Migration Scenario: Migrating Batch Processes to the AWS Cloud

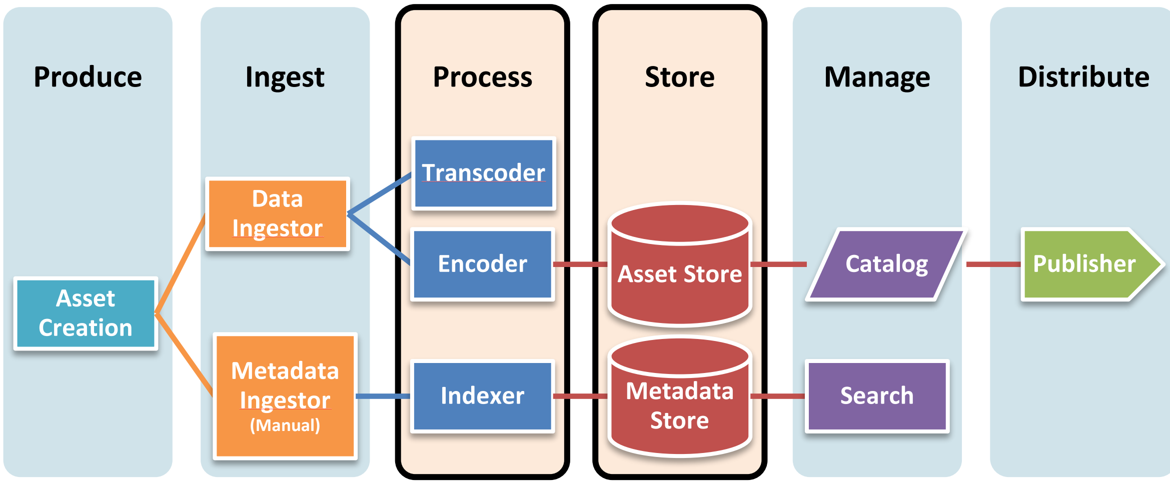

Figure 1: Company B Use case (Digital Asset Management System)

Use Case

Publishing Company B leverages a Digital Asset Management System (DAM) to manage digital assets such as documents and other types of media files. The DAM System is an end-to-end digital supply chain that processes, encodes, catalogs, stores and manages digital assets, distributes these assets to creative groups within various departments of the company and publishes them after encoding in different formats for different devices. See Figure 1 for the company's current digital pipeline.

Each night a batch process pumps in approximately 300 jobs to the processing pipeline, each taking an average of 1 hour on a single server to process. The system cannot scale more than 300 jobs a night due to the limited server capacity. During the day, developers and system administrators maintain the system by performing upgrades, applying patches and completing routine maintenance. System administrators often spend their time adding more storage capacity, replacing the "dead" drives, and "scrubbing" the old drives to ensure that proprietary data does not fall in to the wrong hands.

Maintaining IT infrastructure was not the core competency of the company, and they felt that reducing their IT footprint would free up resources to focus on what they were best at. The company was desperately looking to reduce the transit time through the pipeline (publish digital content more quickly) and focus on expanding its offering to new channels such as Web and mobile publishing.

Application Architecture

The batch processing and storage components were the main components of the DAM system. The batch processing infrastructure was hosted on-premise and consisted of 30 medium servers for transcoding and watermarking along with 5 high-end servers for encrypting of digital rights management (DRM). The scheduling and monitoring components of the processing pipeline would queue the jobs in a database and kept track of job status.

Figure 2 illustrates how the company leveraged the scheduler across the transcoder and encoder farm of servers. All the business logic was contained in C++ libraries with proprietary licensed software for transcoding, encoding, and encrypting.

The storage system used a standard Storage Area Network (SAN) with fixed capacity. All the digital assets (raw and processed) were stored in the asset store. An internal website kept track of the process while the reporting application provides reports and metadata to other components for the DAM system.

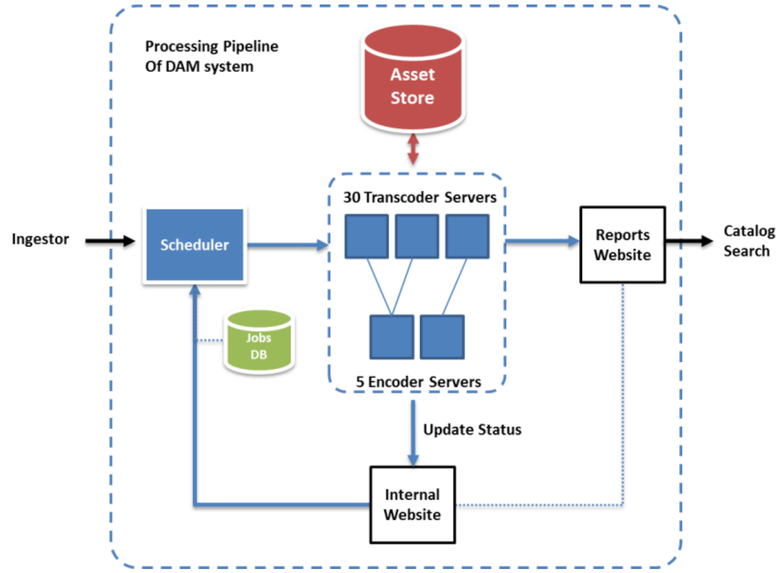

Figure 2: Architecture of processing pipeline (before migration)

Motivation for Migration

Company B was in need of a solution that would enable them to scale the Batch processing system without having to incur the capital expenditure of buying new hardware and hiring additional personnel to physically manage the servers. The current batch processing system could not scale to address the growing demand due to limited server capacity. Due to the company's infrastructure constraints and lack of automation, system administrators wasted a lot of time doing repetitive tasks such as prepping the servers and fixing failed hardware. Maintenance of the equipment was a major bottleneck and loading down the business, impeding growth and introducing inefficiencies.

Company B therefore decided to leverage the AWS cloud to address their growing pains, while continuing to keep the overall cost of infrastructure within their budget.

Migration Plan, Strategy, & Execution Steps

Cloud Assessment

The team assessed the different components of the DAM system and determined that the immediate candidates for the cloud were the 'processing and the storage pipelines' as those areas needed to be scaled out immediately to meet the growing demand. The team identified some new ways to improve their existing software development and system administration life cycle through automation. During financial assessment, they analyzed their past maintenance contracts, hardware configurations, storage size and used the Simple Monthly Calculator and concluded that the equivalent system in the cloud would be over 30% less costly to run.

During technical and functional assessment, it appeared that the main engines could easily be moved to Amazon EC2. However, tools and scripts might need to be created to smooth the migration process. The company decided to leverage its existing resource management tools - a website used to manage digital assets and an internal monitoring site used for managing the processing pipeline and reporting. They decided to store all the digital assets in Amazon S3 and the corresponding metadata in Amazon SimpleDB. They worked with the licensing vendors and confirmed that their DRM license was valid in the Amazon EC2 environment. The team prepared a migration roadmap in which the development team would run the systems in parallel for a month (co-existence phase) then switch jobs to new system after thorough testing and turn off the old system.

Proof of Concept

The team at Company B realized that they had several unknowns within their plan, mainly performance and latency of the AWS cloud and determining the right EC2 instance type(s) for their particular workload. Therefore, they decided to conduct a small proof of concept that would help them understand how their solution would perform in the cloud.

In doing so, they provisioned 3 Extra Large Amazon EC2 instances (for the transcoder and encoder components), installed the relevant libraries and manually injected test jobs to examine the performance and latency of AWS. The team was able to install and provision the entire stack on single Amazon EC2 instance. They concluded the performance and latency to be satisfactory. Additional optimizations could be done later. The benefits of elasticity and not having to manage the hardware were the biggest driving factors.

The proof of concept was helpful in letting the team test their existing software in the cloud, gave them additional confidence in AWS, and also helped the team identify other components of the DAM System that they would move to AWS.

Data Migration

Data Migration phase involved moving all the existing digital assets and associated metadata of the digital assets to the cloud.

The team was able to collate several catalogs from their SAN into categories and upload most of the assets to Amazon S3 in batches. They leveraged the AWS SDK for Java to upload the assets concurrently and AWS Management Console to verify. For some digital assets, they encrypted the assets prior to uploading them in the cloud using a commercially available encryption software package. To maximize the upload throughput into Amazon S3, the team used multiple threads in parallel across a partitioned set of Amazon S3 buckets. For one particular catalog that could not be imported over the Internet due to its size in the time frame the company required, the Amazon Import/Export Service was used. The service allowed the catalog to be shipped to AWS on a USB 2.0 hard drive and loaded into Amazon S3. The entire migration process took just a few weeks.

For metadata (name, authors etc.) associated to the digital assets, the team decided to get rid of their internal relational database and leverage Amazon SimpleDB. Metadata of each asset of a given catalog was stored in an Amazon SimpleDB domain. The team uploaded all the metadata multi-threaded BatchPut requests within a week.

Application Migration

The company decided to implement the “Forklift Migration Strategy” and moved most of the components and their dependencies at once. Custom Amazon Machine Images (AMIs) were built for the Transcoder and Encoder engines, so that when additional capacity was required for these components, the team could simply use the pre-built AMIs manually. Some of the scheduler components were replaced by Amazon SQS queues (input, output), which required refactoring and modification of a few parts of the code. At this stage, major components of the processing pipeline were moved to the cloud and tested using data stored on Amazon S3. The health of the instances was monitored and managed using the AWS Management Console. After end-to-end testing of the DAM System for a few weeks (running in parallel with the mainline), Company B decided to switch to the cloud. See Figure 3 for a picture of the final architecture after migration.

In the process of moving the application, the team also modernized some of their IT assets. They built a load testing framework consisting of various Amazon EC2 instances to help analyze performance of cloud applications and set appropriate expectations for how to handle growth. The team also built a key management solution to manage and rotate all the keys used for encryption of digital assets.

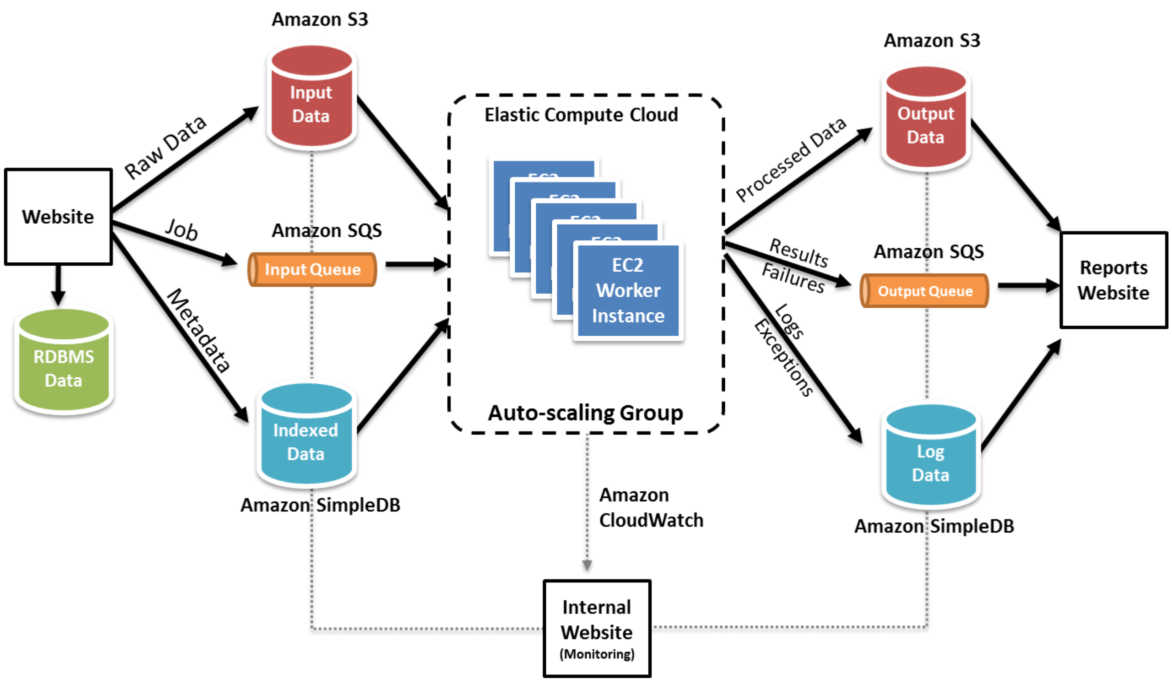

Figure 3: Company B Batch Processing System (After Migration)

Leveraging the Cloud

In order to achieve maximum flexibility and reap the benefits of elasticity, Company B's development team decided to leverage some of the advanced features of the cloud. They bootstrapped each instance in the system such that an instance could be stopped and restarted from an AMI and resume the state. An Auto Scaling Group was created to automatically scale the number of instances up and down based on system load. Each instance played a distinct role in the system and was self-discoverable on boot. Each instance knows what configuration to pull down based on user data passed to it at launch e.g. For example, an instance was passed "Transcoder" string at launch. Transcoder instance dynamically self-discovers itself on boot, grabs the appropriate apps, installs the necessary apps and patches automatically, and configures itself to join the auto-scaling group cluster on-demand. The overall system was now not only scalable elastically but also highly available in an event of any failure.

Overall security of the system was also hardened by a variety of means. For example, each production instance is launched with a strict security group which restricts access to other IP addresses and only certain users have access to source files of digital assets in the cloud.

Optimization

Once deployed, there were a number of ways the team decided to optimize their deployment in the cloud. Firstly, for higher utilization, they ran some analysis and determined that rather than having a fleet of 30 servers running regardless of load, the new system would expand and contract the fleet as needed to ensure maximum utilization of running servers. The team was also able to fine tune the configuration and processes because they were no longer dependent on the hardware.

Secondly, the Company B team was able to achieve higher performance by caching frequently accessed digital assets (popular documents, videos) by implementing a *memcached* cluster running on a separate cluster of Amazon EC2 instances. These two optimizations reduced the system management workload for the team and also helped reduce costs.

Conclusion

The final DAM System consisted of components that were completely hosted in the AWS cloud. The entire processing system is now highly scalable because it is using highly scalable components (Amazon S3, Amazon SQS, Amazon SimpleDB). The system is elastic and auto-scalable because the resources can now be provisioned and decommissioned on-demand with just a few clicks (Amazon EC2). The monitoring website monitors and manages capacity – the new dynamic cloud of transcoder and encoder Servers. Amazon S3 and Amazon SimpleDB conceptually provided infinite storage capacity and as such, the asset and metadata stores will never run out of capacity which is one less thing for their System Administrators to worry about. Developer productivity was increased due to automation and they are now able to spend more time developing features and upgrades. Overall, Company B's business is able to address the growing demand and publish the digital assets in record time.