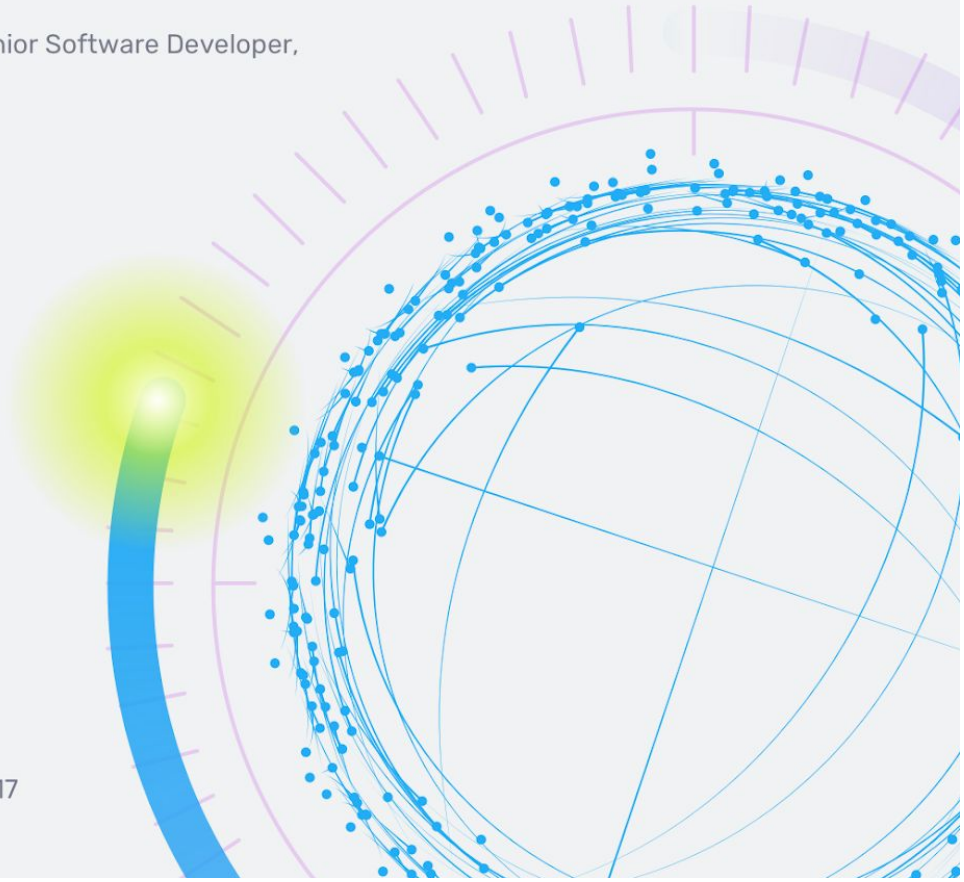
influxdata

Building Granular Metered Pricing for SaaS

AN INFLUXDATA CASE STUDY

John Burk, Senior Software Developer,
PipelineFX



September 2017

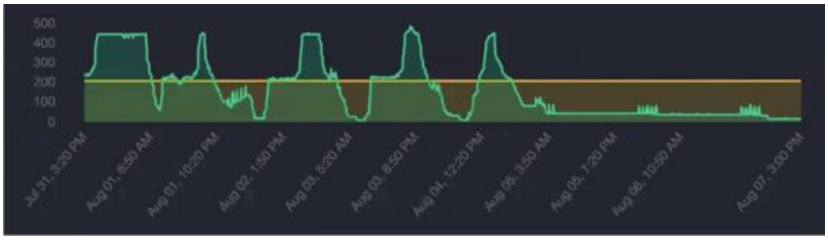
Company in brief

PipelineFX has been helping organizations of all sizes to better manage rendering for digital media applications and programs. Its software product, “Qube!” was originally built for Square USA’s groundbreaking animated movie “Final Fantasy: The Spirits Within,” and has been used on hundreds of feature films since. PipelineFX works hard to understand your rendering workflow and requirements, and offer comprehensive products and services to dramatically improve your rendering performance. Success in digital media today requires maximum efficiency, and PipelineFX will strive to optimize your existing infrastructure as well as planned future expansion.

Case overview

PipelineFX provides a SaaS offering, called “Qube!”, which allows organizations of all sizes to better manage rendering for digital media applications and programs. The issue with rendering any digital asset or movie is that it is a very dynamic process that can consume a variable number of resources during the rendering and then use none when the rendering is completed.

Typical Node Usage vs. Time for Digital Rendering



The business problem is how to help customers achieve maximum efficiency paying only for the resources they consume, while allowing the PipelineFX service to achieve maximum utilization across the whole server rendering farm. PipelineFX chose InfluxDB Cloud and MySQL as their solution to provide a real-time metrics and billing at 1 minute level precision. This allows their customers to maximize their investment in Qube without having to commit in perpetual licenses that may go unused.

They also use InfluxDB Cloud to provide monitoring and management of the customer’s digital asset batch, providing insight into disk and node utilization and performance.

> *"MySQL is not intended for time series data...I can testify it is like pounding nails with a screwdriver. It's definitely not what you want to do in any relational database."*
>
> ***John Burk,*** *senior software developer*

The business problem

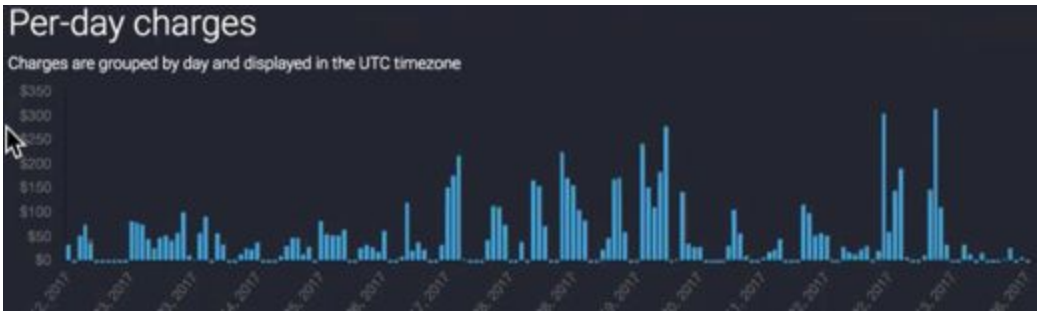
PipelineFX helps organizations of all sizes to better manage rendering for digital media applications and programs. For over 16 years, organizations of all sizes in the film, visual effects, post production, broadcast, design and education industries have been using Qube! to manage their render pipeline and make critical business decisions, faster and smarter than the competition.

Problem: Allow customers to scale to meet deadlines

The business problem is that the whole rendering process is a very "bursty process." So one customer may need 10,000 cores one night and only a couple of hundred the next night. In addition, based on the level of computer graphics, layers per shot, resolution of the rendered project, and the number of frames per second, the time needed to render each project is very different; and yet, the customer's deadlines are not variable. In order to meet a deadline, they may need to scale from 500 to 1,500 machines for 3 days. But after the 3 days, the customer does not want to pay for the rendering licenses that are not being used. Their job is completed, so they should not need to pay!

Problem: Allow customers to be billed at minutes increments

Due to the nature of the bursty process and the variable described above, resource utilization usually looks something like this:



What makes the PipelineFX offering so compelling is the ability to measure utilization every minute and bill at a very high frequency. This high level of granularity ensures that customers get a great return on their investment in PipelineFX. They also provide a very flexible pricing model that can accommodate this high frequency charging for any license utilization over a certain per-paid minimum. In addition,

the system needs to provide reports and insight into what average utilization rates are and if a customer should consider increasing their minimums.

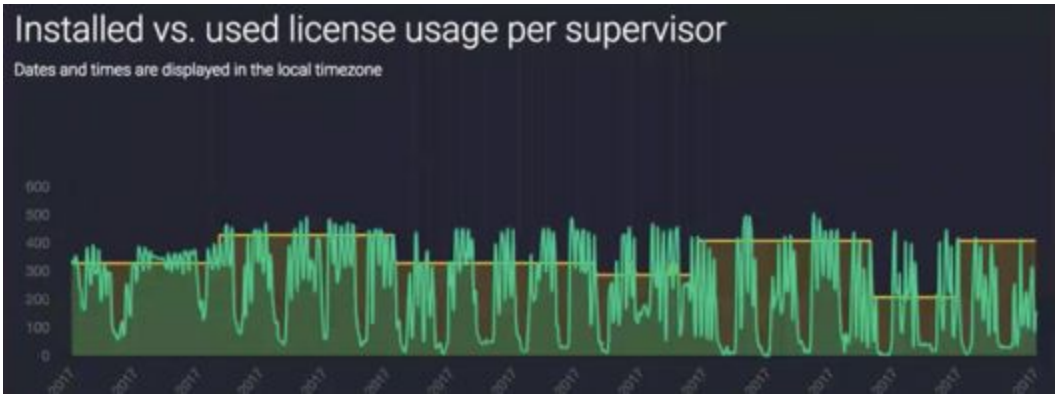
Problem: Optimize the SaaS infrastructure to optimize customer demand and server costs
PipelineFX needs to be able to reserve the right number of servers and nodes to hit the peak demands, but they do not want to oversubscribe on nodes for licenses. They don't want to buy licenses or nodes where the demand doesn't justify the costs.

For both of these problems, since they want to measure and bill at a high frequency time interval as well as determine real-time peak demands, the traditional relational database that they used for these uses will no longer work on its own.

Solution

PipelineFX deployed InfluxData collectors (Telegraf) on-premise and on the cloud, basically wherever the rendering was being performed. The data is collected per minute and then transmitted to the PipelineFX service every 15 minutes. If they lose network connectivity, it just accumulates the batch, and once connectivity is restored the whole batch is transmitted. Often 8 to 12 hours of data is uploaded to InfluxDB, some late arriving, but the system takes care of this as well. PipelineFX uses this data stored in InfluxDB to generate customer dashboards and provide reporting of real-time usage.

For example, this report shows the licenses purchased (orange lines) and the actual usage in green:



Graphs like this allow customers to answer questions like:

- "Do I have enough licenses?"
- "Did I buy enough licenses for the period in May and early June?"
- "When did I peak over my license costs?"
- "Why were my costs so high over this period?"

By providing visibility into their usage data, vs. license data, they are better equipped to make business decisions to optimize their costs.

> *"Providing insight into the actual usage vs. purchased licenses allows our customers to optimize their ROI. Indeed sometimes they need to save money by spending more money!"*

Although all the data is collected and stored in InfluxDB, all the license usage data is stored in a MySQL database, and MySQL is used for billing. MySQL database is required because the whole license scheme has a highly relational schema — usage data is keyed off its primary MAC address, and license keys and version entitlement are keyed through that MAC address as well. And that key ties back into billing accounts, users, etc. InfluxDB Cloud is also used to store all the system metrics to provide visibility into their rendering batch performance. It is also vital for remote troubleshooting. With InfluxDB Cloud, PipelineFX can drill down into usage and determine if a customer's machine is running low on disk space or has too many jobs running.

> *"InfluxDB Cloud's database (InfluxDB) is a high-IO database server that tracks a large number of metrics that allows us to drill down and solve customer problems remotely."*

Why Telegraf?

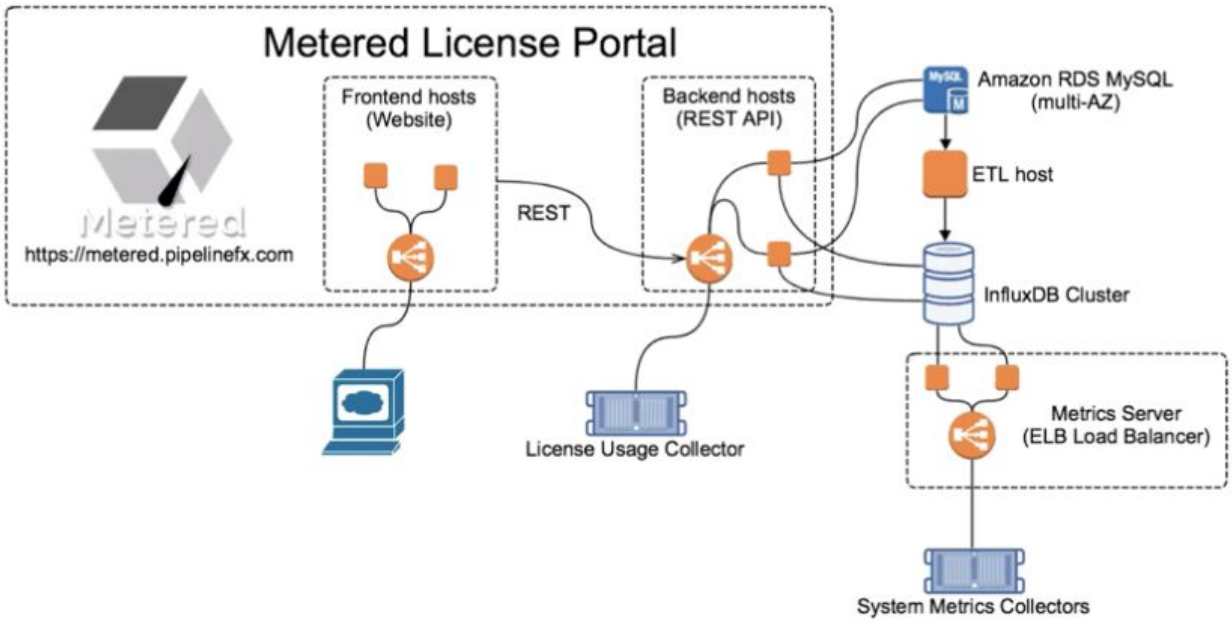
When they initially deployed InfluxData, they built custom collectors for Linux and Windows in Python with Python virtual libraries. Their long-time experience with monitoring with tools like RRD and with software development made it seem like an easy thing to do. But over time, they have found the packaging to be problematic and therefore have made the decision to use the Telegraf plugins instead to save time and money.

Why InfluxDB Cloud?

PipelineFX started out with the open-source InfluxDB product but needed high availability and scalability because this was a customer-facing application. They also wanted to future-proof their offerings because they want to scale to over 1,500 customers. They just looked at the cost of hosting InfluxDB themselves vs. the hosted InfluxDB Cloud offering and from an ROI perspective, InfluxDB Cloud just makes better financial sense. In addition, they know they can count on the experts at InfluxData to ensure that the software is always up to date, their clusters are optimized, and the overall solution is monitored 24x7.

> *"With InfluxDB Cloud, we basically wanted a managed solution. We don't need to monitor it, manage it, configure it. InfluxData has the experts, and the system just seems to always be up and performing nicely. It's been basically, a fire and forget."*

Technical architecture



The metered license portal is the customer's interface to the dashboard. This points to an Amazon ELB (elastic load balancer) and there's one or more hosts behind that. All interaction is over REST APIs. The license usage collectors talk directly to the backend hosts over REST, the time series data is stored in InfluxDB Cloud (InfluxDB Cluster in the diagram), and MySQL keeps all the billing information. The charts in the customer portal are rendered using AngularJS using ChartJS and the Bootstrap framework.

Results

PipelineFX has been able to help their customers to achieve maximum efficiency and optimize their existing and future infrastructure. The combination of InfluxDB Cloud and MySQL has allowed for

metering and billing at minute-level precision. In addition, they use InfluxDB Cloud to allow them to provide a real-time view to their customers on their current usage and billing charges. This allows their customers to maximize their investment in Qube! without having to commit in perpetual licenses that may go unused.

About InfluxData

InfluxData is the creator of InfluxDB, the open source time series database. Our technology is purpose-built to handle the massive volumes of time-stamped data produced by IoT devices, applications, networks, containers and computers. We are on a mission to help developers and organizations, such as Cisco, IBM, PayPal, and Tesla, store and analyze real-time data, empowering them to build transformative monitoring, analytics, and IoT applications quicker and to scale. InfluxData is headquartered in San Francisco with a workforce distributed throughout the U.S. and across Europe.

Learn more.

InfluxDB documentation, downloads & guides

Download InfluxDB
Get documentation
Additional case studies
Join the InfluxDB community

799 Market Street
San Francisco, CA 94103
(415) 295-1901
www.InfluxData.com
Twitter: @InfluxDB
Facebook: @InfluxDB