There’s another aspect, too. When the customer picks up his order, it should be the right one! That is, the product we deliver really needs to match the product the customer ordered. It seems like a trivial statement, but it is very important to render the production scale images in the same way that the on-screen image was rendered. We worked hard to ensure that exactly the same rendering code would be used in production as in the studio. We also made sure that the rendering engine would use the same fonts and backgrounds in production.

In our render engine, we adopted a philosophy of “Fail Fast, Fail Loudly.” As soon as the render engine pulls a job from PCS, it checks through all the instructions, validating that all the resources the job requires are actually available. If the job includes text, the render engine loads the font right away. If the job includes some background images or an alpha mask, the render engine loads the underlying images right away. If anything is missing, it immediately notifies PCS of the error and aborts that job. Out of the 16 steps in the rendering pipeline, the first 5 all deal with validation.

After several months in production, we finally found one error that the render engine didn’t detect early: it didn’t reserve disk space for the rendered image up front. One day when PCS filled its storage volumes, render jobs started to fail late instead of failing early. In all the preceding time, there were no remakes due to bad renders.

Scale out

Each render engine operates independently. PCS doesn’t keep a roster of the render engines that exist; each engine just pulls jobs from PCS. In fact, engines can be added or removed as needed. Because each engine looks for a new job as soon as it finishes the previous one, we automatically get load balancing, scaled to the horsepower of the individual engines. Faster render engines just consume jobs at a higher rate. Heterogeneous render engines are no problem.

The only bottleneck would be PCS itself. Because the render engines call stored procedures to pull jobs and update status, each render engine generates two transactions every three to five minutes. PCS runs on a decent-sized cluster of Microsoft SQL Server hosts, so it is in no danger of limiting throughput anytime soon.

User Response

Our first release was installed at two local studios, both within easy “drive-and-debug” distance. The associates’ feedback was immediate and very positive. One studio manager estimated that the new system was so much faster and easier to use that she would be able to handle 50% more customers during the holiday season. One customer was reported to ask where she could buy a copy of the software. We commonly heard reports of customers taking the mouse directly and making their own enhancements. You can imagine that customers are much more likely to order products they’ve created themselves.