THE CLIPPER GROUP
Navigator™
Navigating Information Technology Horizons℠
Published Since 1993 Report #TCG2011007 February 9, 2011

TransLattice Provides the Best of Both Worlds in a Single Platform

Analyst: Anne MacFarland

Management Summary

The cloud imperative, when contrasted with *build-it-yourself, run-it-yourself,* is a very good idea – economically. The customization inherent in *BIY/RIY*, particularly when it is built out over time, makes comprehensive change difficult. In that environment, enterprise applications, particularly data-centric applications such as database management systems, become chronically needy, like the carnivorous plant *Audrey* in the musical and movie *The Little Shop of Horrors*. *Feed Me* is Audrey's mantra. By contrast, cloud services are broadly available and share-nothing cloned environments that are less expensive to deploy and support because all customization is buried inside the application. However, a pure and absolute move to an outsourcer or cloud, while financially attractive, also introduces latency and, often, telecommunications costs that reduce that strategy's cost-effectiveness, particularly in the case of core enterprise applications that consume and digest large quantities of data. In that case, there is a good argument for local, on-premises deployment. **What is really needed is the best of both worlds (local and remote cloud deployment) *in a single system*.**

TransLattice, Inc., of Santa Clara, offers just such a hybrid system of local and cloud instances. Its *TransLattice Application Platform (TAP)* adds system-level autonomics that reduce the cost of maintaining the application. Like any infrastructure, its value is an intertwined matter of scale, resilience, and cost. The distributed nature of the TAP architecture modernizes all three. Scaling operations is very flexible, due to its hybrid-with-cloud services. Resilience (both reliability and availability) of distributed architectures is cumulative – the more nodes, the more the resilience that can be built in. How that resilience is implemented determines both the effectiveness and the cost of the effort. TransLattice has many patents in this area. Finally, TransLattice has focused on redesigning its control elements to take out costs, learning from previous distributed architectures, such as Grid. In any system, cost must be aggregated to the system as a whole and measured against the results, not by item-by-item comparisons of piece parts. There are many tradeoffs to consider.

Leveraging cloud as a part of a system can do more than merely provide a destination. **TransLattice leverages the multi-core, multithreaded processors of Intel commodity servers, sharding, business rules, polling engines, and sophisticated exception handling to create application infrastructure with far lower operational costs than traditionally managed application systems.** TAP's hybrid of on-premises and cloud deployment provides both local high performance and global access and resilience. Its initial offering is for Enterprise applications. For more details about how a cloud/on-premises hybrid system of appliances can support an enterprise database with better economics, please read on.

IN THIS ISSUE

The Hybrid Approach

The simultaneous business needs for high performance, guaranteed application access, and controlled costs require not just a hybrid local-and-cloud[1] approach, but a well thought-out hybrid system. The need to avoid common patterns of node failure drives an additional requirement for redundancy as an inherent feature – not something that is tacked on by one or more additional server instances.

The on-premises version of the *TransLattice Application Platform (TAP)* is a system of standard Intel 2U servers (nodes) bonded together with TransLattice's system management elements (i.e., its software) that take out many of the traditional costs for administration, stand-by systems, and traditional recovery processes. It is distributed by design, not by happenstance. There is no central node – and, thus, no single point of failure. Instead, polling engines rules (about business, location, and number of replicas, for example) and an index of exceptions, all shared between nodes, provide the required data management structures.

The TransLattice Application Platform

The TAP appliance is a server built on multi-core, multi-threaded x86-processor architecture, which is well suited to support the atomic nature of database operations. These often are mostly CRUD, (create, read, update, delete) – as contrasted with the workflows of database access. An on-premises appliance provides minimum response time and local data for analytics. Cloud-based instances provide remote redundancy and resiliency. So far, the story is prosaic. TransLattice's distributed intelligence and systemic autonomic functionality are what distinguishes its capabilities and economics from those of traditional platforms.

Each node provides storage, processing, and network capacity. It may be a physical appliance or it may be a utility computing instance. Both are subject to the rules implemented by the TAP *Cluster Hierarchy* (discussed on the next page). Each is deployed as a *share-nothing* architecture – but one that cooperates with its peers through shared metadata, a polling engine, and consistent rules about data durability and data placement. It is largely self-managing and uses dynamic addressing. The magic number for resilience is three appliances. For the polling approach to work effectively, four or more nodes are better. Each appliance includes an application container to allow existing software (including custom applications – anything written in *J2EE*) to leverage a cloud implementation. TransLattice expects that, in most cases, it is an I/O limitation that will invoke the need for additional hardware.

Each appliance has eight disk drives (2 TB SATA, 3.5" format) in a 2U rack mountable server. The Intel processor has 2-sockets, with a total of 4-cores. Three to four percent of the total storage capacity can be SSD, implemented on a PCI card. According to TransLattice, with a distributed system, a little SSD goes a long way.

Each TAP appliance costs $100,000. Early customers have found that TAP pays for itself in the first year of operations via infrastructure simplification from operational automation and a reduction in the number of application licenses that are needed, due to a denser infrastructure. Because it is both application server and database server, there are potentially fewer software layers to be managed and fewer copies of software to be licensed.

Licensing is per node, but utility and per-user pricing will also be available. Version 2 will be available in the first half of 2011.

TransLattice's competition comes from SANs and WAN optimizers and, at the other extreme, from enterprise DBMSs, such as Oracle's. SANs and WAN optimizers do not have the integrated database functionality needed to constitute a self-optimizing application. Traditional enterprise DBMSs have the database end but usually do not result in the simplification of infrastructure.

TransLattice has the policy mechanisms to maintain transactional integrity across all systems. This alone justifies a closer look. The balance of the system elements allows TAP to achieve system resilience, scale, and consistency.

TransLattice Software Elements

In TAP, completely automated *sharding*[2] gives redundancy and multiple data sources to speed rebuilds. *Polling* gives a rational basis to

[1] TransLattice, since its launch, has a partnership with Amazon *EC2*.

[2] Sharding by DBAs is a traditional and common database practice that is used to cut the unwieldy structures of large traditional databases down to a more easily manipulated size. It can be based on rows or columns (the latter is used for normalization).

orchestrate rebuilds for quickest return to functionality[3], and also provides a voting platform to identify aberrant data – be it merely stale or the result of bit rot. *Business and data placement rules* guide where and how many instances of data are kept. Because it is assumed that hardware will fail, a globally-distributed *exception index* guides ongoing re-optimization of the system.

Sharding

The imperative of keeping bits safe, due to the fragility of the bit itself, continues to be addressed best with replication. The challenge is to know what granularity should be replicated, and what is the minimum number of times to be sufficiently adequate. More copies means resilience, but the overhead can be punishing.

TransLattice shards groups of rows based on attributes, such as a primary key, rather than on a set of adjacent rows or based on a raw slice of storage capacity. This allows business policies to be more congruent with shards. Because TransLattice does it in the *access layer*, not the *application layer*, the data looks like one database to the application.

TAP uses different combinations of storage elements to store each data object redundantly. Thus, if a node or storage element fails, restoring redundancy involves very little work, which is distributed across the multiple nodes that hold the redundant shards.

Cluster Hierarchy

TAP supports the grouping of nodes in a structure called a *Cluster Hierarchy*. This allows nodes to be aligned and managed independently from the actual topography of physical nodes (appliances). Clusters can be at national, regional, city or local levels, which is useful for meeting governmental regulations. TransLattice presents them as a wheel, with circumferential rings representing policy tiers (with inheritance by outer levels). Therefore, local nodes would be governed by the local and national rules that apply. As this is an overlay approach to policy, other overlay policy structures could be developed to meet the needs of specific industry business processes.

Administrators manage nodes using the Cluster Hierarchy wheel as a tool. Its interface supports full drill down to the particulars of each node's status.

Data Placement Rules

Policies in this area are determined by governmental regulations,[4] resiliency, the need for immediate durability, and the intransigence of latency. *Location policy rules* dictate the placement of tables, or portions of tables, to meet national regulations. *Redundancy policy rules* specify how many replicas of each type of data must be stored.

TAP maintains distributed metadata about data placement on each node. The TAP system allocates storage across the nodes but within these policy constraints to optimize access. The system monitors access patterns. If the patterns change, the location of data can be automatically re-optimized, following the two sets of rules outlined above. It uses historic data and its system awareness of topography to determine the most cost-efficient data placement strategy.

Durability

For transactions, TransLattice provides full ACID (Atomicity, Consistency, Isolation, Durability) support. It allows *eventually-consistent operations*[5] on slow WAN links while ensuring durability locally. TAP policies allow the specification for certain classes of transactions to be less than ACID, i.e., completed asynchronously. ACID is required for payments processing – but ASAP (as soon as possible) may be just fine for utility processes and some updates. The automation in the system will assure that they are completed. That TAP can assure durability across geographies has high business value and is a big TransLattice differentiator.

TransLattice Self-Management

The TransLattice system manages redundancy in both normal and rebuild modes. It monitors the round-trip time of TCP/IP to respond to utilization status of underlying network links. If a node is empty, it will get filled. Since data is spread in multiple instances across multiple nodes, few data objects lose redundancy when a node fails. Because any rebuild can be shared between the many participating nodes that hold the needed data and because this

[3] This typically DBA task is automated in TAP. The quickness of the return to functionality is aided by the index of exceptions, discussed at length at the end of the section on polling algorithms on page 4.

[4] For example, some organizations require redundancy across geographies, such as three replicas on two continents. In other cases, regulations might require that business information must stay within national boundaries.

[5] Like asynchronous completion.

is orchestrated by the polling mechanisms, rebuild is quick.

Polling Algorithms

TransLattice offers polling as an approach that is determinative, but avoids the constraints of centralized management, therefore keeping the pyramid as flat as possible. A consensus protocol (*Paxos*) is integrated with a local write-ahead log mechanism to ensure that transactions that require durability obtain it before the commitment completes. The consensus protocol regulates all changes to the distributed database. Parallel propagation of changes is inherent to TransLattice's management process, so change is orderly and controlled.

If a processor fails, its in-process transactions are immediately redirected to another node. The consensus protocol makes sure no information is lost and that the transaction has been completed. Users, also, are redirected to another node. The Recovery Time Objective (RTO) for these users connected to the failing node is under two minutes, while users connected elsewhere in the system remain unaffected.

When hardware fails, the system will wait five minutes to make sure that it is really dead. Then the system, by polling, will replicate the information on the failed node from other shards to spare capacity in the system[6]. Over time, the system uses spare bandwidth to relocate the recovered shard to its optimal placement. The system keeps an index of exceptions to optimal placement – think of it as a "to do" list. By its nature, this exceptions index is self-pruning.

Security and other System Participation

The basis for authentication of the elements of the TransLattice system is a white list. Nodes carry a PKI certificate of their serial number. Communications between nodes is on a SSL-secured overlay network. Users and applications access the system through a *Service Entry Point (SEP)*. Node polling arbitrates which node will take over the SEP address of a failed node, depending on data and user location and the nature of the service contract involved.

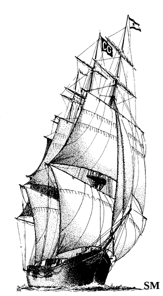
The underlying database uses a *multi-version concurrency control (MVCC)* mechanism for snapshot isolation and concurrency. A transaction sees the contents of the database as it existed when the transaction began. This consistent view can be used for backup. The system will recognize the backup destination and will configure the feed appropriately.

Conclusion

The TransLattice Application Platform provides an echo of Grid Computing but with more sophisticated control elements. It takes computing resources and weaves them together in a consistent system that TransLattice calls *Lattice Computing.* The name of the company is a meld of the concept of *transactions* with that of *mathematical lattices*.

The TransLattice approach goes far beyond building merely on best practices or perceived best-of-breed components. Yet, it uses seasoned elements. Data architects have been thinking about sharding, Byzantine generals[7], and swarm algorithms for decades. The sophisticated decision engines that guide Web experiences show that navigation can be a matter of customizing the destination, not of maps. TransLattice's distributed approach, and its management of problems by rule, rather than by physical resource, allows new tools to be brought to bear on traditional enterprise IT challenges.

The history of the Internet and other overlay networks, as well as high-performance computing clustering, has long provided new paradigms with less capital-intensive costs. TransLattice uses these approaches to support enterprise applications (i.e. the enterprise's ravenous *Audreys*) with a system of intravenous-like support that will keep everyone happy. Consider the future of your enterprise and how you wish to leverage your information. TransLattice provides an alternate strategy to the inadequacies of stove-pipe-style divide-and-conquer.



[6] Remember, the transaction has already been redirected to and completed by another processor.

[7] Fault tolerance in distributed systems is a challenge often compared to that of the Byzantine armed forces, where multiple kinds of failure may have unpredictable results. This can be countered by polling to determine the exact nature of the failure.

About The Clipper Group, Inc.

The Clipper Group, Inc., is an independent consulting firm specializing in acquisition decisions and strategic advice regarding complex, enterprise-class information technologies. Our team of industry professionals averages more than 25 years of real-world experience. A team of staff consultants augments our capabilities, with significant experience across a broad spectrum of applications and environments.

- ***The Clipper Group can be reached at 781-235-0085 and found on the web at www.clipper.com.***

About the Author

Anne MacFarland **is a Senior Contributing Analyst for The Clipper Group.** Ms. MacFarland specializes in strategic business solutions offered by enterprise systems, software, and storage vendors, in trends in enterprise systems and networks, and in explaining these trends and the underlying technologies in simple business terms. She joined The Clipper Group in 2001 after a long career in library systems, business archives, consulting, research, and freelance writing. Ms. MacFarland earned a Bachelor of Arts degree from Cornell University, where she was a College Scholar, and a Masters of Library Science from Southern Connecticut State University.

- ***Reach Anne MacFarland via e-mail at Anne.MacFarland@clipper.com or at 781-235-0085 Ext. 128. (Please dial "128" when you hear the automated attendant.)***

Regarding Trademarks and Service Marks

The Clipper Group Navigator, **The Clipper Group Explorer**, **The Clipper Group Observer**, **The Clipper Group** *Captain's Log*, **The Clipper Group Voyager**, **Clipper Notes**, and *"clipper.com"* are trademarks of The Clipper Group, Inc., and the clipper ship drawings, *"Navigating Information Technology Horizons"*, and *"teraproductivity"* are service marks of The Clipper Group, Inc. The Clipper Group, Inc., reserves all rights regarding its trademarks and service marks. All other trademarks, etc., belong to their respective owners.

Disclosures

Officers and/or employees of The Clipper Group may own as individuals, directly or indirectly, shares in one or more companies discussed in this bulletin. Company policy prohibits any officer or employee from holding more than one percent of the outstanding shares of any company covered by The Clipper Group. The Clipper Group, Inc., has no such equity holdings.

After publication of a bulletin on *clipper.com*, The Clipper Group offers all vendors and users the opportunity to license its publications for a fee, since linking to Clipper's web pages, posting of Clipper documents on other's websites, and printing of hard-copy reprints is not allowed without payment of related fee(s). Less than half of our publications are licensed in this way. In addition, analysts regularly receive briefings from many vendors. Occasionally, Clipper analysts' travel and/or lodging expenses and/or conference fees have been subsidized by a vendor, in order to participate in briefings. The Clipper Group does not charge any professional fees to participate in these information-gathering events. In addition, some vendors sometime provide binders, USB drives containing presentations, and other conference-related paraphernalia to Clipper's analysts.

Regarding the Information in this Issue

The Clipper Group believes the information included in this report to be accurate. Data has been received from a variety of sources, which we believe to be reliable, including manufacturers, distributors, or users of the products discussed herein. The Clipper Group, Inc., cannot be held responsible for any consequential damages resulting from the application of information or opinions contained in this report.