

Adobe Experience Manager Scalability, Performance, and Disaster Recovery

A flexible, scalable platform promoting technical agility, high performance, and sound disaster recovery features.

Table of contents

- 1: Flexibility in Design
- 2: TarMK
- 2: MongoMK
- 3: Performance
- 3: Digital Asset Management
- 4: Adobe Managed Services Best Practices for Disaster Recovery
- 4: Summary

Adobe Experience Manager provides the ability to manage content and applications in an integrated fashion on one platform, making it possible for applications to be deployed as easily as content. When applications and content can be treated the same—deployed and managed on a single platform—they work better together. This philosophy is what makes the Adobe Experience Manager solution a powerful content management system.

Users are demanding more and more personal and customized experiences, especially on mobile devices. By the end of 2015, the number of smartphone users will surpass 3 billion and they are changing the way people think about communication, marketing, engagement, and commerce.¹ Over 90 percent of the Smartphone owners expect stores and businesses of all kinds to provide important services via mobile apps now or they will consider taking their business elsewhere.²

Software solutions must allow for growth to handle the inevitable increase in scale of every aspect of content and applications to meet user demands and be as hardened as possible against unpredictable events.

This white paper discusses the ways scalability has been built into Experience Manager along with outstanding performance indicators and excellent disaster recovery features.

Flexibility in Design

Experience Manager can be deployed within Adobe's cloud or on-premise. The deployment you choose will affect installation choices, scalability, performance, and disaster recovery. While there are scalability requirements, Experience Manager's design allows scalability to be managed depending on deployment type. Different usage patterns require different solutions and AEM allows a variety of implementations.

An Experience Manager installation is composed of a number of distinct parts, each of which can be configured in a variety of ways in order to handle all sorts of usage scenarios. You can easily scale the Experience Manager infrastructure to handle operational demands, including peak customer usage. Inside Experience Manager, there are two fundamental environments: the Author environment and the Publish environment. Each of these environments is built for scalability and each can be scaled independently of the other.

The Author environment is used by authors to enter and update content. The Publish environment contains content which you make available to the end users of your web site or mobile application. It is important to note that these environments can be scaled independently from one another. For example, a growth in publish side usage does not necessarily correlate to an increase in author side usage, thus adding a new publish server doesn't require adding a new author server.

The backbone of Experience Manager is a data storage system designed for content-centric applications, and combines the characteristics of conventional file systems with relational databases. Experience Manager offers scalability and performance at the web-scale level, thanks to the modern technology stack built in the Content Repository. Modern, world-class web experiences profit from the revised system architecture, which combines the advantages of the industry-leading JCR (Java Content Repository) specification together with significant performance and scalability improvements. These improvements benefit all use cases – including read, write, and search.

¹ Worldwide Active Smartphone Users Forecast 2014-2018, Dazeinfo, December 18, 2014. On the web at [this link](#).

² <http://apigeer.com/about/press-release/2014-holiday-shopping-alert-new-research-shows-mobile-devices-have-changed-shopping>

New Repository Foundation

In an effort to enhance scalability and performance, the content repository within Experience Manager has been rewritten from the ground up as part of the *Apache Jackrabbit Oak* project. Apache Jackrabbit Oak is a scalable implementation of the Java Content Repository API with support for structured and unstructured content, full text search, versioning, transactions, observation and more. It is used as the foundation of the content repository.

Apache Jackrabbit Oak is a new JCR implementation with a completely new internal architecture. Based on concepts like eventual consistency and multi-version concurrency control, and borrowing ideas from distributed version control systems and cloud-scale databases, the Oak architecture is a major leap ahead for Jackrabbit.³

Modern, world-class web experiences profit from the revised system architecture, which combines the advantages of the industry-leading JCR specification together with significant performance improvements.

There are two options for the persistence layer backend used by Oak—TarMK and MongoMK.

TarMK

TarMK is the standard persistence mechanism used by Adobe Experience Manager. It is a data storage engine purpose-built for content management applications, including web content management, digital asset management, and mobile application management. TarMK supports very high rates of both read and write throughput with zero external dependencies. Due to this lack of dependencies, TarMK can be operated entirely within the Adobe Experience Manager user interface; while external tool integration is available through standard Java Management Extensions (JMX) MBeans, such integration is entirely optional. When using TarMK, content is stored in standard POSIX Tar files on local file system. By default, binaries are stored in the same Tar files as metadata, but can optionally be stored separately. While Adobe does provide a backup utility for Adobe Experience Manager, many customers choose to integrate Experience Manager with their current backup infrastructure. Due to Oak's Multiversion Concurrency Control (MVCC) architecture, TarMK's files are always consistent and can be backed up using any file-based backup tool without any special quiescing procedures. All in all, TarMK provides high performance for content-based applications with minimal operational overhead.

MongoMK

The second backend storage option is MongoMK, a microkernel that uses the MongoDB NoSQL database. MongoMK allows linear horizontal scalability. Generally a single MongoDB cluster can support more than 5 Author AEM instances.

Usage Recommendations

In the Author environment, MongoMK is intended to be used in cases where the concurrent authoring activities are beyond the capacity of a single server, either in terms of processing power (CPU), memory, or a combination of both. In these circumstances, applying horizontal scalability can be a component of the solution.

As different Experience Manager deployments have different usage patterns, only real world performance testing will demonstrate whether or not this horizontal scalability is necessary. The following metrics have been identified as indicative that MongoMK should be evaluated:

- Thousands of named authors.
- Hundreds of concurrent authors.
- Hundreds of thousands of assets are ingested per day.
- Hundreds of thousands of page modifications (including automatic modifications, for example, from data feed import processes) per day.
- Tens of thousands of searches per day.

In the Publish environment, it is almost never appropriate to use MongoMK for web and asset management scenarios. There are some user-generated content and community scenarios that can be well served by using MongoDB. Customers with user-generated content requirements should work with Adobe certified experts to define the proper configuration for their specific business requirements.

³ Durig, Michael, Senior Developer at Adobe Research, "Oak, The Architecture of the new Repository," *Slideshare presentation*, September 22, 2014.

Project Considerations

The following items should be considered as prerequisites to any project using MongoMK:

1. MongoDB deployment architecture and sizing must be part of the project implementation with help from Adobe Consulting and/or MongoDB Architects.
2. MongoDB expertise must be present within the project team to have confidence in being able to sustain and maintain a MongoDB environment
3. While both the commercial or open source versions of MongoDB are supported, on-premise customers must purchase MongoDB Maintenance & Support contract directly from MongoDB Inc.
4. Overall AEM and MongoDB architectures and infrastructures well defined and validated by an Adobe Certified Experience Manager Architect

Performance

Maximizing performance is a partnership between a well-designed piece of software and a development team with a flexible implementation plan. Experience Manager has been redesigned to allow the user to optimize performance based on each individual's usage profile. Adobe has created a number of [guides](#) to help the user optimize performance and our team is there to help our enterprise customers customize their implementation.

Performance (or the lack of it) is one of the first things that your users notice, so as with any application with a user interface, performance is of key importance. To optimize the performance of your Experience Manager installation you need to monitor various attributes of the instance and its behavior.

Many Adobe Experience Manager installations begin with the default configuration and work well for small- or medium-based applications where performance of both publishing and authoring instances are good. But when your site grows and the content authors start their work, do your first performance and stress tests using numbers provided by the requirements. This will provide important data to help you understand how you need to optimize the system.

Experience Manager's Author environment is optimized for a relatively small number of users who each generate a high number of performance intensive requests when updating content pages and the individual elements on those pages. In the Publish environment, the number of requests is significantly greater and the speed is just as vital, but since the nature of the requests is less dynamic, additional performance enhancing mechanisms can be applied, such as caching or load-balancing.

Once Experience Manager has been installed and configured, performance can be optimized depending on the level of complexity of a project and the experience of the development team. Since your system load changes and the performance profiles of your system shifts over time, AEM allows for performance "tune-ups" to optimize system performance.

Performance tuning is an iterative process that involves, measuring, analysis, optimization and validation until the goal is reached. In order to properly take this aspect into account, implement an agile validation process in the optimization phase rather than a more heavy-weight testing process after each iteration.

The way a content repository is structured can impact performance as well. In general, Experience Manager has been optimized for deep, narrow trees, i.e. an individual Page (for Sites) or Folder (for Assets) should have less than 1,000 children. However, Experience Manager's content repository is also able to handle very flat structures, with millions of child nodes, assuming that an appropriate user interface is created to manage this content.

Digital Asset Management

Poor performance in Digital Asset Management (DAM) can impact the user experience in three ways: interactive performance, asset processing, and download speed. To improve performance, it is important to measure the observed performance properly and to establish target metrics.

Adobe Managed Services Best Practices for Disaster Recovery

This section describes Adobe's best practices as implemented by Adobe Managed Services (AMS) and that on-premises deployments should follow similar processes, just adapted to their particular deployment.

Adobe Managed Services is a reliable, secure, and flexible hosting and managed service for AEM cloud deployments. It allows businesses to focus on creating rich websites by shifting the infrastructure management, operation, support activities, and disaster recovery to Adobe Managed Services and support teams.

The foundation of AMS is the solid Amazon Web Services (AWS) infrastructure. The main building blocks are the Elastic Compute Cloud (EC2), Elastic Block Storage (EBS), Elastic Load Balancer (ELB), and Simple Storage Service (S3) technologies.

Adobe operates its customers' instances spread across the eight AWS Cloud Regions around the globe. These are in turn organized into two or more physically and logically separated availability zones or data centers, within which redundancy takes place. At kick-off, the customer and Adobe agree on instance sizing, deployment topology, and capacity configuration. At any point, the customer can have additional capacity quickly provisioned if the business demands it.

Instances of publish and author tiers are periodically backed up to Amazon S3 (while encrypted) using snapshots. This process takes only a few minutes and is followed by the distribution of snapshots to all availability zones in a given cloud region.

Differential backups are performed on a daily basis and retained for seven days by default. However, you can request to change the frequency and retention period (some limitations may apply).

For disaster recovery, five recovery modes target specific types of failure with different service-level agreements (SLAs) and recovery windows.

- Failure of an individual application server or data volume
- Failure of all application servers or data volumes in a solution tier
- Failure of the Amazon ELB, with or without an accompanying failure of instances
- Complete failure of an Amazon availability zone.
- Failure of an entire Amazon cloud region.

TarMK does support hot backups to provide redundancy and disaster recovery. MongoMK can support replica sets across data centers.

To ensure quality and SLA adherence, Adobe Managed Services practices "war game" exercises every two to four weeks, with periodic participation of Amazon support. Customers may also join with prior arrangement. In addition, Adobe mirrors real customer solutions and executes disaster drills at least once every six months.

Summary

In summary, most Experience Manager installations start with the default configuration that works well for small or medium sized applications. When your site grows and the content authors start their work, do your first performance and stress tests to understand how you need to customize the installation. Experience Manager can be scaled to meet the needs of your operation and performance optimized once you have determined the demands on your digital assets. In a cloud-based implementation, a number of disaster recovery checks and balances exist. For self-hosted instances, many Adobe resources exist to help you protect your Experience Manager installation and its digital assets.

For more information

www.adobe.com/go/aem

