

Heterogeneous Data Replication: Cost-Effective Information Leverage

Merv Adrian, Principal, IT Market Strategy
www.itmarketstrategy.com

An Asset Approach to the Value of Data

Data is the lifeblood of organizations. But while it is used intensely, and grows more dramatically every year, at substantial cost for storage and processing, few organizations take an asset-based, comprehensive approach to it. It drives too much cost because of ungoverned redundant proliferation, is poorly synchronized across applications creating accuracy and timeliness issues, often lacks comprehensive availability policies that span all important data instead of specific files, databases or application products, and consumes far too much time and too many skills due to the lack of a designed, policy-driven approach to its management.

Compare this asset model of data to that of other corporate assets. Few organizations keep multiple unused "copies" of trucks as they do with data that is stored unchanged in multiple locations, creating wasteful, expensive redundancy. Few keep multiple stockpiles of slightly different versions of the raw materials they use for finished goods unless there are price or efficiency reasons for doing so, but they store multiple differing unsynchronized customer records in multiple purchased software applications with little thought to keeping them aligned. Few maintain multiple local office locations that they allow one employee to occupy and modify at will, even though they have access to the central one, but they likely have duplicative data marts and spreadsheets on employees' laptops that cause synchronization and timeliness issues.

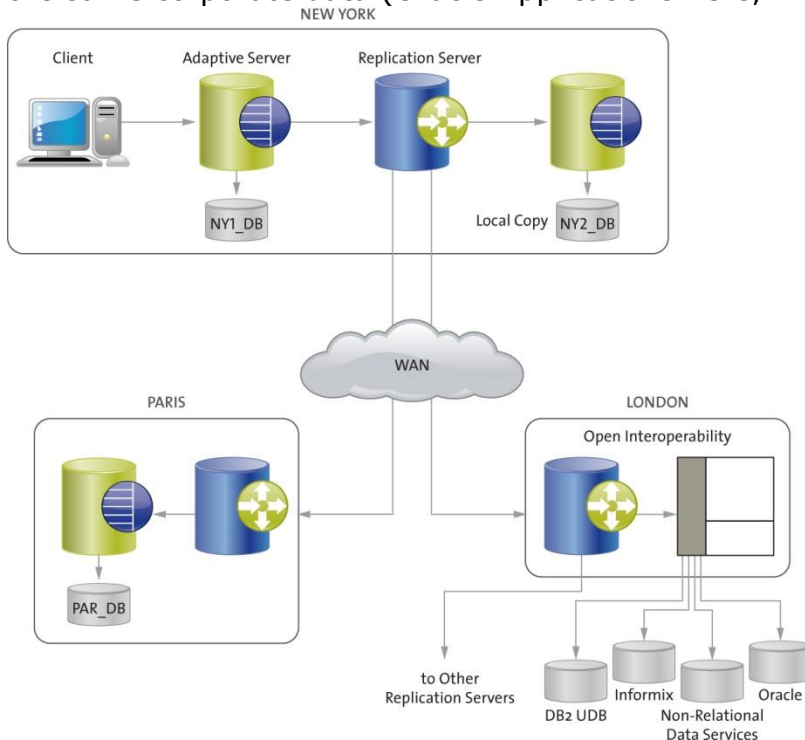
Effective data integration strategies help companies reduce information-related costs, manage a variety of data risks, and improve business agility. Data replication is one of the key technologies in a market sized by IDC as driving US\$2B in revenue for decision support (BI), \$5B for data profiling and cleansing, and \$11B in operational integration (data migration, replication and synchronization, and application upgrade.) Since its introduction of Replication Server in 1991, Sybase has been a leader in data replication, with over 2800 customers, and recent steadily accelerating growth. One key driver of this uptick, and a driver for interest elsewhere in the industry (such as Oracle's acquisition of GoldenGate in 2009) has been heterogeneous replication; while only 15% of Sybase's revenue is derived from non-Sybase data today, this part of its portfolio is growing rapidly as customers and prospects recognize the value of a single, centralized, model-driven, database-independent approach to replicating data. In this white paper, we examine some of the use cases and considerations in heterogeneous data replication, drawn from Sybase customer interviews and technical briefings.

Use Cases Demonstrate Replication's Flexibility

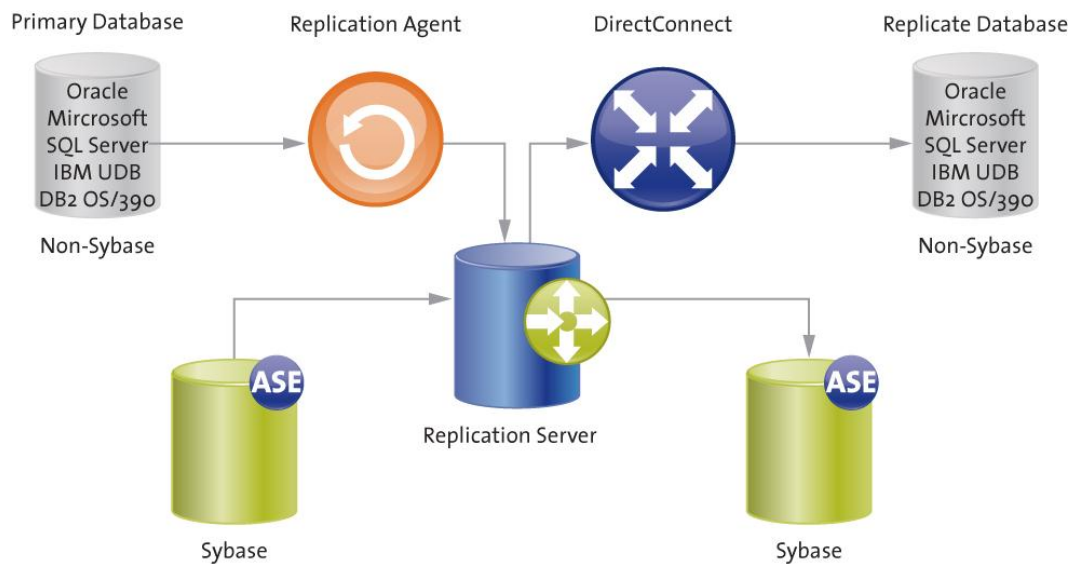
Most high-level database offerings used for mission-critical applications provide some sort of homogeneous replication product for their own data, but handling other source and target databases is a different matter entirely. Many important challenges and opportunities are left unmet because of this. Some examples:

Supporting Business Intelligence (BI) across multiple databases. This has enormous value in shops that have selected a preferred platform for analytics but need to load data from disparate sources. At Spice Telecom in India, for example, Sybase's Oracle Replication Option is used for real-time updates from Oracle to a Sybase IQ analytic database. In such cases, the issue is timeliness, the need for up-to-date data and analytic performance justifying the additional data copy. In some cases, replication is used to offload and consolidate operational reporting to preserve the performance of multiple production systems; the goal of single-system mixed workloads handling both transactions and reporting is often far too costly by comparison to the use of an additional engine with its own store.

Enabling global, but local, operation. As companies "go global," their business applications must follow suit. Localizing applications – from data encoding to language support to varying regulatory, privacy and reporting requirements – enables distributed business operations and drives growing demand for data distribution. These distributed applications may work on subsetted, or otherwise transformed data that must be kept current – or even exact copies that have an equal need to be synchronized. Replication automatically maintains up-to-date data among distributed databases for these scenarios – and can handle configurations where another geography, or a subsidiary, is using a different application against the same corporate data (Oracle Applications here, Infor there, for example.)



Enabling heterogeneous multi-platform business processes. Even organizations that build their own applications may have created them on different databases due to tooling, language, or platform issues – or because they acquired another company and adopted its system. Order taking may be deployed on a Microsoft SQL Server system, feeding a manufacturing system built on IBM DB2 with financial accounting on a Sybase system. A single, model-built environment that operates across these multiple endpoints is invaluable. At a large auto rental firm, DB2-based reporting is fed by an Oracle system that tracks availability in franchisee location, using Replication Server to replace brittle old hand-coded scripts. ¹



Additional standby databases for different workloads. Replication is not always one-to-one; one-to-many and many-to-many are very frequent scenarios. In fact, Sybase reports that 85% of its Replication Server customers use its multi-site availability feature. One database may be used for disaster recovery, while another is in constant use for reporting. A centralized replication architecture (as opposed to a point-to-point one, as discussed below) permits this flexibility with minimal complexity. Paradoxically, this may lead to a reduction in data copies: for some firms, the opportunity to consolidate multiple existing operational reporting systems onto a single platform can lead to fewer copies of the data.

Migrating to new platforms. There are several variants of this case, which may have a short duration but high exposure. Moving to a new version of a database, transferring data from one application to another, changing the database upon which an application is hosted – all are possibilities. A generalized tool that is

¹ In The Data Warehouse Institute's The State Of Operational Data Integration, 336 Respondents reported "regularly" syncing customer data (36%), product data (34%.) The same report indicated that non-BI/DW data integration has risen from 19% to 49% in 4 years.

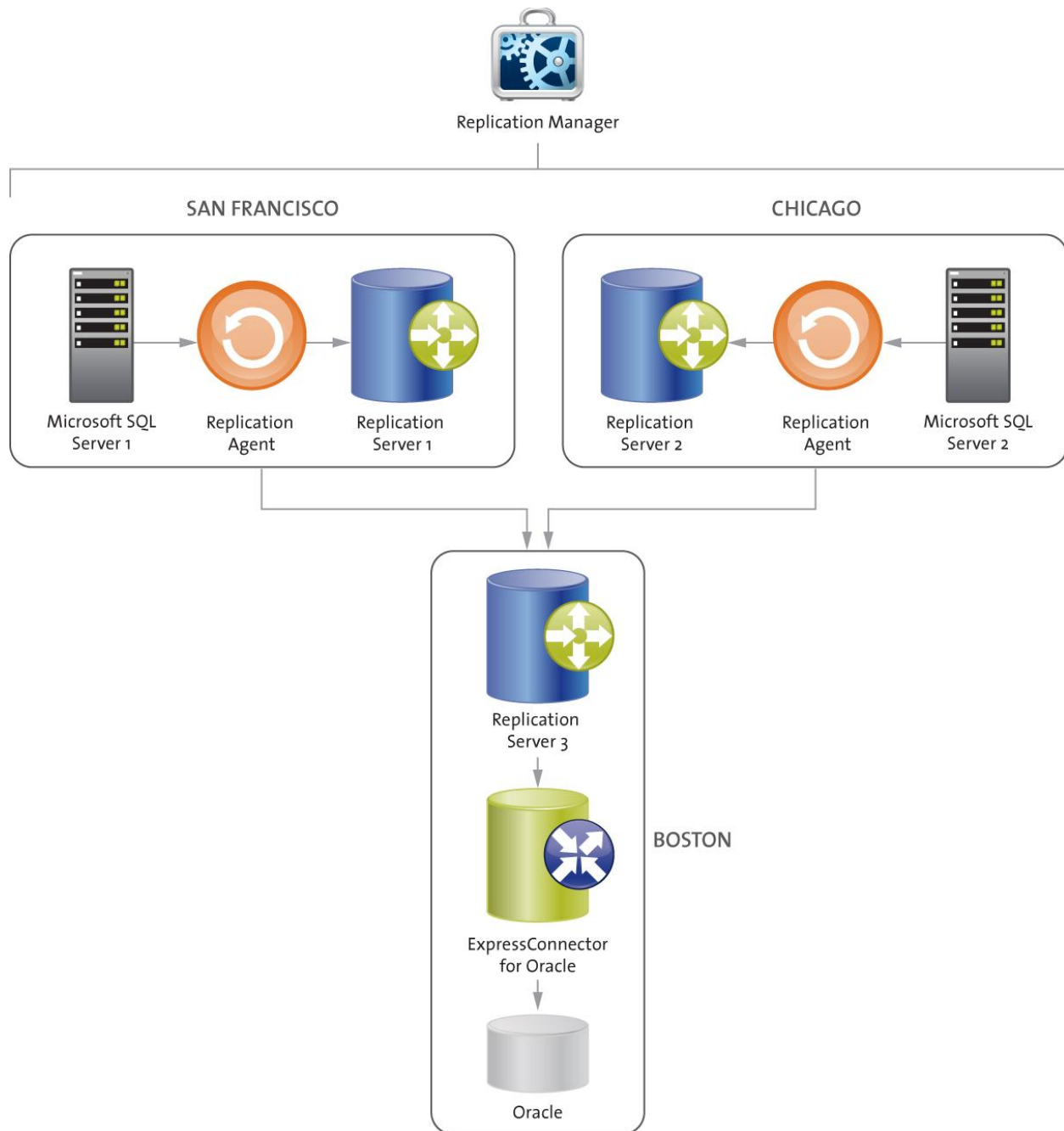
flexible and doesn't require a new purchase or expensive consultant-driven process involving hand-coded scripts can save time and money.

Design, Not Default

How are these opportunities/challenges met? In many organizations, if they are tackled at all, the solution is hand-coded via scripts or simple DBMS exports. This is a far from adequate approach; although architectural considerations are relatively straightforward for point-to-point connections, they quickly become overwhelming with multiple sources and endpoints. Moreover, if different solutions are used between differing endpoints, the number of skills required makes maintenance challenging and requires multiple skill sets that must be kept fresh. Documentation becomes a challenge, and the solutions are fragile and brittle as a result.

A proactive, design-based approach has considerable power over ad hoc, reactive replication strategies. For example, Sybase's Power Designer provides tools to model the environment, specify the desired information flows, and generate the necessary code directly from the model. It can reverse engineer Replication Server System Databases (RSSDs, where the metadata is kept) for existing replication server environments, so prior data flows may be imported into the design environment for model-driven extensions and enhancement, task refactoring and optimization. PowerDesigner generates the necessary DDL code using UML modeling, process modeling and XML modeling as appropriate. Models are checked for consistency to identify errors.

For many organizations, replication can provide an effective, value-based entry point for master data management (MDM) or other metadata efforts associated with data governance or data quality projects. Because it preserves the information in the model, it becomes a document usable as such efforts become richer and more comprehensive, and may become a tool in the hands of architects tackling more systematic, enterprise-wide data efforts, allowing them to reuse the pragmatic projects designed to serve immediate business needs.



Architecture Considerations

Many factors drive an organization's approach to the architecture used for heterogeneous data replication. Most firms have complex topologies, multiple applications and data stores, and most numerous *de facto* choices available. By far, the most important driver of architecture tends to be inertia: the tools used are the ones already available. This can be a mistake because the tools impose limitations, require skills and maintenance of multiple environments, and often have associated

costs that can be substantial – especially if multiple tools are in use that can do the same thing. Buyers should consider the business needs first, make architecture decisions accordingly, and be willing to do the hard work of migrating off the familiar, in-place product if another will perform the same functions. Some of the dimensions to consider include:

Volume pattern: bulk or near real-time load? For some business needs, the batch or periodic move of large volumes may be sufficient. Native database tools, scripts and system data movement utilities may be in place for such activity. More real-time approaches, typically called changed data capture (CDC), propagate changes when they occur – these are most often seen in availability scenarios but are applicable in the other use cases discussed above as well.² Replication tools that can do both, especially those that can use native APIs for bulk load, are the better choice.

Point-to-point or centralized? Some shops have acquired point-to-point solutions (sometimes several of them) to move data between sources. These can quickly become unwieldy in their complexity, creating a crazy quilt of crisscrossing connections, and multiple data streams that might well allow consolidation. The multiplicative sources/targets phenomenon can become unmanageable rapidly, and the variety of offerings used may also create a challenge in monitoring and skills, as well as potential conflicts as separate connections are built to or from the same data. A single point of design, deployment, skills and management able to support many sources and targets is a far better alternative. Acquisition and maintenance costs for additional targets and sources are likely to be lower than paying for multiple tools. Powerful, (separate) engine for batch processing
Lightweight, in-line for continuous, low volume

Trigger or log-based? Replication from the database, as opposed to using application-to-application is a particularly useful way to avoid invasive processes inside applications; changes are handled at the data level, not the logical level. CDC may be implemented with triggers in source databases that support triggers, but this approach can have a substantial impact on performance; it creates overhead and causes scalability issues. The alternative is to read the transactions from the log – the source database is already writing that log and no additional impact is created on the engine. A further benefit comes with the replication of SQL statements for execution in the target DBMS (instead of row-by-row changes); this is a feature of Sybase’s most recent update.

Selection Issues

There are numerous criteria that inform the selection of a replication solution, and a full discussion would be beyond the scope of this paper. But there are some key guiding issues to keep in mind:

² As far back as 2008, IDC reported that “dynamic data movement” was growing 5 times faster than as “staged data movement.” [Worldwide Data Integration and Access Software 2007 Vendor Shares.](#)

Focus on business benefits. Many organizations take a “boil the ocean” approach, tackling ambitious plans to remake their data architecture before they have solved the simpler problems with the immediate business benefit. Keep it simple; solve a real problem like this listed above. Identify the business value of the solution and determine measurements that will validate it. And measure after the solution is in place to assure that the benefits are realized; those successes will make it easier to justify the next project.

Get the architecture right. Even though it may not lead, architecture is critical. Look for reusability, efficiency and performance. Use solutions that leverage common skills like design as opposed to niche ones like familiarity with a single source or target’s syntax. Don’t over-engineer for ultra-low latency if your business need is “close of business yesterday;” if the design is sound, the performance profile of the technical implementation may be changed at a later time.

Create an inventory of key data and understand its value – build a solution that will work with the key heterogeneous sources and targets for cross-organization and cross-process needs. Some sources and targets may not be supported by your chosen solution if legacy systems have been in place for a long time, but look for the product that will handle the majority of them.

Document required datatype support and mapping. This is a stumbling block for organizations that discover too late that their inventory was not “technical enough.” Don’t assume routine transformation from one product to another, be sure that your selected product can handle your needs. Another example of this issue is support for character sets – critical for AP operations. While English documentation may be acceptable for overseas operation, limiting support to western character sets may not be. Even as subtle an issue as the language of error messages can be a major issue, requiring an entirely different skill set on the ground in global operations centers.

Build models. Store metadata. Reuse. The replication process can be the basis of a broader information modeling exercise if done right. Beginning with the objective contains the problem and makes delivery on schedule possible and measurable. Modeling makes it extensible and provides leverage for later projects.

Outsource transitory work. Some projects are one-time moves; version migrations, conversions to new standard database platforms are over when they’re done. You may not need ongoing skills.

For ongoing needs, consider a competency center. Replication on a continuous basis is likely to be a generally needed skill; if this is your situation then staff accordingly, develop and sustain the skills investment and document what’s needed.