SALIENT

Salient UXT®

A New In-Memory
Analytical Processing Platform

Salient
UXT®

UXT, A New In-Memory Analytical Processing Platform

Executive Summary / Overview:

Salient Corporation's UXT is an in-memory multi-dimensional data compute platform and integrated analytical tool set that defeats traditional limitations of other on line analytical processing (OLAP) architectures such as, relational (ROLAP), multidimensional (MOLAP) and the hybrid (HOLAP). To one degree or another, all of these architectures involve serious tradeoffs among speed, granularity, simplicity and flexibility in use.

Both ROLAP and HOLAP offer the user potential access to unlimited detail of the underlying Relational Database tables but must depend, ultimately, on relational database operations, which may impose minutes or hours of latency, at least for analytical processing purposes, when dealing with data that has not been previously aggregated.

For this reason, ROLAP and HOLAP products are most often presented as IT toolkits, out of which IT professionals may fashion "reporting" systems. These systems deliver very good speed, but must sacrifice for users the possibility of interactive and dynamic inquiry.

Traditional MOLAP Architecture

This paper will concern itself mostly with typical MOLAP, the architecture to which Salient's UXT technology is most often compared. MOLAP engines can operate independently of the supporting databases and, like HOLAP, can deliver very fast response times for pre-aggregated data. Also, like HOLAP and ROLAP, MOLAP greatly limits the potential for interactive and dynamic query.

Typical MOLAP limits database size and often increases build (load and calc.) times.

The result is a compromise that may:

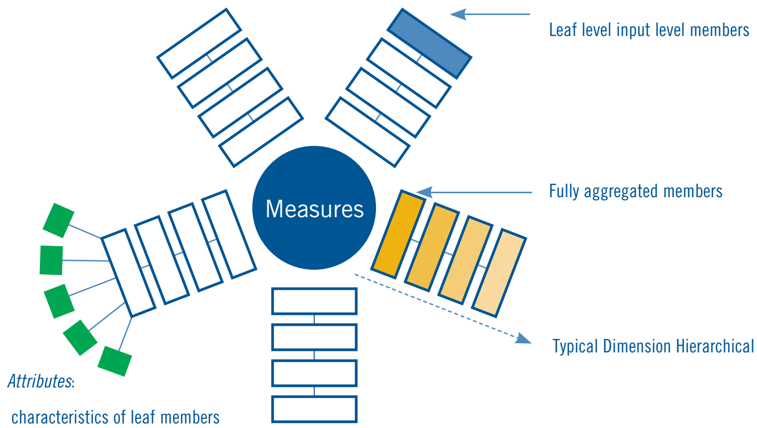
- Limit the number of input records, reducing time range for analyses
- Reduce the number of dimensions available, limiting user flexibility
- Increase cube count, making use more difficult
- Reduce response, often making the information irrelevant to the current need to apply it

Commonly, attributes (or properties) may be used to reduce size and build time. These may provide the ability to further filter dimensions. Attributes are index-based and do not require pre-aggregations, and require less disk space and build time. However, the user must face two significant challenges:

a. Use of attributes negatively affects performance, often dramatically

b. The interface will not support attributes as richly as it will dimensions, reducing the overall value to the user... they may see more, quickly, at a high level, and lose drill down capabilities – or – drill down but with limited visibility of key information.

Figure 1: Traditional MOLAP Architecture



Salient UXT Architecture

Like MOLAP engines, UXT has a mathematical science model basis. However, the optimization methods employed are completely different.

UXT does not use a hierarchical organizational model for data. Rather, it uses a model that links all data to all other data on whatever axis – they are associated.

UXT does not aggregate or pre-calculate data whatsoever. Rather it uses a variety of techniques (see associated papers) to manage extremely large volumes.

An additional benefit is the elimination of consulting time required to analyze how data is to be calculated or summarized.

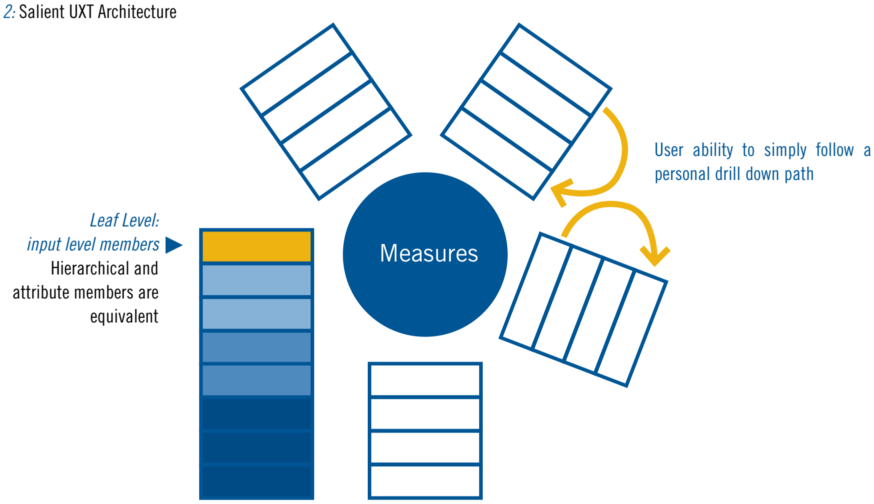
Highly compressed data in an associative data model with all calculation and aggregation performed at run time provides users with extremely fast response time, usually seconds or sub-seconds.

The UXT approach reduces issues of database size and build time while providing a time-scope and dimensionally rich, simple to use, speed of thought technology.

Users get "dimensional freedom" because attributes are implemented in UXT as dimensions.

The Salient user interface differs from rigid MOLAP hierarchies and through the Salient Interactive Minder™ (client) the user is free to create a natural and simple path to the information they need ad hoc.

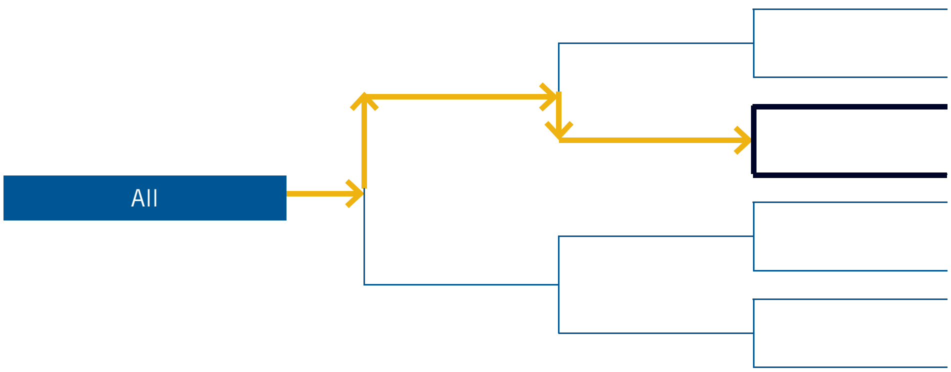
Figure 2: Salient UXT Architecture



MOLAP

The typical MOLAP drill path allows the user to progress one level at a time. That means users can not intuitively make visible information on any dimension or attribute without following a specific path – translated, if a user has an insight requiring different data, he or she must start over. Although not difficult, rigid hierarchies limit free association of activity to results and thus are limiting to users.

Figure 3: Typical MOLAP Drill Path



Drill Paths – MOLAP vs. Salient UXT (continued)

UXT – Flexible Drill Path

In UXT the hierarchy is virtually invisible to the user and never gets in the way of moving through any data reviewed by any dimensions or stopping and selecting a completely different view of the information essentially on the fly, with ease and speed, slicing and dicing to get to root data that supports accurate assessment of how activities are connected to performance results for any person, place or thing that may affect the outcome.

Figure 4: Salient UXT Flexible Drill Path

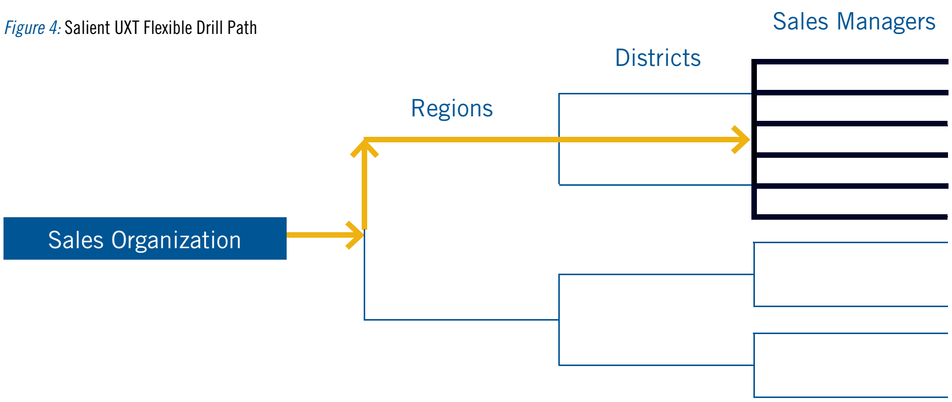
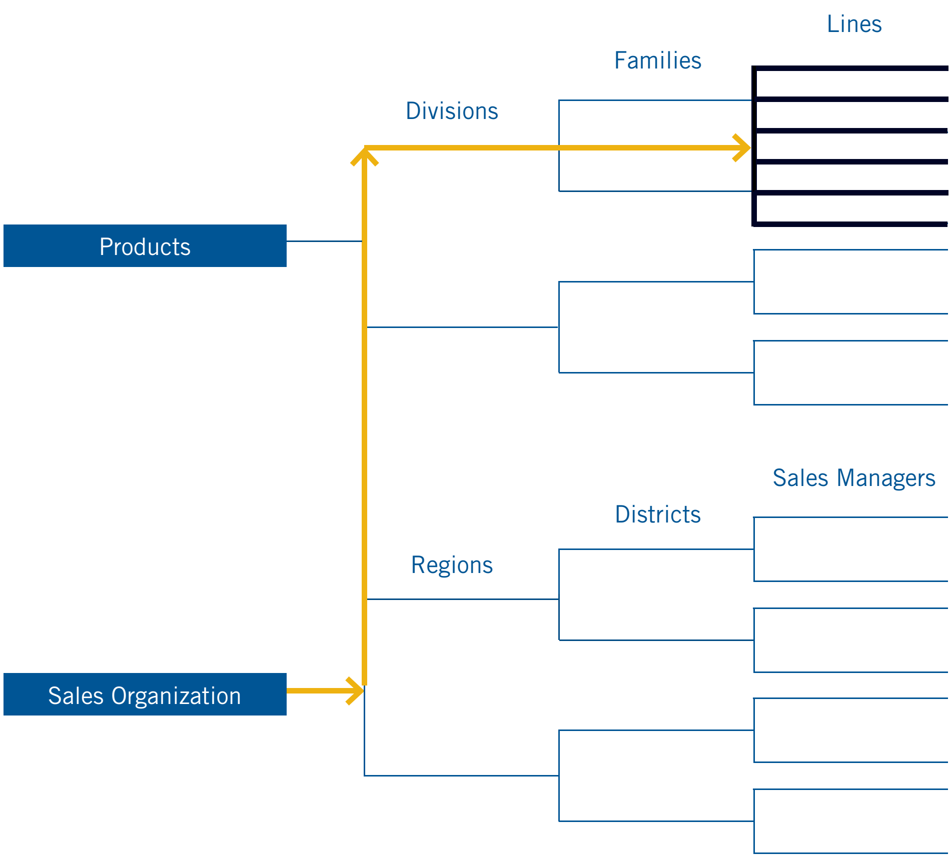


Figure 5: Salient UXT Unlimited Possibilities



** A dimension,* in the context of this white paper, refers to an answer set that results from a user request from pre-aggregated data. This is standard OLAP terminology.

 # An attribute, in the context of this white paper, refers to a rollup that is aggregated at the time of the user request. This option allows more flexibility in rollups of granular data, but sacrifices response time. It is often done to deal with operational constraints for space or aggregation processing time.

SALIENT

Salient UXT® System Scalability

The Salient UXT® system is a highly scalable, multi-dimensional on-line analytical processing (MOLAP) system, that does not pre-summarize or pre-calculate granular data, and does not rely on relational databases for access to granular data.

Executive Summary / Overview:

The use of OLAP and BI (Business Intelligence) tools, once limited to high-level corporate analysts, is now an integral part of the decision making process at all company levels. The application has also changed from retroactive reports at the end of a financial period to proactive day to day analysis of integrated business data.

Moreover, as greater numbers of people need to draw intelligence, and as companies collect and store more and more granular data from a widening number of business operations, databases have grown into huge information silos, each holding the history of millions, even billions, of individual transactions.

The compounding effect of database size and number, together with spreading user demand for interactive transactional and managerial decision support, have created a need for BI systems to integrate information from many processes, and scale to very large data volumes, without sacrificing ease of use, query performance, or uptime. This paper will discuss how Salient's UXT® technology achieves high performance and high scalability.

The Salient UXT® system is a highly scalable, multi-dimensional on-line analytical processing (MOLAP) system, that does not pre-summarize or pre-calculate granular data, and does not rely on relational databases for access to granular data. The technology represents a new approach to the problem of high-volume information distribution, taking maximum advantage of ever-increasing modern computing power, and using a variety of techniques to achieve new levels of scalability and performance, without sacrificing the informational value contained in the fine grain of everyday business data.

Single Machine Scaling

Keeping data in random access memory (RAM) allows a system to process data hundreds of times faster than by electro-mechanical input-output (processor to disc) operations. Advanced data compression techniques allow UXT to handle very large volumes, and at the same time, take advantage of the speed of in-memory processing. This speed advantage is compounded by Salient's proprietary n-dimensional GRID indexing scheme, which allows a processor to go through only that portion of data most relevant to the query at hand.

UXT also takes full advantage of both multi-threading platforms, and multiprocessor machines to accommodate very large numbers of concurrent user queries without performance degradation. Increasing the number of processors will scale the number of concurrent users in near linear fashion.

More advanced operating systems, such as Windows® Advanced Server, are able to use memory above the physical 4GB byte limit of 32-bit processors, allowing even greater data volumes on a single machine.

A fully configured 32-bit PC server, for example, will handle data volumes of greater than 100 million annual business transactions (a single sales transaction, will be compressed to consume ~ 11 - 20 bytes of memory), and accommodate hundreds of concurrent users with query response time, typically, of less than 5 seconds.

Finally, UXT has been built from the ground up to take advantage of emerging 64-bit processor and operating system technology, eliminating the physical limit of 32-bit processing altogether.

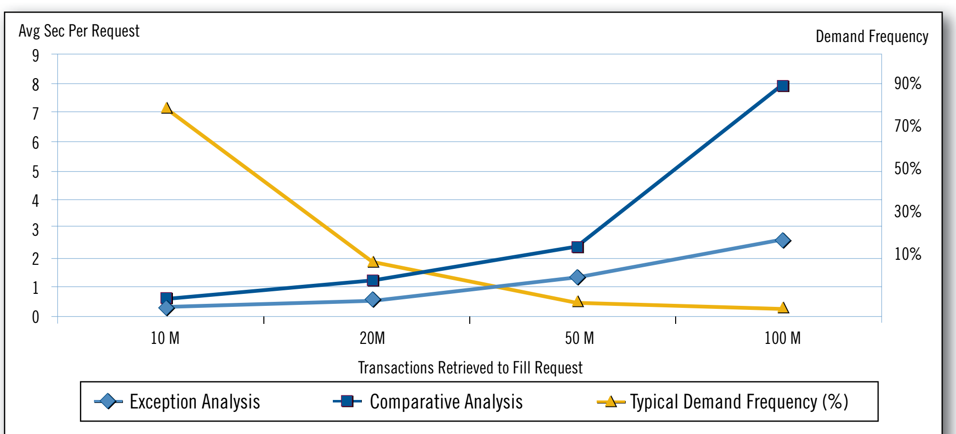


Figure 1: UXT Scalability for Large Comparative and Exception Reports

Findings: Approximately 90% of typical requests are handled in 1 sec. while over 95% of requests are handled in under 3 sec.

Study Size: average of 100 requests per datapoint

Server Specifications: 2 2.4 MHz CPU's with 8 Gb RAM

* Exception Analysis tests a criteria (e.g. customers that have not made a particular product purchase in the last 6 months) and presents results as pass/fail counts and percentages with the ability to drill to details.

** Comparative Analysis compares two time periods (ie. year-over-year or this month vs. last month) and presents results as this period, last period, difference, change %, mix, etc.

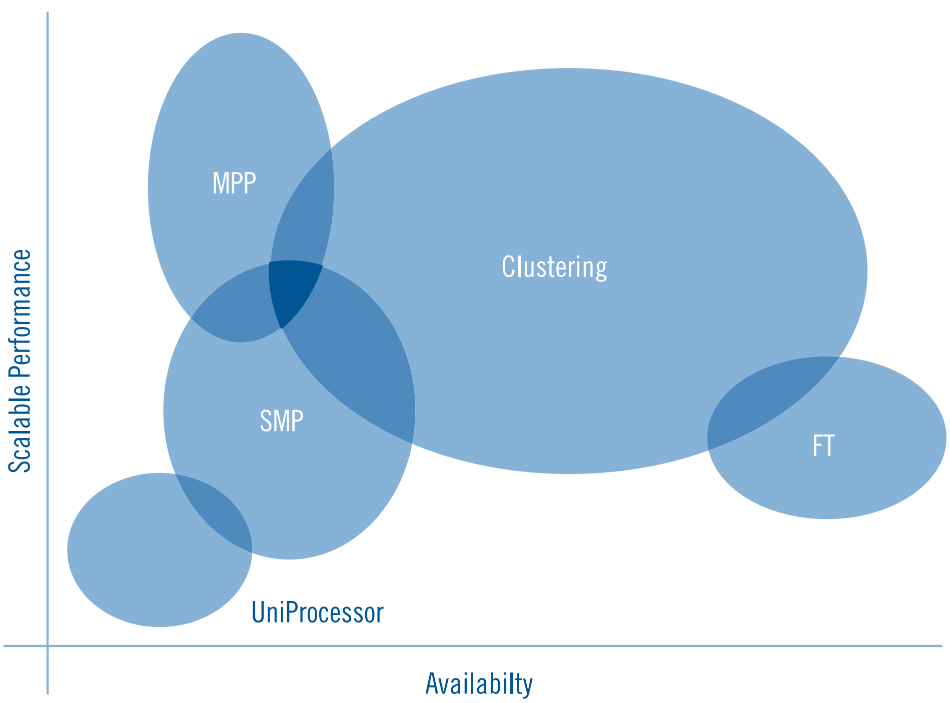
Massive Scaling With Partnered Arrays of Computer Equipment (PACE)

Salient UXT uses a 'divide-and-conquer' methodology to achieve very high scalability. A Salient PACE array consists of a server, known as Array Master, controlling multiple individual servers, referred to as Nodes. The Array Master is responsible for presenting a coherent view of the data to clients. Users (clients) may access single-server data on any one server, or distributed data through the Array Master.

Any of the nodes in the UXT PACE have complete control over their portion of the data and their computing assets, and use multi-threading to take advantage of multiple CPU's to enhance performance.

This is Massively Parallel Processing (MPP) in its most efficient form. As shown in Figure 2, MPP is the most scalable form of multiprocessing. The Salient UXT PACE also allows for each node to have its own hot backup to push availability to its limit.

Figure 2: Scalable Performance (Y) vs. Availability (X) for Monolithic, SMP, MPP, FT and Clustering Systems [1]



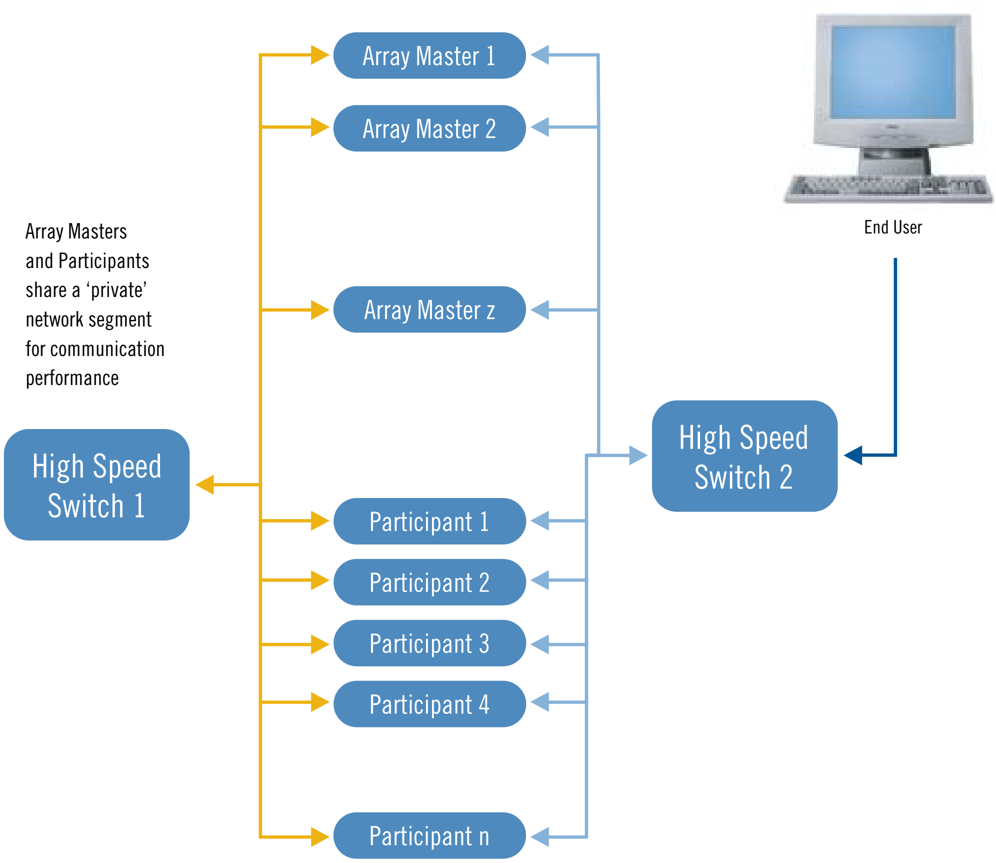
Multiple Access Points

UXT PACE array technology scales to thousands of users by providing for multiple access points to a partitioned database. Users who need to access data that is contained entirely on one of the Nodes, will access only that machine. On the other hand, those who need access to data that is spread across more than one participant will gain access through an Array Master.

A useful example is a major tennis event, where many matches are happening at the same time. Some people may hold tickets for only one match, and some may hold tickets for all of the matches. If everyone is required to go through one entrance and exit, then an artificial bottleneck is created, and the spectators may not get seated in time. On the other hand, if each court has its own set of entry and exit points, all fans can get to their seats before the matches start. This corresponds to UXT's direct Node access scheme.

The tennis complex may also have an area where special ticket holders and media reporters can move about more freely from match to match. This corresponds to access through the Array Master.

Figure 3: PACE Configuration



One way to avoid the single-point-of-entry bottleneck is to partition the data according to company organization. Those data consumers who need the overall view have their entrance, and those who need only their own restricted view of a portion of the data have their own entrances.

Another way is to partition by data type. For example, one partition may have all sales and customer facing data, and a second all manufacturing data. A high-level executive may need to know and combine information from both of these areas of the business, but a sales manager only needs to see the sales information, and the manufacturing manager may only need to see manufacturing information.. The PACE Array may also be partitioned by time, or any combination of time and the above examples. In the case of partitioning on time, if a request only needs data for a particular month, then that request is only distributed to those nodes which carry data for that time frame.

Because of the flexible nature of the UXT PACE, multiple array participants may have more than one master. For example, suppose sales information is partitioned onto four computers based on year, and manufacturing data is partitioned onto five computers based on manufacturing facility. You may have three array masters. As shown in the diagram below, Array Master 1 is gathering and consolidating answer sets from all nine participants, for the high-level executive. Array Master 2 is gathering and consolidating answer sets from the four sales participants for the sales manager, and Array Master 3 is gathering and consolidating answer sets from the five manufacturing participants. Individual manufacturing facility managers could just attach to the one manufacturing participant that has the data that is relevant to them. So as you can see, the load may be distributed and accessed by many different methods, creating a flexible and extensible schema.

Figure 4: Example of Array Masters

Uptime

The second issue that the UXT PACE system addresses is uptime. The PACE may be set up with redundant participants to provide near-instant fail-over, so that answer sets may continue to be delivered to the end data consumer with minimal interruption. This method is much more economical than a huge fault tolerant system of comparable computing power.

For example, in the above scenario with five sales participants, an additional five computers can be available on ‘hot’ standby. The array master will be configured to know that there are five participant groups and each group has participant A and participant A’. If the participant A server no longer responds to the Array Master, the Master automatically switches to the participant A’ server for that group, and continues to gather and return answer sets to the end data consumer.

Another scenario would be to have one or two backup servers defined in the array. An employee with responsibility for the array could be notified when one of the participants goes down. They could then bring up one of the backup systems to take the place of the failed array participant. This method gives high up-time without a 100% duplication of array participants. In this scenario, the data must be kept on a shared network resource available to both the primary array participants and the backup participants.

The UXT PACE Array Masters also use intelligent data caching. If a participant in the array is removed from service for a short period of time to add or update data and then brought back up, the array master can determine what data has changed. It uses this information to only run a subset of initialization steps required.

UXT Server Speed

The Salient UXT system utilizes several techniques to achieve high data-processing speeds. Among these are data compression, n-dimensional grid indexing, and application of modern programming techniques.

The Salient UXT system makes extensive use of 'data-based' compression. By this we mean that the data in any particular time-dependent cube is examined, and the most efficient data field compression is determined. For cyclical / seasonal businesses this could mean large savings in database size for those months which have smaller data values.

For example, if a company sells bathing suits only in the North Eastern United States, sales quantities and income dollars per shipment to stores are probably not very large in November and December, and very large in May / June / July. The UXT system will use the most efficient compression technique for each of these months.

This compression allows the UXT server to keep more of the database resident in memory for rapid access. Even if your business is not cyclical, the UXT system will still use the most efficient data field width possible.

Our n-dimensional grid indexing schema is a modern, extremely efficient indexing system that is optimized for our multi-dimensional / multi-cubic view of the world. If you have more keys per record, more index dimensions are automatically created to keep extraction and summarization of data as efficient as possible.

When the UXT server is processing data, it first gets the set of possible matching records from the grid index. With the use of modern generic C++ programming with template classes, a highly efficient inner loop is used to extract and summarize data from the selected records.

Another technology used by the UXT server allows the large data fact files to be loaded into the server in chunks. This chunking allows the server to keep only that portion of the file in memory that is absolutely necessary to answer a particular question. Portions may be swapped out to make room for other data facts needed by other threads.

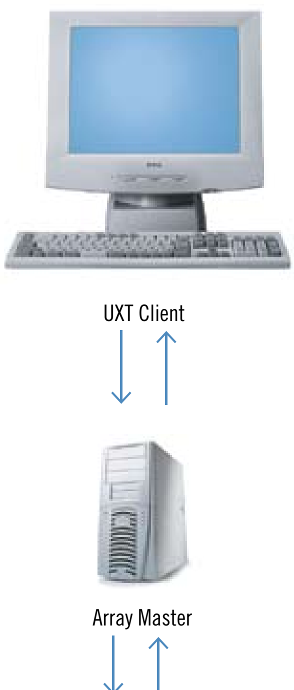
Putting it all Together Salient's PACE Technology provides comparable throughput to an individual UXT server solution with solution scalability moving into billions of annual business transactions. The diagram below depicts the life cycle of a user request utilizing an Array Master server and many Array servers. The Array Master server adds little overhead to the ultimate throughput of the client request.

Typical Request Elapsed Time

approximately

.5-5 seconds

~ 0 sec - approx. 0 secs



UXT request made by client ~ 0 sec

Latency time request to Meta Server ~ 0 sec

Meta turns client request into a request per appropriate Array Server ~ 0 sec

Latency time Meta request to appropriate Array Servers ~ 0 sec

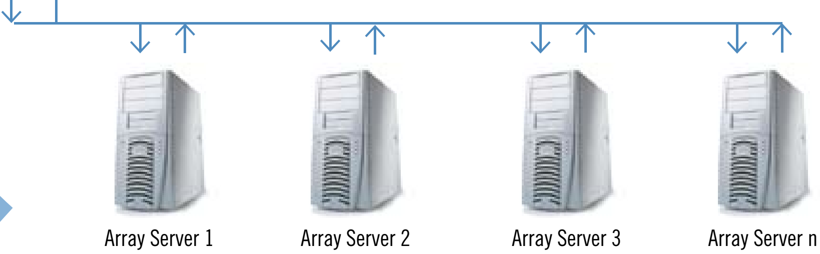
Results returned to client ~ 0 sec

Latency time Meta results to client ~ 0-.5 sec (based on result set size)

Meta server consolidates and aggregates results
from Array Servers ~ .1-.5 sec (based on result set size)

Latency time Array Server results returned to the Meta Server ~ 0-.5 sec
(based on result set size)

Array servers process individual requests ~ .1-.3 sec (slowest of requested Array servers)



Assumptions:

- Client connects to Meta Server minimum of 100 Base -T
- Meta Server connection to Array Servers 1000 Base-T

SALIENT

Pre-aggregation and Data Explosion

The traditional MOLAP architecture provides the user an extremely fast response time by pre-aggregating data. The weakness of this architecture is that a balance must be reached between build time, response time, data storage requirements and the delivered solution.

Build Times

Build times, the time to load new data to a cube, may take hours in order to preaggregate and store tens of millions of possible outcomes. Most MOLAP databases allow the database administrator to control the extent of the pre-aggregation to reduce build times (and/or data storage requirements). But, by shortening a build time the administrator will be negatively affecting the response time to the end user. UXT server stores and maintains the data at the input level. It does not preaggregate, minimizing downtime due to data loading. Throughput of the “Acquire” build process will commonly approach 1 million line items per minute. This administrative “non-issue” means that less time is spent on solution administration.

Response Times

The designator On-Line Analytical Processing (OLAP) means that the response time should allow interactive analysis (1-5 sec per request). As data requirements have grown in volume, granularity and scope, many OLAP products can no longer meet the speed criteria in the vast majority of solutions.

UXT continues to provide a significantly fast response time for most requests. Designed for large transaction volume data, unlimited granularity and enterprise level integration, UXT performs at OLAP speed as demands grow.

Data Storage Requirements

MOLAP cubes are at the risk of exponential data explosion with the addition of any new cube requirements: extending time scope, increasing measures, adding to granularity (scope of dimensionality), etc. The pre-aggregation of data drives this constant risk of requiring more and more disk space. Database administrators commonly will have to put limits upon the time scope of the data, the dimensionality, etc. in order to have a maintainable cube.

UXT not only stores and maintains the data at the input level but it compresses the data to further minimize how much disk space is required. To date, disk space has never affected an UXT solution design.

Delivered Solution

As stated above, the typical MOLAP solution must balance an optimal user solution with multiple architectural constraints with respect to build times, response times and disk space. The sacrifices to the optimal user solution are oftentimes substantial (though, under publicized): 5-8 dimensions vs. practically unlimited (UXT); base data in weeks or months vs. days (UXT); and timeframe of data in months vs. years (UXT).

Relational OLAP

An alternative solution to MOLAP is Relational OLAP (ROLAP). ROLAP uses a relational database with a minimum amount of data aggregation. This approach allows for the same breadth of solution as Salient's UXT, included more transactions, more granularity and a solution with greater breadth. Unlike UXT, ROLAP does not provide true interactive analytics because the response time can be minutes, or even hours.

In Conclusion

Salient's UXT System optimizes the use of modern computing power to provide the best-of-breed Business Intelligence application for very large transaction volumes. Scalable to billions of annual transactions, UXT provides benchmark OLAP quality response times without sacrificing anything in the solution deliverables.

References

[1] Johnson, Jim 1998. *The UNIX® Operating System: A Robust, Standardized Foundation for Cluster Architectures. The Journal of Open Computing*

About Salient

Salient Corporation makes very large scale in-memory intelligence technology for ad hoc data interrogation, visualization and root cause analysis. The company provides continuous audit, performance monitoring and forensic applications for business, health care, education and government.

SALIENT

Salient
UXT®

SALIENT

Salient Corporation
203 Colonial Drive
Horseheads, NY 14845 USA

phone: 607 739 4511
fax: 607 739 4045
salient.com