Adobe Experience Manager Scalability, Performance, and Disaster Recovery

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

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

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.

Mongo . ,

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.

U age Recommenda ion

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 demonstrK,a.

P ojec Con ide a ion

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 perf(e)-11.9(r)-y-12. pehria8.4(e)-13.(n)-10.2(c)8.9(a)-9(u)1.(a)-9(c)-6.1(.2(c)1924.6(a)-4.9(g (o)-