High-Availability Web Site/NSK Cooperative Processing Using Database Replication in a ZLE Architecture

Paul J. Holenstein
Executive Vice President, Professional Services
ITI, Inc.
Paoli, Pennsylvania

Dr. Bruce D. Holenstein
President
ITI, Inc.
Paoli, Pennsylvania

"Everything changes but change."
– Zangwill

Introduction

The Zero Latency Enterprise (ZLE) initiative has created tremendous excitement as business personnel come to understand the benefits that the architectural concept provides. After all, what enterprise wouldn't want to instantaneously collect, disseminate, digest, and truly understand customer information as that customer interacts with various touch points across the organization?

Unfortunately, this excitement and comprehension often doesn't quickly filter down to the overburdened IT staff as they are once again charged with rapidly implementing the next "greatest" architecture or technology (remember Systems Application Architecture and CASE tools/knowledge engineers?). For example, at the March 2001 Canadian TUG User's Conference, one speaker asked the room of about 70 attendees for a show of hands indicating how many in the audience knew what "ZLE" meant (pronounced ZED-L-E in those parts). Only three or four hands went up. Does this indicate ignorance, overwork, shyness, or apathy? Perhaps it means that the prospect of ZLE is a long way off at many organizations that don't have immediate funding, or have tried and can't figure out how to implement the concept.

We can thank the forward-thinkers at the Gartner Group, Inc. for formalizing the concept, and Compaq for aggressively promoting an architecture based on the ZLE principles. Simply stated, a ZLE architecture promotes extremely fast responses to business stimuli, and provides a readily available consolidated view of the events across the entire enterprise or enterprises, regardless of the source of those events. ZLE eliminates operational inconsistencies inherent in typical legacy system and application data/business logic sharing, and drives the time lag between information "float" (time from event occurring and the event being properly disseminated/incorporated into all aspects of the enterprise) towards zero.

The Gartner Group makes the bold prediction that almost every successful business will need a ZLE infrastructure (www.compaq.com/newsroom/pr/2001/pr2001041702.html). Surely, this concept should not be overlooked. But, as is common with many powerful concepts and architectures, exactly how to achieve a goal remains elusive because people don't always immediately understand or have time to digest how the broader picture works. Lofty principles need to be grounded in the here-and-now nature of our daily lives.

The goal of this article is to demystify some aspects of ZLE that have been, and continue to be, implemented via the well-known and full-featured data integration technologies. The focus will be on presenting a real-world business methodology that uses database replication for integrating a Web site into an NSK environment, preserving the security and OLTP autonomy of the NSK environment, while leveraging the Web interface characteristics of NT/UNIX platforms running Internet-enabled production databases such as Oracle.

Using Data Integration to Provide a Low-Latency Decision Framework

Other articles in ***The Connection***, as well as various Compaq white papers (available at www.compaq.com/zle), provide considerable background into the ZLE architecture's evolution and overall structure (i.e., *ZLE Architecture White Paper*, at http://zle.himalaya.compaq.com/view.asp?IOID=5351).

Classic ZLE architecture defines enterprise application integration (EAI) as the method to synchronize and route information across an enterprise. Simply stated, the ZLE approach allows heterogeneous, and sometimes asynchronous, business processes/systems to rapidly share information about business events via "pushing" the information to/from each other. EAI prompts each component of the system to act on the event in near real time.

Classic ZLE architecture also defines the concept of the operational data store (ODS) as the central repository for business information, providing a complete and consistent

Paul J. Holenstein is executive vice president, professional services of ITI, Inc. Holenstein has more than 20 years of experience providing professional services and turnkey application development solutions on a variety of platforms, dating back to the Nonstop I. His expertise areas include high-availability designs, replication technologies, heterogeneous application integration, communications, data warehousing, and performance analysis. Holenstein holds patents in the field of data replication.

Dr. Bruce D. Holenstein, president of ITI, Inc., began his career in software development in 1980 on a Tandem NonStop I. His fields of expertise include algorithms, communications, data warehousing, data replication, imaging systems, process control, and turnkey software. His database expertise includes Tandem NonStop SQL/MP and Enscribe, IBM, Oracle, Microsoft SQL Server, and Sybase. Holenstein has co-founded and run three successful companies, and holds patents in the field of data replication.

The authors can be reached at 610/647-6250, Shadowbase@iticsc.com, or visit www.iticsc.com.

view of the customer and that customer's relationship with the organization. A central ODS becomes the standardized site to retrieve (or pull) information about the customer, and is used to feed all other external (to the ZLE, core) system components, such as data warehouses and marts, with operational data.

Compaq also defines additional snap on applications and interfaces geared towards simplifying the interface of the ZLE core components to other customer desired capabilities such as analytic modules or popular industry software such as SAP or Siebel. One would expect such interfaces and mapping transformers to help "sell" the acceptance of the ZLE core into complex legacy user environments.

Note, however, that many ZLE benefits can be realized through effective deployment of straight-forward data integration technologies as well. As a matter of fact, businesses have been attaining these ZLE benefits for many years, long before the term ZLE became so popular. For example, integrating two heterogeneous applications (possibly with separate databases and on separate nodes) can be as straight-forward as using a near real-time data replication product to synchronize the data across the environments (providing the consistent-data-view pull capability). Using the product's equivalent of a database triggering mechanism can drive event-driven business processes to act on the data in near real time as well (the so-called push capabilities of application integration).

Often, using data integration can minimize overall system impact and disruption as well since legacy applications do not need to be modified to interface to newer EAI technologies (such as MQ Series, Tuxedo, CORBA features, and Enterprise Java Bean facilities). Whereas EAI approaches are often tightly-coupled into the legacy applications, data integration is typically loosely coupled and isolated at the database level.

The term low latency is a critical component of any ZLE implementation. It is defined as the time lag between an event occurring and the business' ability to fully process and digest and/or disseminate it. Extreme low latency enables what Compaq refers to as "Business At The Speed Of Now." If you don't have low latency (where latency is typically defined in the low subsecond range), you have "business at the speed of almost, sort-of, or any-minute-now now" – clearly not acceptable in many 21st century applications. Note that latency is directly related to the amount of time the implementation takes to capture the event, process it, and integrate it to the rest of the components in the framework.

A "decision framework" refers to the ability of the organization to make accurate business decisions based on the events/data that are provided to it. Typically, this framework revolves around one or more powerful, complete relational databases, and it includes processes for manipulating that data into useful results.

Now that we've introduced data integration as a powerful method to attain ZLE goals, and understand that low latency is a core component of any ZLE architecture, we will delve into a specific data integration method called bi-directional data replication and its benefits.

Using Bi-Directional Data Replication to Provide True Heterogeneous Cooperative Processing in a High-Availability Solution

Figure 1 provides a glimpse of the framework we are striving for by showing a real-world example of a heterogeneous cooperative processing environment – a brokerage application that has been scaled across a Web-enabled UNIX server running Oracle and an NSK server. As the rest of this section will show, these results are readily attainable by following the design principles described herein.

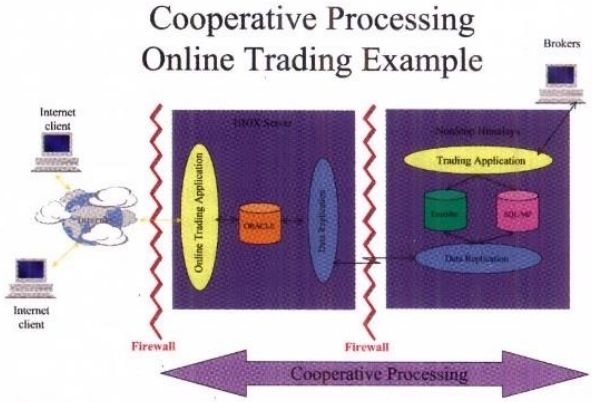

Figure 1

Cooperative processing refers to the ability of the system to effectively share the workload across all of the components in the system, in a peer relationship, regardless of whether or not they are homo- or heterogeneous system and database types. All components effectively carry their own weight, each performing their selective duties. Think of a system in this context not as an individual node of a cluster, but rather as the entire network of interconnected nodes.

In homogeneous cooperative processing environments, it does not matter which node in the system processes the event since each and every node would produce the same result if directed to process the event. Typically, we would have the nodes share the information derived from processing the event so that the nodes remain synchronized (fully knowledgeable about the data in the others). No node has to be designated as a hot or warm standby node, as each has full, immediate access to all pertinent data and logic processes it needs to perform its designated tasks. Note that loss of one node in such a cooperative processing environment is transparent to the application, allowing for instantaneous backup takeovers, and on-line operating system or application upgrades.

Fully bi-directional data replication enables cooperative processing at the peer level, regardless of the pedigree of the peers. More specifically, bi-directional data replication keeps the two (or more) databases synchronized, so that all transactions that occur on the first node are replicated properly to the second, and all on the second are replicated properly to the first. This form of replication should be scalable in that it is not bound by two peers, rather it should scale (preferably

linearly) to handle much larger workloads across a much larger set of cooperating nodes. Fully bi-directional data replication has two critical components: *ping-pong avoidance* and *collision detection and avoidance*.

Ping-pong avoidance refers to the replication engine's ability to properly replicate a transaction (the ping) that initially occurs on one node to a replicate database, while avoiding the erroneous reverse replication of that same transaction back to the initiating node (the pong). Without such technology, transactions would endlessly ping-pong between the nodes, causing system overload and data corruption. For reference, consider the US Patent 6,122,630 by Strickler, et al (available at www.uspto.gov) as a background to a system and method for ping-pong avoidance. What is important about the Strickler et al patent is that it teaches the concept of *selective* ping-pong. Selective ping-pong, or replication of certain transaction parts back to the source node, can be used to verify that replication happens completely, correctly, and only once. Selective ping-pong thus can be used in other (patented and patent pending) technologies that provide *collision detection and avoidance*.

Collision detection and avoidance refers to the ability (some say magical or mystical) of the replication engine to detect pending application data collisions, regardless of the node(s) involved in the collision, and prevent that collision from occurring. Systems designers know of this problem as a version of the single node (or database) simultaneous update problem, whereby two peer applications collide when they update the same data on disk without first checking if their copy of the data is current.

The collision problem can best be illustrated via a simple example. Assuming that your bank contains two branches, with each branch replicating all transactions to the other for both cooperative processing and disaster recovery purposes. Also assume that you and your spouse each access your account at the same time from a different branch, with each branch using its local copy of the database. You each want to withdraw $100, and your account initially contains a balance of $120. Without collision detection/avoidance, you each would be able to withdraw $100, clearly not in your bank's best interests. Using a collision detection/avoidance algorithm, the system would detect the data collision, and provide for application-defined business rules resolution (typically let one of you proceed while the other receives the insufficient funds message).

Practical collision avoidance implemented with selective ping-pong requires that the data replication processes run at a higher priority than the application software and that care is used to choose only the important parts of a transaction to protect.

An interesting byproduct of collision avoidance is the (optional) ability for the application to periodically gate itself waiting for significant transaction events to be properly safe-stored into a peer node before proceeding. This may be useful, for example, in cases of extremely important or sensitive transactional data in a disaster recovery scenario to eliminate data lost in the pipe in the event of total node failure.

Due to the latency inherent in a selective ping-pong approach (and remembering that low latency is a critical component of ZLE), a low latency data replication engine is required. This is especially true in disaster recovery scenarios as one wants to minimize (or eliminate as described above) data that are lost in the pipe in the event of a total node failure. It is also critical that the data replication engine follows the natural-flow-of-transactions. The natural-flow-of-transactions refers to the replication engine's ability to replay the source node's transaction mix on the target node, following the same sequence as the events occur on the source node. Basically, the target node's database flows through the same state transitions as the source node's database did, offset by some time delta t (typically, t is measured in milliseconds for low latency database replication systems).

It is important to contrast the natural-flow-of-transactions approach to other transaction replay schemes, such as those that modify the flow by replaying the transactions in transaction-commit-timestamp order. Unfortunately, these schemes violate a cooperative processing principle – they remove the ability of the replication engine to effectively provide collision avoidance, as they replicate the data to the target in an after the fact sequence. One cannot avoid a collision if the data are already committed on either or both nodes.

Leveraging Heterogeneous Component Strengths of NSK and NT/UNIX, While Preserving Security and Autonomy

So far we have seen that the data integration method can provide cooperative processing environments with extremely low latency. Noting that the peers in this relationship can be

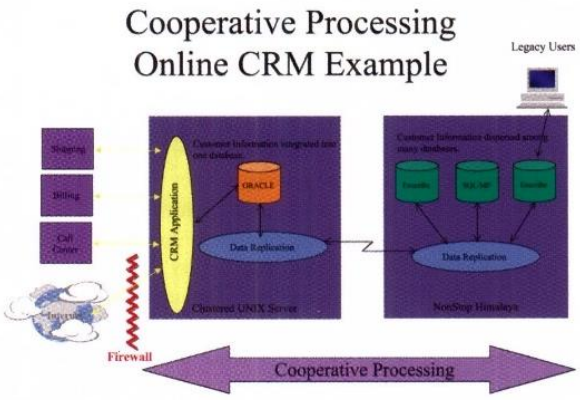

Figure 2

heterogeneous is interesting, and provides a variety of additional benefits, such as leveraging each system type's unique strengths and preserving each system type's individual security profile.

Bi-directional replication enables the separate nodes involved in the replication system to autonomously perform their individual tasks. They each implement their respective tasks as necessary, and only interact at the lowest (data integration) levels. Actually, as far as each node is concerned, they are independently performing their roles, using a local-

ized database, and can be totally blind to the existence of the other node(s).

By maintaining this level of autonomy and low level of integration, the heterogeneous architecture provides an extremely secure environment. It is only the replication engine that interfaces between two peers, via well-understood paths, rather than the applications, Web browsers, or users. For instance, when building a system of NSK and NT/UNIX Web servers, the Web-browsing public remains on the NT/UNIX Web server, and never directly interfaces with the NSK components. Refer to Figure 2 for an example case study of a Web-enabled CRM application where the database-of-record is stored on the back-end NSK platform while the CRM application is running on an Oracle platform.

Using Clustering to Increase Reliability

Clustering is inherent in the Compaq NSK architecture. But, individual cluster components (Expand nodes) do indeed fail for a variety of reasons, including sabotage, natural disasters, and operator error. Hence, the use of data integration to provide additional levels of reliability is typically warranted and cost effective, particularly when the data integration approach can provide cooperative processing using a bi-directional data replication engine.

For enhanced reliability, the data integration algorithms must be able to utilize the reliable architectures of the NT/UNIX world, for example *Microsoft Clustering Services* or Legato Clustering Services, as clustering (typically coupled with RAID disk technology) has become the method of improving overall system availability. Additionally, the data integration algorithms must be able to handle virtual IP and/or dual rail technologies for communication, and be cluster-aware as to the environment in which they are running.

Summary

In summary, we know that ZLE is a set of goals that can be attained via various approaches. One method to attain these goals is via data integration using database replication, a mature, straight-forward, and efficient approach. When using data integration, heterogeneous cooperative processing is available, allowing for disparate systems and processes to be connected in powerful ways. One benefit of this form of integration is that the systems remain autonomous, and individual security is preserved, of particular importance when integrating systems such as your Web-hosting sites into your NSK back office systems.

Replies from our Readers

The Connection *readers respond to recent columns from ITUG Chairman Janice Reeder-Highleyman.*

I just finished reading your message in the July/August issue of ***The Connection***, and I agree with you. Established customers are the bread and butter and shouldn't be taken for granted. I sometimes have to remind Compaq of how hard I have to fight every year to keep my budget for Tandem products. You are right - it is expensive, but worth it. The problem is, some people take what we do on the *NonStop™ Himalaya* platform for granted and expect the same performance from cheap, off-the-shelf solutions. I wish Compaq would work as hard selling *Himalayas* as they do selling PCs. After all, if they really want to follow IBM's lead, they should put their mainframes, with a decent profit margin, in the limelight and leave the low-margin stuff to Dell and Gateway.

Phillip A Kriley, Allegheny Ludlum Corp

You asked for the names of Compaq staff who have gone to great lengths to support me as a user. Steve Bradley, of the Pittsburgh office, is probably the most important Compaq employee to my company. Steve has consistently gone above and beyond to support us - everything from getting me numbers to crunch in preparing my budget to recommending solutions and alternatives. He is exceptionally knowledgeable about the *Himalayas* and the other Compaq offerings. In a recent presentation I was amazed at how he answered specific questions about the various products without looking up the info - it was all in his head! He has been a great resource to me and my company.

The field engineers in Pittsburgh also do a great job - Fernando Alarcon, Jim Logan, and Greg Metts. From the OSC, the best folks have been Mary Ward for DSM/SCM, Mike McCarthy for TCP/IP issues, and Barry Scott for hardware support.

At the executive level, Pauline Nist has consistently answered my questions, provided quotes, and been very supportive in helping me sell Compaq *NonStop Himalaya* to my superiors. She really believes in the Tandem products and I don't know where the NonStop Division would be today without her enthusiasm and support.

Phillip A Kriley, Allegheny Ludlum Corp