Dylan Zaragoza

Work Queue & Google App Engine
Overview

• Render a video of images in a fractal structure

• Uses the Mandelbrot set

• Compare performances between:
  1. Work Queue framework & the Notre Dame Condor pool
  2. Google App Engine
Work Queue Program

**Usage:** python project.py <num_frames> <num_workers>

*<num_frames>* = number of frames in the video

*<num_workers>* = number of work queue workers

**Functions:**

- main()
- workQueue(num_frames)
- animate()
- play()
- cleanFrames(num_frames)
- cleanWorkQueue()
Work Queue Flowchart

project.py ➔ mandel ➔ master

video ➔ ffmpeg ➔ m frames ➔ catalog ➔ worker 1 ➔ worker 2 ➔ worker n

WORK QUEUE
Mandelbrot Fractal (.mpg)
## Performance Results

<table>
<thead>
<tr>
<th>Number of Workers</th>
<th>Run Time</th>
<th>Speedup</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6950.593529 s</td>
<td>1</td>
<td>100 %</td>
</tr>
<tr>
<td></td>
<td>= 1.931 hours</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>231.186616 s</td>
<td>30.0648612331</td>
<td>60.1297224662 %</td>
</tr>
<tr>
<td></td>
<td>= 3.853 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>146.842121208 s</td>
<td>47.333785918</td>
<td>47.333785918 %</td>
</tr>
<tr>
<td></td>
<td>= 2.447 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>161.802911 s</td>
<td>42.9571599549</td>
<td>28.6381066366 %</td>
</tr>
<tr>
<td></td>
<td>= 2.697 minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>159.602332 s</td>
<td>43.5494484441</td>
<td>21.7747242221 %</td>
</tr>
<tr>
<td></td>
<td>= 2.660 minutes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Google App Engine

• Platform as a Service (PaaS) model

• Hosts web applications on Google data centers

• Build an application and deploy it (<project-id>.appspot.com)

• Uses Google Cloud Storage
App Engine Application

**Design:** Web application taking user arguments

1) num_frames
2) num_workers

**Files:**

- app.yaml (cores, RAM, automatic scaling, etc.)
- main.py (request handler)
- queue.yaml

- Implementation of Task Queues
Task Queues

1. Get queue by name
2. Create task
3. Add task to queue

Problems

• App Engine typically allocates to one machine
• Scaling happens dynamically
• Worker “threads” take tasks off the queue for processing
Future Work

1. Complete/change task queue functionality
2. Determine an accurate way to compare cloud metrics
3. Add more design to the front-facing application interface
4. Allow additional user inputs to change fractal complexity
Questions?