represent the information that they are processing while the storage management engine delivers the transactional guarantees that a traditional enterprise class database management system would be expected to provide. Second, the system provides distributed cache coherency across all instances of the application. This is essentially transactional distributed shared memory that can work in a heterogeneous network of varied hardware and operating

systems while ensuring that all copies of the data being used in these environments are consistent and up to date.

The DataXtend product line delivers another key capability in providing a bridge between these memory-centric environments and the enterprise systems that often remain as the "single source of truth" for the data being manipulated. The mapping capabilities allow the application to choose an appropriate in-memory representation of the data which may be quite different than the source data that resides in the enterprise information systems. At the same time, the synchronization features of this product ensure that these heterogeneous views of the data are kept up to date and in sync with the systems of record.

A memory-centric approach is, in fact, required for Progress Apama, the Event Stream Processing (ESP) product. In this environment, the incoming data streams are constantly being analyzed with respect to the queries or scenarios that have been set up by the users of the system. These incoming events are processed by this engine before they even reach disk to ensure absolute minimum latency in their processing. In systems such as algorithmic trading, RFID tracking or network surveillance there may be tens of thousands of scenarios in play at any given time and the data may be streaming at a rate of tens of thousands of events per second. Without a memory-centric approach, it would be impossible to achieve these volumes.

Q. What are the products' major differentiating features?
A.
- ObjectStore: Cache Forward Architecture as described above.
- DataXtend: model-driven tools for high developer productivity.
- Apama: EventStore (which leverages ObjectStore technology) which captures tens of thousands of events per second to disk as well as processing them.

Q. What architectural choices have you made in designing your memory-centric products that you would not have made if they weren't memory-centric?
A. For ObjectStore, we would have focused much more on server side buffering and query optimizations. For DataXtend we would have used a more traditional "request response" model for populating the cache vs. a push-based approach that pre-populates data from the data sources to capitalize on memory available to the application. Apama's Event Store uses "vector-like" data structures instead of "B-Tree like" data structures optimized for disk paging.

Q. What future product directions would you care to disclose?
A. Progress is working on combining the strengths of ObjectStore, DataXtend CE and DataXtend RE into a single architectural stack to provision data from existing data sources into the applications that need it. For Apama and Event Stream Processing: improved dashboard/visualization capabilities and better integration of EventStore with the core event manager.

Q. Which other products do you compete against most often on customers' short lists? When you lose, why do you lose?
A. ObjectStore competes with other object-oriented DBMS including: Versant, Objectivity, and Gemstone. DataXtend CE often competes against a home-grown approach combining point solutions for caching, such as Tangosol, along with a mapping tool such as Hibernate. Apama competes with Streambase, ISpheres, and Tibco's Business Events.

Appendix – SAP Q&A

Q. Please give a brief company overview of SAP.
A. SAP is the world's leading provider of business software solutions. Today, more than 24,450 customers in over 120 countries run more than 84,000 installations of SAP® software -- from distinct solutions addressing the needs of small and midsize businesses to enterprise-scale suite solutions for global organizations. Powered by the SAP NetWeaver™ platform to drive innovation and enable business change, mySAP™ Business Suite solutions are helping enterprises around the world improve customer relationships, enhance partner collaboration and create efficiencies across their supply chains and business operations. SAP industry solutions support the unique business processes of more than 25 industry segments, including high tech, retail, public sector and financial services. With subsidiaries in more than 50 countries, the company is listed on several exchanges, including the Frankfurt stock exchange and NYSE, under the symbol "SAP." (Additional information at http://www.sap.com)

Q. Please describe your products in memory-centric data management and their target market(s).
A. By your definition, SAP's primary memory-centric data management product is BI accelerator, a functionality of SAP NetWeaver Business Intelligence (SAP NetWeaver BI). Principal target markets of BI accelerator are SAP NetWeaver deployments for analytics, especially in the following segments:

- Analytics with high levels of granular data (such as for example in verticals such as retail, telco, consumer packaged goods, or utilities)
- Analytics in industries and environments experiencing frequent change and innovation (hence requiring utmost flexibility).

Over time we expect to see strong adoption of BI accelerator in other market segments for SAP NetWeaver.

Also possessing noteworthy memory-centric aspects is our LiveCache technology, used for example for demand and supply planning in mySAP SCM (supply chain management),

Q. What are BI Accelerator's antecedents?
A. SAP NetWeaver has been shipping since several years now as one technology platform. SAP NetWeaver BI technology had been around in its core already since the 90's, although at that stage still as a separate dedicated product offering.

BI accelerator is a fairly recent functionality offering, which was added with the latest release of SAP NetWeaver in 2005. It has experienced strong adoption and demand since then.

BI accelerator uses search engine technology ('TREX') from SAP NetWeaver, to perform in-memory index scans. TREX was introduced originally for knowledge management, and continuously has been enhanced since then,

Q. To whom do you sell NetWeaver BI (of which BI Accelerator is a part)?
A. SAP NetWeaver BI technology and its adoption have evolved significantly over the last couple of years. While in the 90's there had been a strong focus on traditional reporting and dashboards, we now see a major trend towards process-embedded analytics and real-time business intelligence. Also, the scope of implementations has changed. Thousands of customers are using SAP NetWeaver today as their primary and strategic, enterprise-wide platform for all BI and analytics needs, often integrating significant number of non-SAP systems as well. In total, there were over 11,000 implementation of SAP NetWeaver as of 12/05.

Q. What exactly is the latest version called, and when was it released?
A. The BI accelerator is available with the latest version of SAP NetWeaver, called SAP NetWeaver 2004s. It ships and installs as an appliance, running on Intel processors on a choice of either IBM or HP blade systems.

Q. What are typical deployment configurations for BI Accelerator?
A. BI accelerator runs on a choice of IBM or HP hardware, equipped with Intel processors. Specifically, the supported hardware configurations are: IBM BladeCenter with IBM TotalStorage, and HP BladeSystem with HP StorageWorks. There are multiple appliance 'sizes' available, from 'small' boxes to 'very large' boxes. Those appliance boxes are shipped preconfigured with Linux operating systems, as well as the respective SAP NetWeaver BI accelerator software. Installation is very straightforward: plug the appliance into the network, then check off the information cubes in SAP NetWeaver BI to accelerate. It's as simple as that.

Q. Memory-centricity aside, what are BI Accelerator's other major differentiating features and benefits?
A. BI accelerator has been built specifically for SAP NetWeaver BI, hence is fully optimized for (and integrated with) SAP NetWeaver BI. This results in great advantages over other 'bolt-on' or hub-and-spoke scenarios. Data does not have to be moved between different systems, which greatly streamlines processing and fosters data integration and data quality. This also helped in the overall system optimization, in maximizing performance and compression. Not to mention the additional benefits of dealing with one technology platform, and potentially just one vendor.

Q. What architectural choices have you made in designing your memory-centric products that you would not have made if they weren't memory-centric?
A. For starters, BI accelerator is taking full advantage of Intel chips' first and second level cache. In fact, that is an area where SAP collaborated with Intel, to fully exploit the benefits of the advanced chip design provided. This allows BI accelerator to do full index scans, with a very fast and reliable performance. Disk-based solutions just cannot do that, even if cached.

Q. How do you feel BI Accelerator is superior to other BI vendors' caching technologies?
A. As mentioned, BI accelerator is much more than caching technology. It does not pre-cache any data results, but calculates results 'on the fly', for each query request. BI accelerator can do that, because the code is so efficient, and highly optimized for data structured of SAP NetWeaver. The advantages of this approach over caching are apparent: utmost flexibility, and steep scalability (scalability is due to highly compressed in-memory data structures).

Q. What future product directions would you care to disclose?
A. Over time, additional advanced analytical processing capabilities will be moved into the analytic engine inherent to the BI accelerator functionality,

Q. What is the relationship between SAP Analytics, SAP NetWeaver, and the BI Accelerator? And do you view SAP today as an application company or a technology company?
A. SAP is both -- a technology and an applications company. With SAP NetWeaver, customers receive a leading technology platform to not only run off-the-shelf applications, but also to compose and deploy customer applications. Among other functions, that platform serves as a BI platform, provisioning information in the context of process automation. With SAP xApps Analytics, powered by SAP NetWeaver, customers receive instantly usable composite analytic applications, to effectively manage business performance, on a strategic, tactical and operational level. The BI accelerator, as a functionality of SAP NetWeaver, assures effectively instantaneous response times at maximum scalability -- whether in conjunction with ad-hoc analysis, with planning and budgeting, or with process-embedded, guided analytics.

Appendix – Solid Information Technology Q&A

Q. Please give a brief company overview of Solid Information Technology.
A. Solid Information Technology is a provider of database management solutions. Solid is the only company that provides a fast, always-on database. This affordable and compact solution is easy to deploy, use and maintain. More than 3,000,000 Solid databases are deployed worldwide in real-time applications and communications networks. Customers include market leaders such as Alcatel, Cisco, EMC2, HP, NEC, Nokia and Siemens. With Solid, companies experience faster time to results and dramatically lower total cost of ownership. Solid Information Technology has worldwide headquarters in Cupertino, California, and regional offices in North America, Europe and Asia.

Q. Please describe Solid's products in memory-centric data management and their target market(s).
Solid BoostEngine was designed especially for high performance data access to satisfy real-time demands. It enables Network Equipment Providers and ISVs to embed a robust, fully-featured, and compact in-memory database directly into their solution to provide extremely fast performance. Additionally, because all aspects of the database can be controlled and fine-tuned within the application, no DBA is required. Solid BoostEngine also has the unique ability to deploy in conjunction with Solid's disk-based database engine. Deployments that simultaneously utilize this hybrid mode are able to optimize price/performance to adapt to dynamic workloads and system resource availability, from within a single interface and transparent to the application.

Solid has hundreds of customer worldwide in a variety of industries. Over 3,000,000 instances of Solid's database management solution are deployed in highly demanding production environments, where IT support is not available and 99.999% uptime is required. Deployments include trading rooms, telecommunication networks, multimedia network printers, medical devices, fleet management systems and Point of Sale (POS) solutions.

Q. What are BoostEngine's antecedents?
A. In 1994, Solid released EmbeddedEngine, a fully-featured relational database management system for disk-based storage. In 1997, Solid developed the HotStandBy Option which offered the ability to deploy an active/standby database engine instance paired with automatic failover capability to achieve high availability. A year later, Solid delivered SynchroNet, a database replication option for distributing data among servers. In 1999, Solid implemented Accelerator, a feature that enabled customers to link their application directly to the database engine library for increased performance and embedded deployment. In 2003, Solid expanded the database by providing a hybrid on-disk/in-memory database engine. This edition was called Solid BoostEngine. While providing an in-memory database engine in addition to the existing traditional disk-based database engine, it retained all capabilities of Solid EmbeddedEngine, including the Accelerator which provided linked library deployment, and the ability to startup from diskless devices.

Q. Where have BoostEngine sales historically been concentrated?
A. Deployments include trading rooms, communication networks, multi-functional printers, medical devices, fleet management systems and Point of Sale (POS) solutions. The majority of deployments are

within the networking and communications industries. These deployments include the data management of network configurations, provisioning, media gateways, performance, device states, subscriber data, as well as online charging and billing data with volumes ranging from a few gigabytes to the hundreds of gigabytes. Deployments are on different hardware configurations running real-time operating systems, Linux, different versions of UNIX, and Microsoft Windows on blade servers, SMP servers, ATCA architectures, and multi-node configurations.

Q. What are your supporting or companion products to BoostEngine?
A. In addition to BoostEngine and EmbeddedEngine described previously, Solid provides two options. Solid's CarrierGrade option provides the ability to deploy an active/standby database instance pair with automatic failover capability to achieve 99.9999% high availability that is needed in carrier-grade environments. Solid's SmartFlow option provides bi-directional, publish and subscribe replication capabilities with guaranteed data consistency, fine-grained security control, central data integrity management and optimized bandwidth.

Q. What exactly is the latest version called, and when was it released? When will the next version be available?
A. The current version is Solid v4.5. It was released in May 2005. The next major release will be available during the first half of 2006.

Q. What are some typical configurations, and what do they cost?   What would be a typical hardware configuration and cost for each?
A. Deployment configurations vary a great degree, ranging from embedded deployments in solutions that cost from hundreds of dollars to complex deployment topologies of several servers in the range of millions of dollars. There are simple configurations where a single instance of the database engine runs on a device such as a high end network printer. There are also deployments on ATCA architectures where over a dozen instances of Solid BoostEngine run on different blades as well as separate servers hosting Element Management Systems that communicate with the Solid instances on blades using Solid SmartFlow.

Q. What are BoostEngine's other major differentiating features?
A. As demonstrated by the Telecom One benchmark, Solid BoostEngine is an extremely fast in-memory database engine. It is also the only solution on the market that can be deployed simultaneously with a disk-based engine, with each providing the same levels of replication and high-availability under a single storage manager. Solid BoostEngine also features temporary and transient tables, and dynamically controllable durability for relaxed logging. With the Carrier-grade option offering failover as fast as 30 milliseconds, Solid's BoostEngine is the only in-memory, carrier-grade solution available in the market.

Q. What architectural choices have you made in designing BoostEngine that you would not have made if it weren't memory-centric?
A.
- New storage model replacing the fixed-size page oriented structures that had been inducing overhead of reference indirection and high maintenance cost. New structures are typically tree-based and thus very flexible and efficient.

- New access methods that optimize CPU usage in place of disk accesses.

- New checkpointing methods to make the core in-memory engine immune to the delays of the I/O system.

- New transaction logging system to minimize the worst bottleneck of any database system, namely the log writing. This includes maximum use of I/O asynchrony whenever possible.

- New cost model for query optimization: the traditional disk arm movement optimization is replaced with the optimization of CPU cycles.

Q. Which other products do you compete against most often on customers' short lists? When you lose, why do you lose?
A. Solid most often competes against "the incumbent database vendor". Most often the incumbent vendor is Oracle. In cases where the customer does not need an in-memory database approach, and at the same time true carrier grade high availability, i.e. 99.9999% uptime, is not a requirement, then Oracle might be just good enough and displacing it might be a political challenge.