ProgresSmart

A Powerful Vision for Information Asset Management

September 23, 2005

By Dan Tanner

> *A rich, new, competitive technology market is developing around classifying data automatically based on policy. Many of the companies in this market will offer software that automates data placement on a tiered storage infrastructure, in support of the "appropriate placement" mandate of the Information Lifecycle Management (ILM) initiative.*
>
> *But perhaps the software should look beyond merely helping IT administrators meet a service level agreement (SLA) and focus instead on ensuring that data is always handled according to its value to the business. Not only would the ILM function be a byproduct, but also it may be possible to increase data's value beyond original intents and purposes.*
>
> *It all depends on having the right vision and implementation. This paper is about Njini, a company that seems to have embarked on the right track.*

IT managers have lately been busily implementing ILM — whatever they think that means — because they're being pressured into committing to service level agreements. According to the Storage Networking Industry Association (SNIA), "Information Lifecycle Management is comprised of the policies, processes, practices, and tools used to align the business value of information with the most appropriate and cost effective IT infrastructure" and it does so "from the time information is conceived through its final disposition. Information is aligned with business requirements through management policies and service levels associated with applications, metadata, and data."

In practice, attempts to implement ILM have come to rest on two pillars: (1) data classification and (2) a tiered storage infrastructure. And, experience is rapidly teaching a difficult lesson: On the one hand, plenty of good components for storage tiers are available, and architecting and implementing a tiered storage infrastructure is a doable one-time (or at most, infrequent) task. And, on the other hand, data classification is a difficult thing to do, and is necessarily on-going.

If enterprise data isn't classified properly, its assignment to storage in the ILM will be wrong and wasteful at best, and harmful at worst. And if it is, the IT department may meet its SLA mandate while losing sight of the data's business value in the shuffle. ILM, even if it is "achieved" has the potential to merely shift an IT department's workload from storage management to file management.

Introducing Njini

Njini is a company that understands the value of data to businesses. The Njini approach is to provide a level of file categorization (ability to identify different files types, characteristics, etc) and then for the organization implementing ILM to classify its data as critical, sensitive, non-critical, and so on.

By creating an information metadata index at what it terms the "identity layer" of the data at its origination, Njini software both facilitates data use manipulations that can enhance the data's value and in the process optimizes storage use. Njini software deals dynamically with data whose value can rise or fall. It does so using object-oriented techniques.

ILM is supposed to slash the huge recurring costs that enterprises bear in trying to somehow ensure that classes of data get the treatment they need on their IT storage infrastructure. That is supposed to be done by policies applied to data by class. But existing file systems provide scant metadata (data about the data) "handles" with which to classify data for assignment to storage, let alone track files through business processes. That is where Njini has engineered a departure, by treating files as "objects" and so not only endowing the files with additional metadata that is up to the task, but which also carries with it the inheritance properties that can make object-based systems so powerfully useful.

The ENGINE at the heart of Information Asset Management

njiniENGINE lets enterprises take a "content aware" approach to file management. This "information at the identity layer" approach is used for the core data management function of njiniENGINE which, based on their business value at their origin, automatically characterizes semi-structured files (i.e. office productivity documents like text messages, papers, spreadsheets and presentations — these semi-structured files can be searched but not sorted) and unstructured files (i.e. bit-mapped files of the type used for graphics, audio and video — which can be neither sorted nor searched) and then manages changes in objects' business value throughout the data object's lifecycle. njiniENGINE is core Njini technology and part of the company's initial June 2005 product release. The extensible meta-data extracted by njiniENGINE is then used by the Njini modules to map business policies to unstructured data.

njiniENGINE contains a database scalable to hundreds of millions of records to store configuration data and content metadata. njiniENGINE components are integrated with Microsoft Active Directory and other directory services and map users onto Njini profiles. "HELPERS" inside njiniENGINE are used to extract extended meta-data from unstructured data. Examples of such metadata are user identity information (UID), and metadata separated from the file system maintained in Microsoft Office and Adobe PDF files.

Njini believes that in the future its technology can also be applied to structured files (i.e. database types that are amenable to being both sorted and searched).

Also part of the Njini product rollout is njiniENCOUNT, one of four announced modules in its Information Asset Management (IAM) suite that run on top of njiniENGINE. While a number of software products remove file duplicates to minimize storage consumption, shorten backup times, and eliminate potential version conflicts, they tend to do so based on comparatively primitive algorithms. Namely, they usually retain only an archive copy (for data that is known to be unchanging) or an active file and a backup copy (for data that is expected to change). The more advanced njiniENCOUNT algorithm determines — based on a data object's business value — the optimal number of copies to maintain, for placement and usage according to needs. The number can

change to meet changing requirements during the data lifecycle. njiniENCOUNT stores user identity information on all data files and reports to the njiniENGINE database on all existing unstructured data.

Njini has announced plans to ship njiniENROLL, which automates data object placement on the storage infrastructure in accordance with enterprise specified policies, by the end of 2005. njiniENROLL is the part of the Njini IAM suite that does ILM, placing data objects within a database system, e-mail system, or appropriate storage tier(s). ILM contains hierarchal storage management (HSM) as a subset, but ILM has more to offer. HSM is a simple algorithm for moving files from more to less expensive storage as the data ages.

HSM has been around for years in the mainframe computer environment, with data movement's typical order going from memory to disk to tape. But simple HSM has three major drawbacks: (1) when HSM was designed, it was the capital cost of storage that mattered most, while today operating cost for storage usually swamps capital cost by about an order of magnitude, (2) HSM is usually non-transparent, meaning that migrated files are "lost" in terms of connectivity to applications and visibility to users and require IT department intervention for recovery, and (3) HSM is usually a static one-way migration policy that can only take files "downstream" in the storage infrastructure.

Njini appreciates that old data can suddenly take a quantum leap in value — for example, when its production is called for in a legal action — and therefore implements ILM functionality within njiniENROLL. That is, data object movements are transparent to applications and users alike and dynamic in terms of direction up or down the storage infrastructure tiers, with manual IT department involvement minimized or, if possible, eliminated. Additionally, njiniENROLL policies are dynamic, and an enterprise can change them at any time, a critical capability both for serving new business priorities and processes, and for ability to conform to new or changed regulations to which an enterprise may be subject.

Early in 2006, Njini plans to ship its njiniENFORCE module, for policy-directed retention or deletion based on compliance rules imposed on an enterprise from outside.

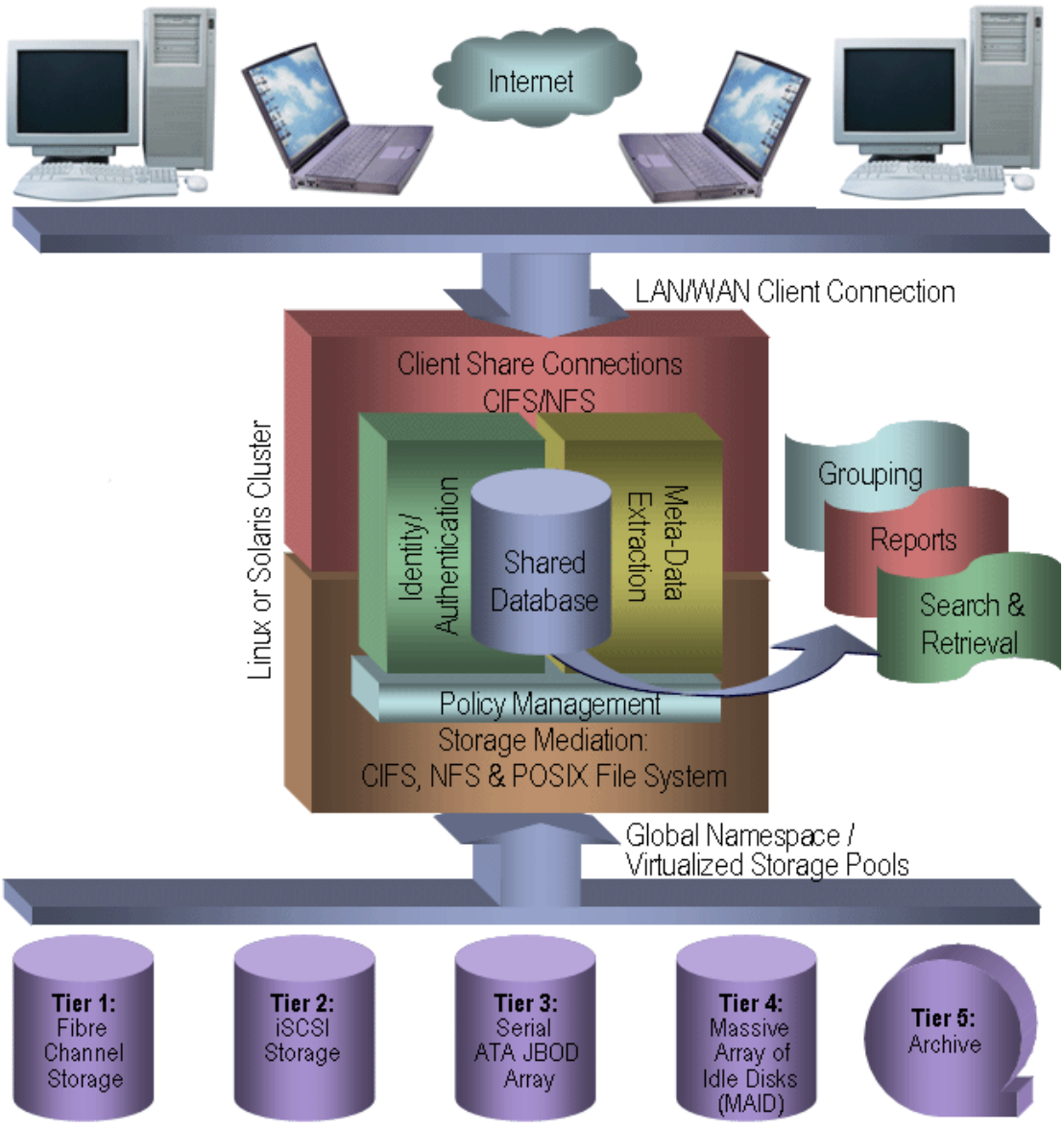
The final module in the Njini IAM suite so far announced (but with delivery date as yet undetermined) is njiniENQUIRE, a tool for searching and/or grouping data objects. The capability will be based on the rich set of object metadata that njiniENGINE creates, and should enhance enterprise productivity by providing easy access to relevant and related information. Importantly, by using the metadata and not the data to achieve this functionality, njiniENQUIRE should not conflict with security policies or products, file compression or data reduction software, or allow "fishing expeditions". And, because of Njini object-orientation, not only instances, but also interrelationships among enterprise information assets can be discovered and exploited.

Overview of the in-band Njini implementation

"In-band" (i.e., in the data path) architecture is central to Njini design and its present and planned functionality. It is because all data must pass through njiniENGINE that the software can gather information about the data and create metadata before the data is

stored. The architecture should also allow Njini to develop additional IAM functionality modules as well as enhance existing ones nondisruptively.

Figure 1: Njini Architecture



Source: ProgresSmart, August 2005

The "in-band" scalability and robustness features of the Njini architecture: All data and control information passes through a shared (replicate) database and resides in a single virtualized storage pool of diverse storage types with global namespace. The central block in the diagram represents one of what can be one of many computers in a Linux cluster with remote direct memory access (RDMA) connections linking multiple njiniENGINEs, and Njini databases. Also shared are client connections and storage connections. Any storage tier structure can be accommodated.

In implementation, njiniENGINE sits on dedicated Solaris or Linux server blades, with the Information Asset Modules (IAMs) running atop njiniENGINE. Client systems can

run either CIFS or NFS file system. The server blades running Njini connect to all storage tiers and types, and support CIFS, NFS, or any POSIX-compliant file system. The connected storage types and tiers can be quite diverse: high-end to serial ATA disks, archive storage, even the emerging multiple array of idle disk (MAID) tier. As noted above, Njini ILM is fully transparent, and that's because of two njiniENGINE attributes: (1) njiniENGINES can be linked together via remote direct memory access (RDMA), and (2) all of the storage is taken into a global namespace virtualized pool that is seen by the unified Njini database.

Additionally, the RDMA linkage capability should provide great scalability (purveyors of out-of-band systems typically claim that in-band designs are less scalable) and robustness (avoiding a single point of failure and providing operational failover).

ProgresSmart Conclusions

Njini has an advanced and extensive vision. Quite clearly, the target market is Fortune 1000 organizations. These are the organizations most severely burdened by the phenomenal growth of unstructured data — growth that can only be expected to accelerate with the worldwide growth in bandwidth and appetite for more, richer media. These same organizations are also most frequently the targets of complex and demanding (and often changing) regulations.

Most such organizations are jumping onto the ILM bandwagon. But a cursory glance at IT exposition agendas and tutorial schedules quickly reveals that while storage vendors cover offerings for tiered storage, the big stumbling block is data classification. In response, a handful of new-companies have entered the market, becoming known as "information classification and management" (ICM), Njini among them.

Not having ICM technology in-house — and possibly sitting on the sidelines to see which of the new market entrants succeed and perhaps either market the most successful's technology on an OEM basis, or simply acquire that company — the large storage vendors are publicly maintaining that data classification is not a product, but rather the set of "management policies and service levels" referred to in the SNIA ILM definition.

Large IT services organizations might also bid to provide the SNIA-defined data classification tasks. The work must be done, but at what price? Product pricing has yet to be established in this new field, and only time will tell whether pricing will be based on "user pain" or become competitive.

ProgresSmart concludes that Njini is in the right market at the right time, and has the right ideas, and only needs to score with a few top-flight customer success stories. And, there should be some willing customers. Although the enterprises in the target group tend to be late adopters and among the most risk-adverse, they should be motivated to try Njini weighed against the risk of doing nothing and drowning in a file management sea of their own making.

ProgresSmart also applauds the Njini design for its commonsense approach. ProgresSmart agrees with Njini founder Philip Tee's vision that the software should be extensible to use as a database optimizer. Wouldn't a single software suite managing information assets and storage plus optimizing databases be great?

◊ About the author

Dan Tanner's computer industry experience dates from 1959. He holds degrees in Electronic Engineering, Physics, Mathematics, and Education, and has matriculated for a Master's Degree in Business Administration. He has held Director and VP positions at leading vendors, consulting firms, and publications. He is a member of the Association of Storage Networking Professionals (ASNP), the Storage Network User Group (SNUG), and the Storage Networking Industry Association (SNIA), where he is an active member of several technical working groups, initiatives, special interest groups, and committees. A collection of his papers is available at www.progressmart.com.