A Fully Cloud-based Global Visual Effects Studio

Jim Vanns*
Senior Production Engineer, Industrial Light & Magic

Aaron Carey†
Production Engineer, Industrial Light & Magic

Figure 1: *A global studio infrastructure and pipeline*

Abstract

Recently, visual effects (VFX) companies have been increasingly developing hybrid cloud solutions with the majority of core infrastructure remaining on premises and secondary compute in the cloud. This has the obvious benefit of allowing 'bursting' into the cloud, but does not take full advantage of the cloud's dynamicity, and usually incurs high egress costs to retrieve data. We present a fully cloud-based global studio, using open source and custom software and highlight the opportunities available with this novel approach to both infrastructure and VFX pipeline.

Keywords: cloud, pipeline, infrastructure, storage, rendering, database

Concepts: •**Networks** → **Cloud computing;** •**Information systems** → **Cloud based storage;**

1 Living Without A Filesystem

One common problem faced by large VFX studios is load on a centrally shared filesystem. This load tends to increase as projects progress and becomes most problematic as deadlines approach.
Initially we evaluated clustered file systems including Lustre, Gluster and Ceph but began to wonder if we could remove the bottleneck entirely (and associated admin effort) by simply not using a centralised filesystem.
In our implementation, all asset data are stored in object storage in the cloud (eg. Amazon S3) and downloaded locally to cloud instances whenever any compute action is required (including artists working interactively with the files). This simplified our work immensely and also provided some additional benefits. A large problem with a traditional shared filesystem arrangement is that a significant portion of metadata is stored intrinsically in the filesystem hierarchy. Just by its position on disk, you can tell a lot about an asset or render, and this information has to be kept in sync with a central database which stores all of the other metadata. By removing the shared filesystem, we could be sure that all metadata are stored only in a single database which doesn't need to be kept in sync with anything. Files exist as binary objects in the object store and references are kept in our central database allowing them to be checked in and out to local scratch disks. As all assets are accessed on a check-in, check-out basis, security is more easily addressed: users only have access to assets they are allowed to see.
Working in this way also means that all of our asset files are now globally available without the pain of large transfers between sites. It also means that any compute tasks will hit local disks, rather than network attached storage, which ensures consistent high performance.
The obvious tradeoff to this way of working is that files are not immediately available to work on. We have explored some open source and commercial caching options, and are in the process of testing our pipeline at scale to better understand how this will affect processing times.

1.1 NoSQL coupled with NoFS

Decoupling the (file) metadata from the content presented the opportunity to model this data as a graph, effectively forming the foundation of an asset database, which could then be combined with the site, studio, show and other pipeline data which we were already modelling in this way. To do this effectively we compared distributed graph databases (eg. Titan+Cassandra) with traditional RDBMS (eg. PostgreSQL) and associated global replication schemes.

2 Moving Beyond Just Rendering

Rendering in the cloud is cheap with regards to compute, but large penalties in both time and cost exist when transferring data in and out of a public cloud network. As part of this project, we wanted to see if it was possible to replace artist workstations, typically expensive high end machines, with virtualised compute resources accessed via thin client.
Through the use of Mesos we have moved to an 'application-as-a-service' (AaaS) approach offering content-creation applications such as Maya as a service, running somewhere in our compute cluster. All compute (inc. the 3D accelerated GFX processing) is performed on the cloud host, in a container, and only the 2D image is

*E-mail: jvanns@ilm.com
†E-mail: acarey@ilm.com

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4282-7/16/07
DOI: http://dx.doi.org/10.1145/2897839.2927432

sent back to the workstation display. This has a number of benefits: allowing content generated to stay within the cloud, close to the compute resources; as well as allowing artists in more distant locations (for example on set) access to high end hardware and applications. The transport method used also allows for SSL/TLS encryption of data in transit to ensure security over public networks. At rest, the data can optionally be encrypted 256-bit AES, for example.
Latency here is obviously an issue, and this approach may not be suitable for all disciplines (depending on the distance to the data centre), but there is precedent for this model already in production at ILM today.

3 Being Truly Global (and following the money!)

The main goal of our prototype pipeline was to be truly global. By this we mean an artist working in the UK can submit a task for rendering on a compute cluster in Singapore which can then be reviewed by a supervisor in the US, and this whole process should be transparent to the user. Indeed we wish to be agnostic to the studio location - no primary site, no inter-site entanglements, dependencies etc. By design this also paves the way for simple segmented or isolated networks (domains, subnets etc.) for high profile or sensitive projects.
This approach has allowed us to make use of global pricing differences in the Amazon Web Services (AWS) spot market to harness the potential savings by running compute tasks in whichever region appeared most cost-effective. This will likely be the most compelling avenue for further development in the future.

3.1 Embracing the datacenter kernel; Mesos

Mesos has influenced the evolution of our prototype and now forms a core part of the solution as a whole. With it we not only implemented AaaS but also wrote our own framework that promises to offer the benefits of a dynamic global cloud market: the supply and demand of resources serves as input for a scheduling system 'sidecar'. This data feed instructs a provisioning system to boot new hosts that supply any required resource - bounded by cost centres. The provisioning can be directed, for example, to the region that offers the cheapest AWS EC2 Spot Market prices.

Acknowledgements

To Amazon Web Services, for their encouragment and support on this project and particularly their expertise in the EC2 Spot Market.