The TransLattice Elastic Database (TED)

Designed for Globally Distributed Transaction Processing

By Anne MacFarland

The Situation

Transaction processing in a global economy comes with some stringent parameters. Transactions must be processed and committed wherever they arise and whenever they occur. The consequences to related systems, like inventory management, must be part of the transaction. To support a prompt, full ACID (Atomic, Consistent, Isolated, Durable) transaction commit, the data involved must be co-located, properly governed, and resilient.

Because technology elements do fail, and can be attacked, the system that manages the data must be inherently responsive to such events. It must be able to rebuild data to keep itself whole as a coherent structure without leveraging a single point of failure, such as a master node. In short, it must support the traditionally incompatible goals of both scale and performance, and minimize the overhead of chattiness or the burden of too many accessories such as indices.

In the context of mobile phone-induced transactions, the efficiencies of traditional batch processing become insufficient. Local replication of data is still a requirement, but centralization of data has less business value if – and this is a big if – the data can be protected and governed wherever it sits "as if it were local." This transaction data use case challenge cannot be met just by replication, or just by a remote copy, particularly when the scale of data transactions is large. With social and event software generating new sources of transactions, a new way to handle the flood of critical business and process documentation is needed.

The TransLattice Approach

Late in 2010 TransLattice, a company based in Santa Clara, CA, launched a solution, the TransLattice Application Platform (TAP), which deftly addressed many of the challenges outlined above. This year, in response to a growing customer preference for database-only solutions, they have launched the TransLattice Elastic Database (TED). It can run on appliances, of course, but it can also run on virtual machines (something large traditional databases cannot do) or on cloud architectures - or on all three - as long as the nodes involved in all three options are roughly comparable in resource capacities. While there is no architectural limit to the number of nodes that can be supported, the current installed base tops out at 16 nodes, each of which holds up to 12 Terabytes (TBs). This gives TransLattice customers a huge range of deployment options, and also decreases the up-front costs of the solution.

Sustainable Elasticity

Sustainable elasticity is what we all want in our muscles, skin – and even, if it could happen, in our

82 Pilgrim Road · Wellesley MA · 02481-2455
781-710-9291 annem@macfarlandconsulting.com

bones. It is also what we want in our database operations. Traditionally, distributing databases over distance was a challenge made necessary by disasters, and power or hardware failures. Many good solutions were designed but they were designed with a keen awareness of limited resources. In contrast, TransLattice is a 21st century organization built on 21st century connectivity and bandwidth assumptions. Distributed operations are a core mantra and the basis for all systemic capabilities. It was designed as a cellular architecture distributed system.

TransLattice Operations

In TransLattice systems, tables are split into thousands of logical segments called shards. These shards are placed to meet business requirements for data redundancy (how many at how many locations) and data location (where and where not) – qualities that vary by geography, industry and use case. This allows administrators to comply with business requirements and business model evolution by making changes independent of data-level attributes, avoiding a passel of trouble.

TransLattice Elements

Shards and Nodes in a Single System

TransLattice shards database tables into chunks, often based on a master key. TransLattice's patent-pending software recognizes which elements are needed for common business transactions and where there are dependencies. TransLattice is not a fan of federation of data; federation introduces interpretive or translational elements that can add latency and clutter the intentionally lean processes of data propagation and node rebuilds. TransLattice encourages users to keep a single system approach.

Data Placement

Placement of data is determined by many factors. The first factor affecting placement is historical access patterns, which place the data close to where it will be used. The second factor is regulatory compliance and business policies. This supports compliance to many specific government and privacy regulations. A third factor is a data-specific assessment of how much redundancy is needed. The overlay of layers of policies, with inheritance, results in a rich and specific topography of data placement without undue redundancy. Finally, in what TransLattice calls *deterministic randomness* – all factors being equal, the data is sprinkled across the cluster. When combined with TransLattice software's ability to learn from transactional patterns (machine learning), it is easy for admins to add or change policies to meet regulatory requirements.

Intelligent Networking

Bandwidth, traffic and the speed of connections between nodes is monitored at the system level. These metrics support optimization of data placement and query execution plans. There are numerous factors that can and should be used to optimize transaction processing in an inherently distributed environment – the origin of the query, the placement of instances of the data that is relevant to the query, and how the data is related, as well as "network load and capacity." The system can dispatch sub-queries to other nodes to improve throughput – and reduce the results to the node that initiated the query, reducing unnecessary hops and response time.

Porting

For existing databases, TransLattice works with tools for interoperability and ETL (Extract, Transform, Load) to move data easily into and out of TED. CTO Mike Lyle says most TAP and TED use today is with new applications. TED supports ANSI-standard SQL, which greatly simplifies porting applications to TED. Applications written for an abstraction layer, such as Hibernate or JPA, can often run on TED unchanged.

Access Layer

Users and applications access TED through Service Entry Points (SEPs), just as they did with TAP. Nodes arbitrate control of the SEPs within a subnet, assuring seamless reconnection, should a node fail, with other nodes in the network segment. The SEPs and subnet also support traditional traffic load balancing. At a node level, TransLattice nodes broadcast actions and changes to one another. They recognize transaction dependencies, and, more usefully, which transactions are NOT dependent on one another.

TAK Change Management Tools

A TransLattice Automation Kit (TAK) helps administrators handle operations at a systemic level (adding nodes, evolving placement policies, configuring database authentication, etc.). TAK also automates the deployment and configuration of nodes. This is useful when setting up lots of nodes, as is needed when deploying a test environment.

Sustainability

The virtue of this approach lies not just in the specificity of placement that can be achieved – think, for example, of about the data needed for international projects – but in the ability to evolve policies as governmental regulations are tweaked, as happens regularly. TransLattice provides a blend of

- local functionality (alerts, rebuilds) at the cluster level,
- ongoing changes, such as updates to data policies somewhat asynchronously, and
- an underlying coherent system image, supported by contributions of changes from throughout the TransLattice environment.

This creates support for both high priority and routine operations. It also supports business evolution. It is not a task or batch-centric cadence but an ongoing, adaptive process – with support for ACID transactions. Without the ACID support, the offering would be merely interesting.

The challenge of this kind of system is to support both the comprehensive coherence of the whole and precisely targeted local functionality. TransLattice supports the comprehensive coherence with a self-correcting, self-updating system image maintained by nodes that recognize and communicate changes and by a consensus protocol, based on PAXOS, that runs on all nodes. These keep the system whole *and* real - but not completely real-time, though the quick-twitch of ACID transactions is supported on all

MACFARLAND CONSULTING
82 Pilgrim Road · Wellesley MA · 02481-2455
781-710-9291 annem@macfarlandconsulting.com

participating nodes. The policy layers can be evolved in accordance with the cadence of regulatory change. While atomic transactions are fully supported, TransLattice is less suitable for real-time applications such as those that support trading desks, GPS- focused functionalities where the rate of change in data is high, or scientific processes where timing is critical.

TAP and TED Today

The launch of TED reflects how technology is leveraged by business. TAP includes an application server, web server, and unified management, and is focused on achieving scale and precision while keeping systemic risk minimized. For customers primarily interested in enhancing the database and not changing the application environment, TED fits those requirements well. Both products sport advanced features, such as the ability to recognize which transactions have interlocking dependencies and which do not. They can then classify transactions to optimize throughput and minimize latency while keeping high availability intact. They can do the same for queries.

Within the well-understood concept of clusters with self-healing nodes, TransLattice introduces not just co-operation (other clusters support this), but the ability to seek and find the nearest instance of needed data to rebuild a failed node without reference to a master node. There is a map of the full extent of business information that all can refer to. It is, for all its distributed nature, one database. All data can be used – but the topography is optimized for transaction performance. In this area, TransLattice also has the software to easily add additional nodes to support access to suddenly-hot data. Usage statistics and a predictive data engine place data on nodes where usage will be greatest. The system recognizes frequently accessed data and can replicate it to additional nodes in the cluster to avoid bottlenecks. Generally speaking the data is only replicated to a small number of nodes closest to end users.

Conclusion

Most businesses are on the entrance ramp to a multi-lane superhighway brimming with data of many kinds. As with any superhighway, achieving and maintaining the speed needed to survive and maneuver is obviously essential. Excess baggage is obviously an encumbrance. Application efficiency (think miles per gallon) is obviously a good thing. TransLattice, both in TAP and now in TED, supports these requirements and gives organizations the ability to think global and act local. TransLattice's early customer base in financial services and government has indicated that the ability to achieve scale and compliance in a world of escalating numbers and varieties of transactions has great value. Consider the probable future of transaction processing and how TransLattice can make it a happy prospect.

MACFARLAND CONSULTING
82 Pilgrim Road · Wellesley MA · 02481-2455
781-710-9291 annem@macfarlandconsulting.com