THE JHCS RENDER FARM SETUP EXPLAINED

Background

Jim Henson's Creature Shop is a studio that has many different types of artists, using many types and versions of software. Their design-preview-redesign workflows can consist of frame render times ranging from seconds to several hours. On top of this there is the problem of having multiple projects, deadlines, and limited render resources.

Basically we have lots of people who often have different requirements of our currently small farm.

With project deadlines approaching there was a feeling that we needed to revise our farm setup and come up with something that was optimized and fair.

Introduction

It's important to first explain some of the terminology. The Alfred software is quite a complicated beast and subtle differences in language can very often cause confusion.

A **dispatcher** is a job queue manager, doing the actual work of reading the job scripts and launching the individual commands. Typically there are several dispatchers on the network, each managing a user's job queue and local clienting load. Dispatchers negotiate with the maitre-d to acquire remote servers. At Henson's we also have "Centralised" dispatchers. These are essentially the same except that many different users can send jobs to these job queues. They do this through in-house tools. They make it easier for the wranglers to manage.

Although it is understood that some artists require personal dispatchers during the day to help their workflow, **ALL** artists should send over night renders to the centralized dispatchers. These jobs will always get priority.

The **maitre-d** is the big guy that arbitrates all the requests for servers (render nodes) emanating from the various dispatchers that are scattered around the building. It operates from a "master schedule" file that we, the wranglers, set up in a clever way so that the rendering resources are shared out in a fair and efficient way. This cleverness is what this document is about.

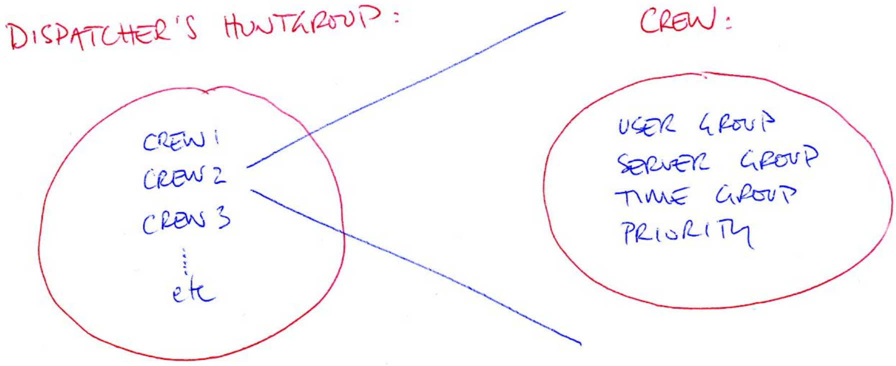

Figure 1: The Dispatcher's Huntgroup

Each dispatcher has a **huntgroup**. A huntgroup is made up of all the crews that are made available to a dispatcher (Figure 1). The distinction between a huntgroup and a crew is an important one that many people seem to miss.

One of the things that the maitre-d does is to make several crews available to dispatchers. The various **dispatchers gain access to the farm through these crews** (Figure 2).

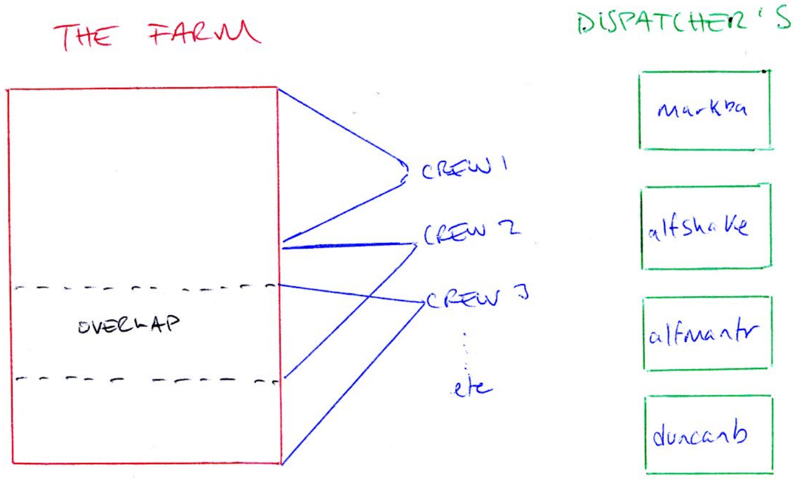

Figure 2: Dispatcher's Access to Farm through Crews

Note that the ranges of servers that the crews map to on the farm may overlap. This is an important point that we will come back to.

A crew is made up of (Figure 1):

1. user group - defines which dispatchers have access to a crew

2. server group - defines which render nodes the crew actually maps to

3. time group - defines which times during the day the crew is active

4. priority - the priority that a job going through a crew will get on the farm

Analysis

In order to solve this complicated resource allocation problem a survey was carried out in order to obtain a set of requirements from which we would then formulate setup goals and eventually an implementation strategy.

Note that historically the farm has been split into a 2D and a 3D part. This has been seen to generally work well as 2D jobs tend to be rather short whereas 3D ones, on average, take far longer.

The survey led to the following list of user requirements:

1. Lighters need to render single frames (usually using net-render) on the farm quickly -> only during the day.
2. Be able to make a user or a particular job a high priority for an arbitrary amount of time.
3. For 2D to spill over into 3D farm when it is not being used.
4. For 3D to spill over into the 2D farm -> between midnight and 8am.
5. For wranglers to be able to prioritise any job or user within 5 minutes of a phone call.

Design & Implementation

The analysis has lead to the following farm setup design goals and their implementations (Figure 3):

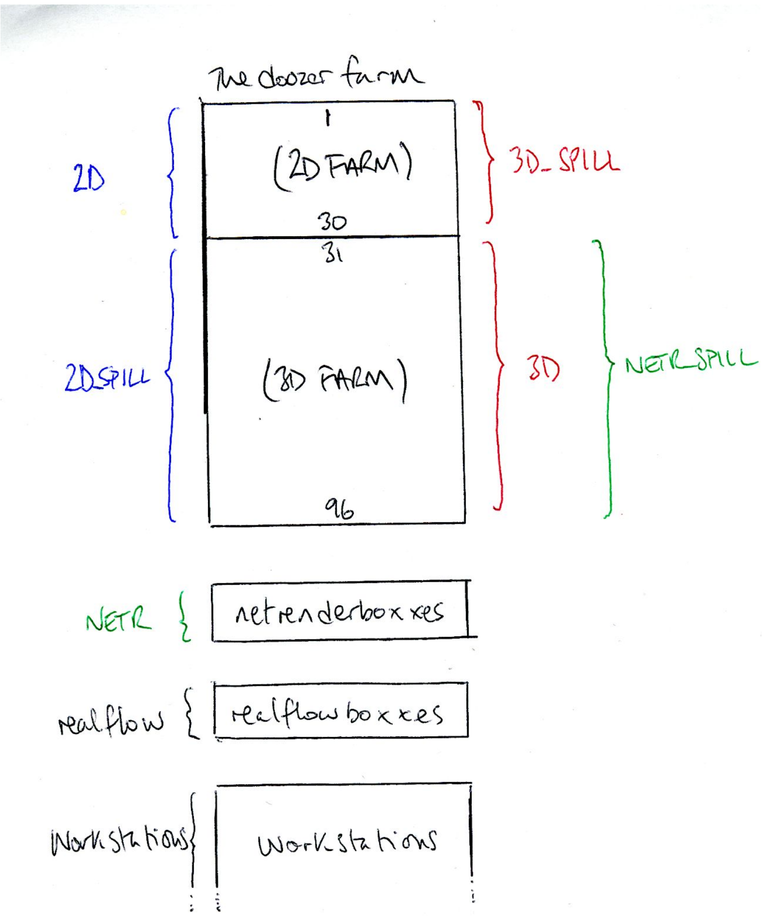

Figure 3: Render Farm Crew Setup

1. Increase efficiency

There was a feeling from both 3D and 2D artists that the divide of the farm was too absolute and that it often led to one side or the other being empty when in fact it could be put to good use.

The 2D artists need quick access to machines during the day for fast rendering but during the night they do not usually require the farm at all - 3D jobs should use this part of the farm at night. 2D also wanted to be able to use the 3D part of the farm should their part be full and the 3D part be empty during the day.

This was implemented using the idea of **spill over** or **overlapping crews**. Essentially every crew has a corresponding “_Spill" version of itself that maps to another part of the farm but that has a much lower priority. Figure 3 illustrates this. Note that 2D jobs will preferentially go to the 2D crew (this has a high priority) but that if this is full they may spillover into the 3D part of the farm via the 2D_Spill crew. This will happen with a much lower priority and so real 3D jobs will win in a fight for machines. In the same way, 3D jobs will preferentially gain access through the 3D crew but they can spill over into the 2D farm via the 3D_Spill crew. This is limited to only happen during the night (midnight - 7am) by the crew’s time group property.

2. Increase automation

The artists noted that though the wranglers are vigilant and skillful in prioritising, it would be nice to be able to automate as much as possible since it just isn't physically possible to watch the farm every second of the day. The main idea here is that we have a **“base line”** setup that **does not change** and that guaranties a certain level of service and farm behaviour.

It means for example that we can automatically make the entire 2D part of the farm available to 3D jobs at nighttime by time-restricting the 3D_Spill crew to working only between the hours of midnight and 7am. This is necessary because we do not want long 3D jobs to clog up the 2D farm.

It is on top of this base line that we build new things to deal with emergencies and special cases.

3. Easily give priorities

The base line priorities are set very low. This gives us far more lea way when trying to rush jobs through. Alfred works with a kind of complicated system that involves all sorts of percentages and fractions. We've worked it all out and honestly, you don't want to go into this...you'll have to take our word for it...its the best way.

It’s also important to note that the priority of the spill over crews are always less than the "native" 2D and 3D crews.

4. Allow for netrendering during the day

The special "netrenderboxxes" crew is set up for this. It is only made available to lighters. Also this crew spills over into the 3D part of the farm and the workstations if it is not being used.

5. Easily and quickly deal with emergencies and special cases

This is done by the creation of special crews on the fly. These work on top of the existing ones but they can be made available to specific people and can be given a very high priority. They are deleted after they have been used.

6. Make the most of resources available

The doozer farm is currently quite small. All the workstations in the building are of pretty much the same specifications as the doozers and so there is a crew called "Workstations" which is available to everyone at all times. At nighttime all users **must** logout so that these machines automatically become part of the render farm, thus making the most of all CPUs in the building.

The main benefit will be felt at night.

Conclusion

The farm setup analysis was undertaken over a period of several weeks, throughout which heated debate slowly eliminated design bugs in the system. Over this period the wranglers also discovered considerable Alfred functionality that was not previously put to good use.

It is felt that through the process of consultation, analysis, design, implementation and the testing carried out so far, that we have come to a farm setup that is optimized and fair. Importantly it is also easily and quickly manageable by the wranglers and has a large degree of automation built in by default.

Happy rendering.