Index-Light MPP Data Warehousing

A Monash Information Services Bulletin

by

Curt A. Monash, Ph.D.

March, 2007

Sponsored by:

Abstract

Different DBMS are best at different tasks.

A single relational database management system (RDBMS) can perform a broad variety of duties. It may even do them all pretty well. But for some uses, a special-purpose product can greatly outperform general-purpose systems. Complex data warehousing is such a task.

Index-light MPP appliances *excel at data warehousing.*

For most data warehouses, market-leading general-purpose RDBMS are good enough. But for complex queries against multi-terabyte data warehouses, *index-light MPP data warehouse appliances* are a much more efficient option. Offered by DATAllegro, Netezza, Teradata (if you use the term "appliance" a bit loosely), and IBM (if you use the term "appliance" *very* loosely), these systems beat their *index-heavy SMP* counterparts on several major criteria:

- Performance
- Price/performance
- Consistency of performance
- Administration costs

Much of this superiority stems from three factors.

The index-light MPP (Massively Parallel Processing) appliance story hinges on three technical factors:

1. *Shared-nothing MPP*. Loosely-coupled systems are significantly cheaper than tightly-coupled ones, for the same level of raw component performance.
2. *Reduced use of indices.* By minimizing redundant references to information, index-light systems can store up to 7X less data than index-heavy ones. This produces enormous savings both in hardware and in administrative costs.
3. *Avoidance of random disk reads.* Disk rotation speeds have only improved 12.5-fold in the past 50 years, making random disk lookup the greatest constraint on conventional RDBMS performance. Index-light systems largely evade this bottleneck.

DATAllegro offers a prime example.

DATAllegro offers what may be the archetype of the index-light MPP appliance strategy. A typical system contains multiple standard servers, each responsible for twelve standard disk drives, for a total installation in the tens of terabytes. (Indeed, as of DATAllegro V3, the servers and storage units are just standard Dell and EMC products respectively.) Data generally comes off the disks in full table or partition scans, in 24-megabyte blocks, but you can use the functionality of Ingres if you want to. And the whole thing is a lot faster and cheaper than conventional index-heavy alternatives.

Index-light MPP data warehousing

Oracle and Microsoft have similar data warehouse strategies.

Oracle and Microsoft took similar approaches to data warehousing: Start with solid OLTP database managers, and add in a bunch of features to accelerate complex queries. The most important of these features are special-purpose index and data access options. Stars/snowflakes, materialized views, cubes – you name it, and one (in most cases both) of those vendors offers it. The basic idea of these various tactics is usually similar – make certain assumptions about the queries that will be run, and accelerate their execution by precomputing some of the steps in advance.* We call this classical approach *index-heavy SMP*, since it is generally pursued on tightly-coupled "shared-everything" SMP (Symmetric Multi-Processing) platforms.

**Bitmaps/column indices are something of an exception to this generalization, as are geospatial and full-text indices.*

Teradata, IBM, DATAllegro, and Netezza favor a different approach.

While the Oracle/Microsoft approach suffices for most data warehouses, a rival strategy has had great success at the high end of the market: *index-light MPP/appliance*. Its key elements include:

- Dedicated "appliances" rather than general-purpose computers.*
- "Shared-nothing" MPP (Massively Parallel Processing) rather than "shared-everything" SMP.
- Limited use of complex indexing, relying instead on the raw speed in executing basic functionality.

Teradata is the long-time standard-bearer for this approach, but in recent years has gotten a lot of company. Upstarts DATAllegro and Netezza follow a purer form of the strategy than Teradata does, and IBM is moving ever more toward an index-light MPP appliance approach as well.

**Reasonable people can disagree as to what really does or doesn't constitute a computing* appliance. *We take a rather expansive view of the term – if something is a single-purpose computer with pre-installed software, we're inclined to call it an "appliance."*

Index-light MPP appliances have multiple advantages:

The index-light MPP appliance approach to data warehousing has some compelling advantages over the OLTP-plus strategy. These include:

Cheaper hardware, …

- *Cheaper hardware.* Integrated hardware is expensive to scale. So if one can divide a job among N modules, that's usually much cheaper than using one tightly integrated system approximately N times as powerful.

… smaller database sizes, …

- *Smaller databases.* Indices consume lots of disk space, sometimes 6-10 times as much as the raw data itself. This is a huge advantage for the index-light approach.

… less overhead, …

- *Less overhead.* Not only do indices have to be stored on disk, they have to retrieved, maintained, and so on. While the purpose of indices is to reduce total processing, too often they have the opposite effect.

… lower administrative costs, …

- *Less administration.* Indices don't just make work for computers. They also make work for people. A large fraction of the DBA (DataBase Administrator) workload consists of managing the complex indices needed for analytical queries. Oracle, Microsoft, and for that matter IBM make huge efforts to offer ever-better automation. Even so, conventional data warehouses are a full-employment program for expensive DBAs.

… more consistent response times, …

- *Consistent response times.* In conventional index-heavy data warehouses, the performance of a query depends greatly on whether the appropriate special index happens to have already been built to accelerate it. In index-light MPP appliances, performance is more even.

… and better actual performance.

- *Better performance.* And those consistent responses are fast. MPP appliances commonly outperform conventional warehouses even on queries the latter are carefully tuned for, and blow them away on others. What's more, this performance comes at much lower total cost of ownership.

Shared-nothing MPP

Parallel processing is inherently more cost-effective.

There are two ways to make more powerful computers:

1. Use more powerful parts – processors, disk drives, etc.
2. Just use more parts of the same power.

Of the two, the more-parts strategy is much more cost-effective. Smaller* parts are much more economical, since the bigger the part, the harder and more costly it is to avoid defects, in manufacturing and initial design alike. Consequently, all high-end computers rely on some kind of parallel processing.

**As measured in terms of capacity, transistor count, etc., not physical size.*

There are two main kinds of parallel processing.

There are two main kinds of parallel processing: *Shared-everything* and *shared-nothing*. In shared-everything systems, multiple processors address a common pool of memory – RAM and disk alike. In shared-nothing systems, there is a much looser coupling of components, which each processor controlling its own RAM and disk as it would in a stand-alone computer. While the two terms are not wholly equivalent, as a practical matter shared-everything systems are typically also SMP (Symmetric Multi-Processing), and SMP machines are typically shared-everything. Similarly, shared-nothing systems are inherently MPP (Massively Parallel Processing), while MPP systems are usually shared-nothing.

Shared-everything SMP doesn't scale well.

When parallel processing became common in the 1990s, shared-everything SMP won out over MPP, for one compelling reason – existing software didn't need to be rewritten. However, SMP has major problems with scalability, in at least two ways. One is a general problem: As each processor keeps track of what the others are doing, SMP overhead increases exponentially with the number of processors. Another is more database-specific: Shared-everything storage bandwidth has trouble keeping up with the data flows that dozens or hundreds of processors demand. Consequently, MPP always played a role in high-end data warehousing, primarily via Teradata.

Shared-nothing MPP data warehousing is well-established.

By now, MPP has gained footholds in various areas of high-end business computing, commonly referred to by names such as "grid," "virtualization," or just "cluster." Its greatest success – research/scientific uses perhaps aside – continues to come in the area of complex data warehousing. Looking at market share, two of the top four data warehouse software providers favor an MPP approach (Teradata and IBM, with the others being Oracle and Microsoft). And if one expands the list to include top technology contenders with lower market shares, MPP providers still account for half or so of the names.

Common MPP design elements include:

Index-light MPP data warehouse appliance (or software) products reflect a variety of design choices and feature sets. But as one examines the various offerings, certain themes keep recurring:

Hash partitioning, …

- *Hash partitioning.* A *hash* is a function that takes a data value and calculates an address or key, almost uniquely (100% uniqueness is usually neither feasible nor necessary). In *hash partitioning*, a hash is used to spread data evenly across MPP nodes. Thus, the work of retrieving data is also typically spread evenly among the nodes, for maximum performance. In DATAllegro systems, data is almost always hash partitioned.

… heavy use of

- *Hash joins.* One of the best ways to join two tables in a relational

hash joins, …

database is to hash on the join keys in each of them and compare values. When the data happens to be pre-hashed, these *hash joins* are even more efficient. If hash partition keys are well chosen, this happy circumstance can occur a significant fraction of the time. In DATAllegro's systems, hash is the join algorithm of choice.

… selective use of indexing, …

- *Limited indexing.* Indices serve two main functions in relational databases – they tell you where to find particular pieces of data, and they precalculate some of the intermediate results needed for certain table joins. Limited-index MPP appliances willingly forgo most of these advantages. Rather than slowly finding exactly the right data, they read larger amounts of data extremely quickly.

… and fast inter-node transport.

- *Fast node-to-node data transport.* MPP data warehouses require moving a lot of data from disk to processor, and then among various processing nodes. As a result, even MPP providers that otherwise use fairly standard hardware and software underpinnings commonly do something "extra" to speed up this transport. DATAllegro, for example, makes aggressive use of Infiniband, currently via Cisco boxes.

Limiting Database Expansion

RDBMS usually rely on indices to find rows.

Traditional relational database managers store data in rows. For each table, they maintain indices on one or more columns or column combinations – i.e., *keys*. For each value of the key, the index stores a list of rows in which that value can be found. More precisely, it will commonly store the address of a block of data in which the specific desired rows are located.

Complex indexing leads to database expansion.

If you index on every column, you in effect reproduce all the information in a database, plus you store row/block addresses over and over again. Naively, therefore, one might think that the most aggressive possible index would increase database size by a factor of 2-3X over what's needed just to store the raw data itself. But it gets worse than that. For example, precalculated aggregates can defeat sparsity compression. And precalculated joins can require the maintenance of views that are larger than the underlying tables themselves. As a result, 6-9X factors of database expansion are not unusual, and more than 10X is not unheard of. And if you get into non-relational MOLAP (Multi-Dimensional OnLine Analytic Processing) systems – something we generally do not recommend -- expansion can be much worse yet.

Expansion causes storage and

The most obvious cost of expansion is disk – if you have more data, you have to pay for platters to store it. But there are human costs as well. All

administration costs.

those indices have to be created and maintained. Two decades after the successful commercialization of RDBMS, tuning them is still a hit-or-miss proposition. Even if you have state-of-the-art toolsets, managing a conventional data warehouse is a highly labor-intensive operation.

Index-free data warehouses are now realistic.

Increasingly, it is turning out that those expensive indices aren't necessary after all!* In some cases, such as most DATAllegro installations, tables are stored with no index whatsoever. This is not as outlandish as it may first sound. When a table is used in a join, it is common to read the whole thing into memory anyway. Range partitioning can also play a lot of the indices' traditional role in expediting data retrieval. Nonetheless, index-free strategies are pursued mainly on MPP data warehouse appliances carefully designed for super-fast *table scans.*

**Why that's happening* now *is explained in the next section.*

In other cases, lightweight indexing can suffice.

That said – while index-free strategies work for some applications, in others indices are needed no matter who your vendor is. Some data warehouse applications, for example, follow up complex queries with simple transactions – and if you're doing transactions, generally it really is best to have a path directly to an individual record. Fortunately, the majority of MPP data warehouse appliance vendors offer full DBMS capabilities. DATAllegro, for example, incorporates the RDBMS Ingres, which is used for many demanding transactional applications by customers such as the New York Stock Exchange.

Sequential access

Most aspects of computer hardware improve 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 rotation speed is a huge 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.

Disk access dominates RDBMS response times.

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), which is thus the absolute floor on average disk access times; 5-6 ms is a more realistic figure for the fastest disks, ranging up to 15 ms for cheaper ones. Even the low end is about a million times longer than raw RAM seek times, which have declined to just a few nanoseconds. Therefore, nothing that happens in silicon is nearly as important to DBMS performance as the raw speed of getting data on and off of disk.

Random disk access can be painfully slow.

Traditional RDBMS use block sizes of 32K-128K. The fastest drives on the market have transfer rates in the 100-300 MB/sec range, depending on who is doing the measuring. If the blocks could be read with no random access latency, that would be in the range of 800-10,000 blocks/second. But even if reading were instantaneous, random seek latency limits that to a mere 70-250/second or so. And that's even before taking into account the fact that – even with state-of-the-art caching -- an index-based lookup can make several disk reads for each row eventually found.

Table scans can be faster than index-based selection.

Sequential table scans, however, can actually read data at close to the theoretically maximum speed. So even though they have to retrieve much more data at a time, appliances that rely on sequential, index-light processing really can be faster than conventional index-heavy RDBMS. And while our argument so far has been pure theory, customer experience has shown that it's true in practice as well.

DATAllegro's MPP data warehouse appliances

DATAllegro is a poster child for modern MPP data warehousing.

DATAllegro is a poster child for index-light MPP data warehousing, with enough customer success and competitive proof-of-concept wins to validate its approach. Key aspects of DATAllegro's technology include:

- Unconventional use of standard computer hardware.
- A full-featured standard DBMS.
- Proprietary parallel data management built on top of the standard DBMS.
- Optimization for sequential rather than random data access.

It used to offer Type 1 appliances.

DATAllegro's hardware strategy resembles that of security and antispam appliance makers. Even when it still made its own hardware, it used conventional processors, disks, and so on, except in two areas where appliance vendors commonly deviate from computing norms – networking and encryption. In those areas, it still used standard parts; but they were ones rarely found in general-purpose computers. This is an example of what we call "Type 1" appliances.

Now it offers Type 2 systems.

As of its latest product generation, however – DATAllegro V3 – DATAllegro has switched to the Type 2 camp. That is, its appliances use utterly standard hardware, albeit in prespecified configurations. The main elements are Dell servers, EMC storage, and Cisco Infiniband boxes. Unlike some appliance vendors, DATAllegro also uses a standard operating system – 64-bit CentOS Linux. Besides the use of Infiniband, DATAllegro's most unusual architectural choice is that the disks within each EMC storage unit are split into two RAID1 arrays of six disks each, with each RAID array being dedicated to one Dell server.

Included is a full-featured RDBMS …

The core DBMS for DATAllegro's appliances is Ingres. Once a close competitor to Oracle, Ingres languished for various business reasons, and is now open sourced. In essence, it's a state-of-the-art 1990s RDBMS, with transactional capabilities robust enough for just about any "operational data warehouse" use. Particularly important are range partitioning capabilities, which commonly obviate the need to do full table scans.

… which has been modified for parallelization.

Ingres itself isn't an MPP system. But DATAllegro has modified and extended it for massively parallel operation. Parts of this work seem straightforward; indeed, there's no need to change query parsing at all, while optimizer modifications in essence just memorialize the changes in the execution structure. Rather, the hard part lies in query execution, specifically in moving data around. The biggest issue is the management of intermediate result sets, and distributing them to the proper node. If joins were only done two tables at a time, MPP probably would have been the standard DBMS industry architecture a decade ago.

The key is how the pieces fit together.

Arguably, the parallelization piece is the only major part of DATAllegro's technology that's proprietary at all. Rather, the big technical accomplishment lies in how it all fits together. MPP exploits parts-manufacturing efficiencies. Sequential reads solve the disk speed bottleneck. Fast data transport takes the sting from MPP. Cheap CPUs slice through the large rowsets brought in by the sequential reads. Yes, MPP software design is hard. But DATAllegro and other vendors have shown how to do it. At least for high-end data warehousing, shared-everything SMP is now an obsolete technology.

About the Author

For more than a quarter-century, Curt Monash has been a leading analyst of and strategic advisor to the software industry. Praised by Lawrence J. Ellison for his "unmatched insight into technology and marketplace trends," Curt was the software/services industry's #1 ranked stock analyst while at PaineWebber, Inc., where he served as a First Vice President until 1987. Since 1990 he has owned and operated Monash Information Services, a highly acclaimed technology analysis firm focused on enterprise software. He has been extensively published and quoted in the technology and general business press, and has been a regular columnist for Application Development Trends, Software Magazine, and Computerworld. To get Curt's latest research, please see www.monash.com/feed.php .

Prior to his business career, Curt earned a Ph.D. in Mathematics (Game Theory) from Harvard University at the age of 19. He has held faculty positions in mathematics, economics and public policy at Harvard, Yale, and Suffolk Universities. For more information please see www.monash.com .

About the Sponsor

DATAllegro entered the market in 2003 with the goal of making data warehousing more affordable and more valuable to companies than any other offering. After researching the technology available at that time, DATAllegro invented a new way of distributing data across a number of servers and then running queries in parallel. Integrated with hardware, storage and a database, the end result was a data warehouse appliance that represented a true breakthrough in data warehouse price/performance. Instead of paying millions for a traditional system, companies could achieve a 10-100x improvement in query performance, at a fraction of the cost of other providers.

The company can be reached via www.datallegro.com .

Further Reading

For more research on the subjects of this white paper, please see www.dbms2.com , specifically www.dbms2.com/category/relational-database-management-systems/rolap/. Future research may be found via the free RSS and e-mail subscriptions at http://www.monash.com/feed.php .