

Scale Out for Enterprise Content Management

Enterprise Content Management is consistently a top-ten initiative of CIOs in the Fortune 1000. Over the last decade, ECM has grown from a nice to have to a necessity. This is because content in its many different forms plays an indispensable role in the running the business. Content is used to present products and conduct commerce with customers. Content documents and proves that business processes are well run and constantly improving especially in regulated environments and compliance requirements like Sarbanes-Oxley. Content is the medium in which designers, scientists, marketing and management plan and address new and existing markets. Content informs everyone involved in the value chain of what is happening and how can it improve. Moore's Law and Gilder's Law have ensured that the flow of information and content increased as desktop computers could hold, produce and download more. Content has come to cover web pages, office documents, scanned images, rich media, e-mails and electronic records.

A new model for managing content is required across the enterprise. The amount of information and number of users want to share is growing exponentially and diffusely. Centralizing and scaling-up the hardware of the organization is no longer sufficient a sufficient solution to users' needs. New content architectures are required to provide the means to scale out the content capacity of the organization with replication, federation and decentralized authority allowing each department to plan for their own needs and to utilize existing hardware.

Alfresco, the first Open Source Enterprise Content Management System, and MySQL, the leading Open Source Database Management System, are working together to employ a new enterprise content architecture to scale an enterprise's content infrastructure at much lower costs than traditional approaches to Enterprise Content Management. Core to this architecture is a new type of content repository that utilizes new clustering techniques, decentralization, modularity, replication, and federated indexing to scale to local and enterprise-wide requirements. The Alfresco architecture also uses MySQL tools to provide high availability in the enterprise environment for mission critical content applications.

What is Enterprise Content Management?

Enterprise Content Management (ECM) is a technology and discipline of managing content assets across an enterprise. These content assets are the pieces of information, documents and records that important for running the business and working with customers and partners. As a technology, many vendors, such as Documentum, IBM, Filenet, Interwoven, Vignette, and Microsoft Sharepoint, have created platforms and application suites to create, manage, publish and archive content as objects in a repository. The ECM market emerged in the late 1990's as part of the client/server technology revolution, was integral to the internet explosion and is now a \$2.5 billion market according to Gartner and Forrester. Alfresco was founded in 2006 by John Newton, founder of Documentum, and John Powell, former COO of Business Objects to provide an open source alternative for Enterprise Content Management.

As a discipline, a whole ecosystem of practitioners and system integrators in ECM has now emerged to describe how best to implement and deploy technology to streamline the use of content in the enterprise, automate paper processes, and show how to effectively use in business and customer interactions. These practitioners have created ECM applications targeted at solving some major issues in enterprises with handling large volumes of content that currently processed manually. The content managed by these applications can range from web content to office documents to business records and scanned images. Typical applications include:

- Document management is managing the content used in day-to-day business operations, particularly office documents, such as word processing documents, spread sheets, presentation and scanned documents.
- Knowledge portals provide search and browsing interfaces on relevant content for activities like competitive intelligence, sales and design.
- Web content management is the term most connected with content management and provides re-use across many sites, support for corporate internets, B2B extranets, multi-lingual sites, media web sites.
- Collaborative content management tends to be project-oriented bring together a team of people focused on a common objective and enhances collaboration with communication, discussion and issue resolution.
- Compliance applications are used to conform to various rules, policies and regulations and ensure that processes are documented, materials are clearly marked and versioned, and usage is audited.

Users generally access these applications through browsers as web applications, but this really depends on how the applications are implemented. Most users would prefer not to use web or desktop applications in preference to interfaces with which they are already familiar in using and accessing content, such as file browsers, Google-like interfaces, enterprise portals and e-mail. ECM vendors in general have implemented these types of interfaces and simulated these structures in their systems. Doing so has been a performance challenge for many as they attempt to meet the performance requirements set by Google and e-mail clients. It is also a challenge to match these interfaces to functional requirements of the ECM applications.

What is a Content Repository?

A content repository is critical Enterprise Content Management as the store of content, its metadata and relationships and provides services for managing and controlling content. A content repository is a server or a set of services used to store, search, access and control content and provide these services to content applications. A common repository provides services to different ECM applications to reuse content, control business processes that cross applications and define control policies just once regardless of the application. Services that content repositories provide applications generally include:

- Library Services, which is a catchall term for the basic content operations that include basic access, store, check-in, check-out, locking, version control, managing meta-data, audit trails and organizing content into folder structures.

- Data Dictionary Services that describe the metadata and relationship types associated with content types and bind them to specific content objects
- Organization Services that classify content into multiple categories and to confine searches.
- Search and retrieval encompasses full-text, metadata and category searches.
- Workflow services that implement business processes and ensure smooth execution by providing workload balancing and tracking which tasks have been performed and which are overdue.
- Collaboration services that revolve around the commenting and annotation of content, project tracking, resource allocation, and issue tracking.
- Lifecycle services manage the long-term policies concerning and are crucial for enacting archiving and deletion policies of records management.
- Security services control permissions on content that include simple read, write and delete to copying, printing, special properties access and delegation of ownership.
- Transform / Publish services converts content and content components into other formats and aggregates components into finished documents and web pages.
- Administration services to manage resources such as storage and database space, backup and recovery, replication, users and job control.

Content repositories have been around for at least 15-years and generally built on relational databases. Metadata about the content, such as descriptive information, process information, security, classification and relationships about content, are stored in the database to provide quick and flexible retrieval and to simplify the many ways in which this information might be use. The actual content stored in these repositories can be as simple as HTML and pictures for a website, but can often be office documents, scanned images, XML and streaming media. Storage may be in the database as binary large objects for large content such as complex office documents, images or rich media, or it may be stored in files to simplify the management of storage and streaming of content for rich media and content transformations. The role of the database is to provide the transaction control and recoverability required when adding, updating or deleting content, metadata, associations and audit information.

What distinguishes content repositories from other typical database applications is the level of control exercised over individual content objects and the ability to search, access, and update content. Access to these services requires wrapping every call with security to prevent unauthorized access or changes to content is more complex than that provided by SQL. The hierarchical nature of how content is used, found, controlled and accessed, such as working in folder structures and hierarchical classifications, requires a different type of API than is currently provided by SQL-2003 or object-relational mapping. The paradigm for search of content introduced by internet search engines such as Google has meant that the highly structured search requirements of the SQL Select statement are not adequate for end user. Content creation, capture and viewing from authoring and viewing tools require that these tools be able to access the content repository with standard file protocols and be able to do this in a scalable fashion that can support thousands of users. The complexity of

these services means that much of the business logic of the content repository can be as large as or larger than the database itself.

Requirements for Enterprise Scale

ECM applications can put a tremendous demand on content repository and the infrastructure to support the repository. The scale of these applications can have several dimensions strain the IT organizations ability to meet the needs of end users. Scalability challenges can be posed by the number of people access the repository, the complexity of the activities that they are performing, the amount of information that they are processing or searching, and how geographically distributed they are.

- **Scale of People** is dependent upon the number of users connected to the system at any time. Very large enterprises can be overwhelmed with many users writing content into the repository in heavy information environments such as intensive sales activity or claims processing.
- **Scale of Activity** measures how intensively users using the system and are they reading or writing information and what validation is required of their activity. In write intense environments, thousands and even hundreds of users can cause problems and even hundreds of users can cause problems. Volumes in the order of millions of reads can be problematic unless the repository is designed to handle this type of volume. Security adds even more complexity to the problem as the rights of a user to access or update information may need to be applied to individual pieces of content.
- **Scale of Information** is growing exponentially due to increasing bandwidth and the availability of information on the internet. Knowledge workers, whose jobs depend on having ready information available to make decisions and assist customers, collect more and manage exponentially more information on an annual basis. End users must be able to comprehend the structure and organization of millions and even billions of content objects as they search for or browse through categories of large or federated content repositories.
- **Scale of Geography** is determined by how distributed users are and how close they are to the information, and information. Regardless of whether a content repository is scalable, users and organizations separated by large distances can experience latent delays imposed by the speed of light and network transition barriers.

The typical response of enterprise organizations is to scale up the hardware in the infrastructure to address these challenges of scale. Aside from being a very expensive solution, particularly if it requires buying all new hardware, this does not necessarily solve the problems of content scale. If there are bottlenecks in the repository architecture that prevent the system from scaling, then adding any amount of hardware to add more people will not help. Unless the content repository is designed to handle these dimensions, it can bring down the system and any business operations dependent on that system.

Technology Stack

The Alfresco system is built with the latest state-of-the-art open source components including MySQL 5.0. The components have been chosen to provide the most

scalable architecture and speed the development of new capabilities. The following are the components that have been the most useful in building a scalable architecture.

- MySQL 5.0 – open source database with which the Alfresco system uses the latest features such as replication, clustering, views and cursors
- Java SE 5 – Programming with the latest Java development environment provides the Alfresco development team with state of the art enterprise scalability tools and our use of the new generics and annotations capabilities helps the team develop new capabilities very quickly
- Spring Framework – provides the Alfresco architecture with the scaffolding to piece together the whole application and allows it to seamlessly add and replace new architectural components with configuration rather than coding.
- Hibernate - the object-relational mapping interface manages the creation and storage of content objects and their metadata
- Axis – Apache Web Services framework that provides access to any programming platform including PHP and Microsoft .NET
- JBPM – is the leading Java-based business process engine, as the basis of our workflow capabilities.
- Lucene – the leading open source full-text engine, which the Alfresco system has extended to not only understand the text within a content object, but also its metadata and categories, and allows several repositories to be searched simultaneously
- Java Server Faces (MyFaces)– this next generation Java-based web development environment is designed to simplify the use of reusable components in a web page and to separate model, view and controller for scalability
- JBoss App Server and Tree Cache – provides linear scalability with loosely coupled machines that provide distributed caching and load balancing.
- CIFS – The only Java-based implementation of the Microsoft Shared File System CIFS (Common Internet File System) protocol made open source by Alfresco and designed to scale to very large organizations.

Enterprise Scale Out Strategies

The Alfresco system was architected and built by the cofounder of Documentum and the team that built Documentum's Java Web Development Kit, WebTop interface and portal integrations. Taking 15 years experience in designing ECM systems, they set out to design from scratch a new generation of ECM to meet the content management needs of today's enterprises.

The Alfresco system is designed to meet the scale requirements of large enterprises not just by scaling up, but also by scaling out through the enterprise. To Scale Out, an enterprise system must expand to be close to the users and take advantage of existing systems and machines. The Alfresco repository runs on existing hardware using a standard Java platform takes advantage of features in MySQL to replicate and cluster information on multiple machines. Built using the principles of Service Oriented Architecture, the Alfresco system integrates communication with other servers and uses web service to provide a scalable, stateless content application platform.

A very important component of Alfresco's Scale Out strategy is the use of a new development technique known as Aspect Oriented Programming (AOP) that was developed at Xerox PARC in the late 1990's. AOP applies "aspects" or modules on objects as a runtime configuration, in this case content objects. This AOP approach allows the repository to add only the functionality that is needed and allows for much more modularity in the architecture. This approach also influenced the way that the architecture presents the Alfresco modules in stateless interfaces. A fast, dynamic rule-base applies the aspects that are required by the application and can be assigned at any time through configuration. Some of the aspects that are available for content objects include: security, locking, versioning, metadata, classification, transformation, and workflow. Some general services that are applied independently of individual objects are remote access, caching, and transaction control.

The Alfresco architecture utilizes modularity, statelessness, lightweight data model, clustering, decentralization, replication, redundancy and portability to achieve this scalability. These elements of distribution and componentization allow the Alfresco system to be federated across existing hardware and to allow control to be pushed down to the individual systems rather than centralized in a single large repository. This contrasts with existing ECM systems that force an organization to consolidate various repositories in order to achieve scale, reusability and search-ability across the enterprise as well as force them to purchase more hardware to support this scaled up architecture.

- **Modularity** – Aspect Oriented Programming allow the Alfresco architecture to be built from configurable pieces that can be removed or replaced with lighter weight implementation to make the system run much faster. For instance, only the metadata needed for an application will be applied and if an application doesn't need versioning or classification, that aspect is not configured in for that application. Even communication boundaries can be configured in or out if they hinder performance. By making these functionalities configurable, it allows the application to strip off functionality that is not needed and run much faster. Removing unnecessary communication boundaries can have a tremendous boost in performance, but still make it available if needed in a distributed environment. Likewise, security checks on individual objects can consume considerable resources, but AOP allows for the introduction of simpler security models if applicable.
- **Stateless Interfaces** – Modules are bound to content objects through stateless interfaces that do not carry around context. The context is carried through the calling interface as the application applies the aspect to the content object. This allows the interface to be deployed on any machine. Load balancing and distribution become much easier as it is not dependent on data on any particular machine other than what might be available in the distributed cache. This style of stateless interface effectively supports exposing the Alfresco systems interfaces as web services.
- **Lightweight Data Model and Decentralization** – Previous generations of ECM systems tended to have complex mappings to various data tables in the database and a lot of business logic to enforce rules on datatypes within those tables. Alfresco has taken a much simpler approach to mapping metadata into tables that simplifies the reading and writing of objects and makes the data mapping simpler. Using attribute-value pairs in the database requires no central

administrator to control the mapping and makes it easier map similar or same attributes in different types and repositories. This makes it easier to decentralize the repository and spread the load down to departmental servers but aggregate searching at the enterprise level. As the data requirements are much lighter, the load on the DBMS is substantially reduced.

- **Federated Search and Decentralization** – Search in Alfresco is based upon the open source Lucene search engine. Unlike ECM systems that centralize search into a relational database, the Alfresco system primarily uses the search engine to perform queries. This allows the Alfresco system to search against multiple Lucene indexes of multiple repositories, thus providing federated search and balancing the load of search across multiple machines. This allows the organization to decentralize the management of content at the departmental level but provide enterprise-wide search. It also simplifies the search paradigm for the users as they do not need to know what attribute information or make a distinction between content and metadata.
- **Clustering** – The Alfresco architecture uses aspects to configure clustering of objects and their metadata into content objects. By simultaneously binding transactions with aspects, Alfresco can control the update of data and the distribution of that data to distributed caches such as JBoss's Tree Cache. The application using them does not even need to know that it has been added. The result is that as load is balanced across several machines, each thread on different machines can get local access to the metadata associated with frequently access objects.
- **Replication** – There are a number of reasons for implementing replication in an ECM system and one of them is for performance. The Alfresco system uses MySQL 5.0 replication to support the replication of metadata from one copy of the repository to another. The Alfresco server supports the replication of content and the synchronization of indexes on the remote copy. The copy of the repository can be a read-only local version of content to provide local access in a distributed environment. This copy of the repository can be used to support hot backups without affecting the performance of the primary repository. With future versions of MySQL, Alfresco will support bi-directional replication for synchronization of geographically distributed sites.
- **Separation of Concerns** – The Alfresco system has partitioned the components so that functionality can be evenly distributed either through remote machines or in a clustered environment. The web client is architected to separate data access into Data Access Objects from the user presentation. Java Server Faces is the latest Java standard designed to work in this type of scalable environment pushing the handling of complex user controls separate from HTML page and encouraging reuse and caching. The open templating engine separates the processing of data from the templating engine to provide for caching and scalable process as users choose the templating language of their choice.
- **Virtual File System** – The CIFS Shared File System implementation is designed to provide a separate, cluster-able file system interface. The CIFS protocol is complex with no clear transaction semantics. From the server's perspective, it is never clear when changes to files should be persisted. The CIFS interface handles updates intelligently while maintaining the transactional integrity of the content. Distributed lock caching ensures that the system can

scale without any one processor becoming a bottleneck in an enterprise environment.

A High-Performance ECM Platform

By implementing, a Scale Out strategy for scaling Enterprise Content Management across the enterprise, the Alfresco system is designed to handle any enterprise requirements. Modular and configurable, the Alfresco system can be adapted to the needs of the enterprise and optimize performance with lower hardware requirements. With much lighter data management requirements, unnecessary interprocess communication eliminated and greater reuse of information in caches, the Alfresco system is at least five times faster than proprietary, commercial systems on single server systems. With the modularity, transaction control, distributed caching, replication, and federated search, the Alfresco system has no barriers to upper end of scalability as systems and processors are added to support the repository or the enterprise. By emphasizing decentralization of content into separate repositories, the Alfresco system removes not just performance bottlenecks from providing enterprise content management architecture, but administrative ones as well.

Power of Open Source

Not much has been said about the value and power of open source yet. The Alfresco architecture stands on its own merits in providing a Scale Out architecture and a new approach to Enterprise Content Management. However, the open source nature of both Alfresco and MySQL provide not only the cost advantages of a product that is not sold to you by sales people, but also the innovation that comes with transparency in the development of each respective system. Concepts like Scale Out come from developers working in conjunction with customers and users to work in real-world environments. The result is a system that minimizes costs through efficient distribution of software and adapted to today's IT environment and budgets.

The Author

John Newton has had one of the longest and most influential careers in content management. In 1990, John co-founded, designed and led the development of Documentum and for the next ten years invented many of the concepts widely used in the industry today. John's latest venture, Alfresco, is creating an open source, open standards Enterprise Content Management System and Repository to extend the use of content management into applications and organizations that have not considered doing so before. John has also been an Entrepreneur-in-Residence at Benchmark Capital. John started his career at Ingres where he was one of the original engineers and ultimately ran the database development group. John has a BS in EECS from UC Berkeley.