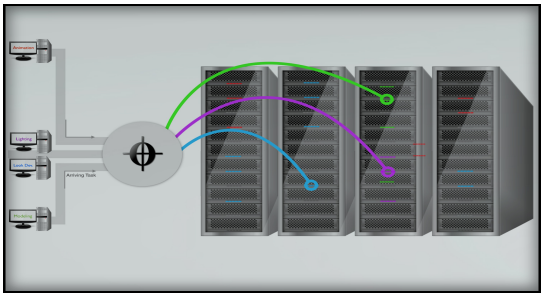
Coda: The cultural effects of queueing at Disney Animation

Marc Jordan Kevin Constantine*
Walt Disney Animation Studios

(a) *Walt Disney Animation Studios Renderfarm*



(b) *Coda Queueing System*

Abstract

When we began designing the new queueing system for Walt Disney Animation Studios, we had a simple goal in mind: Take some shell commands, run them on some remote machines, and return the results. What we thought would be a six month project evolved into an eight year journey that helped revolutionize the culture at our studio, and built a world class queueing system capable of executing millions of tasks per day. Three key features of Coda, automated render wrangling, localized job priority, and advanced desktop rendering, have helped to encourage a culture of trust and collaboration, both amongst the different shows competing for queue resources, and between the production and technology organizations in the studio.

Keywords: batch queueing, studio culture, renderfarm, scheduling

Concepts: •**Software and its engineering** → **Designing software;** •**Human-centered computing** → *Field studies;* •**Computer systems organization** → *Grid computing;* •**Theory of computation** → Distributed computing models;

1 Automated render wrangling

Our studio used to have a render wrangler team in the technology group that was tasked with making sure that the render farm was being used efficiently. This team worked in shifts, around the clock adjusting jobs in the queue to keep the farm fully utilized. They were not only responsible for fixing broken frames and replacing failed hard drives, but this technology group was essentially implementing a manual, human-driven algorithm to share the queue resources between the different productions. This put them in the unfortunate position of having to arbitrate the availability of a scarce resource among multiple groups who each believed they were most important.

*{Marc.Jordan, Kevin.Constantine}@disneyanimation.com

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4282-7/16/07
DOI: http://dx.doi.org/10.1145/2897839.2927420

Coda divides the render resources into multiple allocation pools. These pools are governed by a sharing mechanism, which we call speculation, that allows jobs from one pool to overflow automatically into the compute resources of other pools if there isn't enough work to fill those pools. When more work for those other pools is submitted, jobs running outside their pool will be preempted and returned to the queue. Thus the scheduling automatically re-adjusts to ensure that each pool gets its configured share. This ensures that the queue is always fully utilized as long as there are jobs anywhere in the studio that can be run, and is not limited based on what the current allocations happen to be. Now that the queue auto-balances between pools, our teams focus on troubleshooting the jobs themselves, rather than worrying about getting them running in the first place. Further, we built tools to allow the productions themselves to manage and adjust the allocation of queue resources. The leadership for the different productions communicate regularly with each other to negotiate the allocation of resources across the studio, rather than having those requests be arbitrated by the technology organization. This increases the visibility into the resource needs of the studio as a whole, builds trust between the different competing productions, and allows them to be more agile in responding to changing queue needs in the studio.

2 Job priority is localized

Ask 10 department managers which queue jobs are "high priority", and you'll get 10 different answers. As scheduling needs change over the course of production, the priority of the jobs in the queue get adjusted to meet those needs. In the past, these priority adjustments were proxied through the render wrangler group which had to balance those adjustments within a single priority list of all the other jobs in the queue.

In Coda, each department is given an allocation pool, and each of the pools has it's own discrete priority list, thus changes to job priorities are localized to the department's allocation pool that the jobs are scheduled in. Each department manager now has the ability to change the priority of jobs in their department without having to worry about causing adverse effects to the jobs from other productions or departments in the queue. We've also developed tools to allow department managers to adjust job priorities themselves, giving them immediate control of the jobs in their pool without having to proxy those requests through a third party.

3 Advanced Desktop rendering

Artist productivity is extremely important. To that end, they are each given very powerful workstations. However, their workflow tends to be bursty in nature. That is to say, they have huge computational demands for very short periods of time. This leaves the workstations underutilized for much of the day, especially at night and on weekends.

We approach the solution for accessing these unused processors in three ways. First, we tackle the interactive effects of queue renders on interactive performance by restricting how much of an artist's machine a queue job has access to. Next, we develop a tool that allows artists to specify how much of their machine is available to the queue. Finally, we need to incentivize artists to donate as much as they can to the queue. We approach this through gamification. We track the number of render hours each machine provides over the course of a production. At the end of each show, awards are given to the owners of machines donating the most time.

4 Conclusion

While we initially set out to solve technological barriers with the queueing system we had, we managed to achieve significantly more. Coda increased our render farm utilization which has allowed us to optimize our productivity. Further, Coda has removed the technology group from being the arbitrator of resources within the studio, and engendered a feeling that the queue is a studio resource that, when shared effectively, can meet the needs of the studio as a whole.