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()

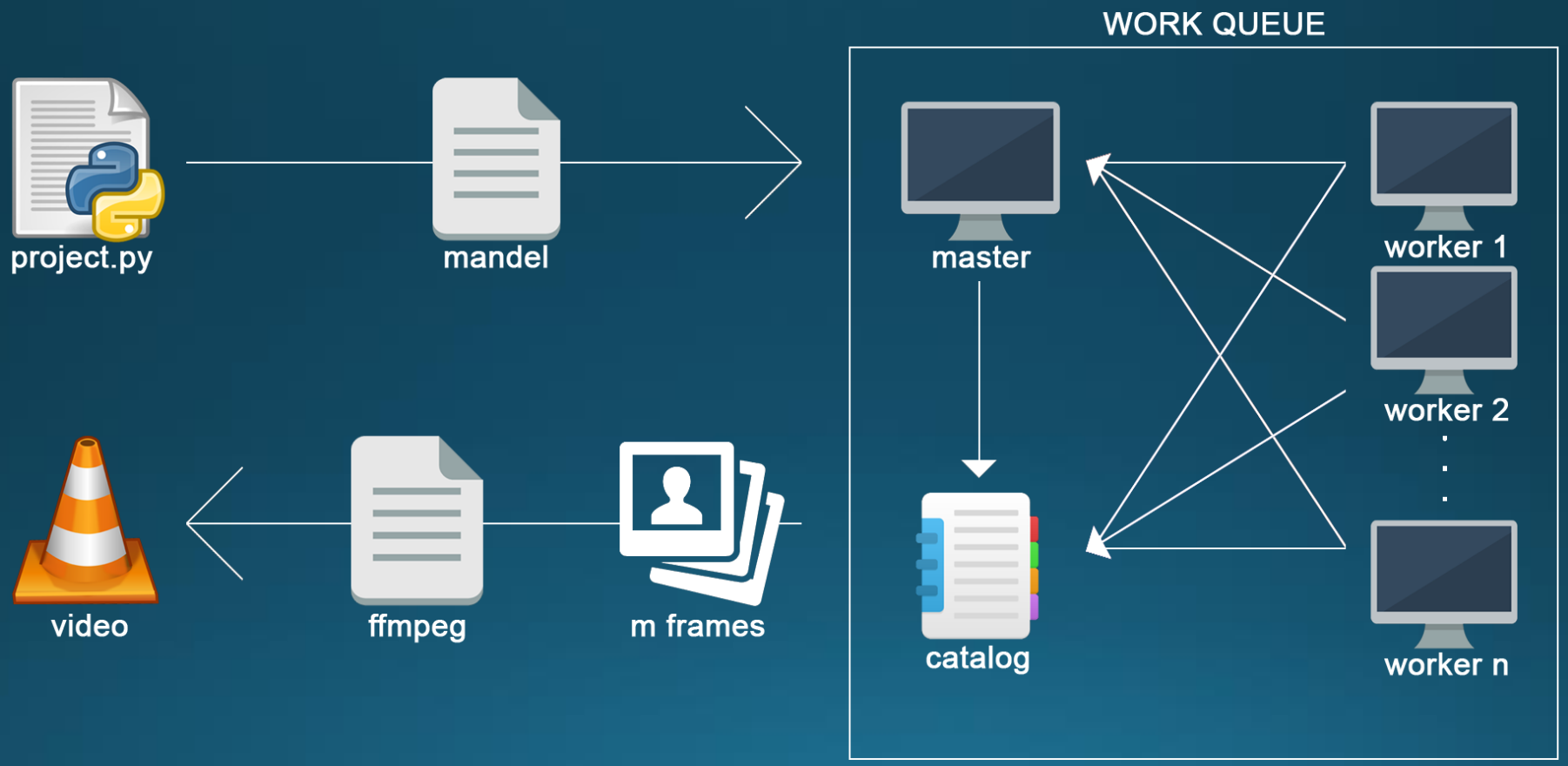
play()

cleanFrames(num_frames)

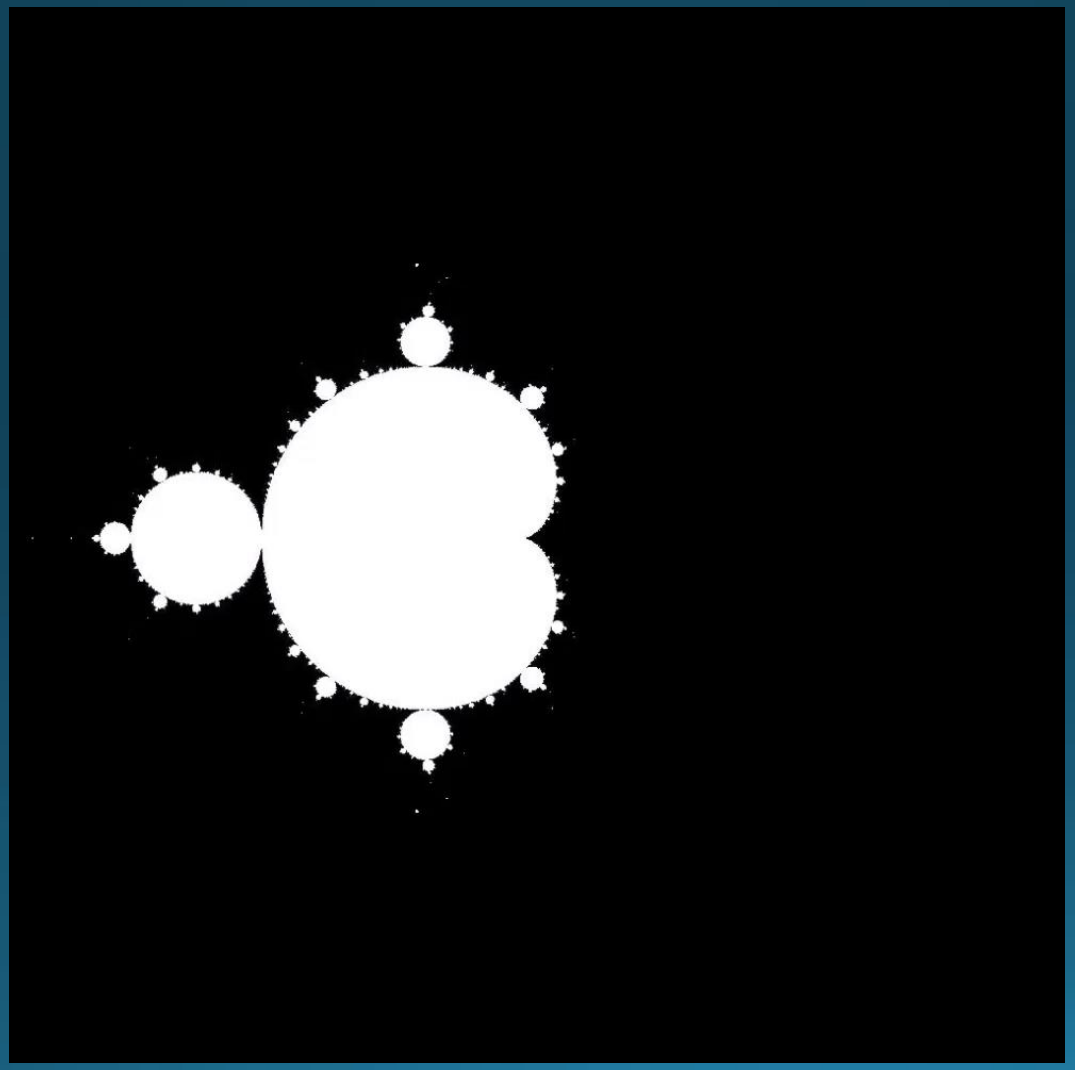
cleanWorkQueue()

```
def workQueue(num_frames):
    mandel_path = "mandel"
    if not os.path.exists(mandel_path):
        print "mandel was not found. Please modify the mandel_path variable accordingly."
        sys.exit(1);

    try:
        q = WorkQueue(port)
    except:
        print "Instantiation of Work Queue failed!"
        sys.exit(1)

    print "listening on port %d..." % q.port

    mandel = "mandel"
    sValue = 2.0
    for i in range(0, num_frames):
        outfile = "mandel%d.bmp" % (i+1)
        command = "mandel -x 0.286715 -y 0.01428 -s %s -m 2000 -W 1000 -H 1000 -o %s" % (sValue, outfile)

        t = Task(command)
        t.specify_file(mandel, mandel, WORK_QUEUE_INPUT, cache=True)
        t.specify_file(outfile, outfile, WORK_QUEUE_OUTPUT, cache=False)
        taskid = q.submit(t)
        print "submitted task (id# %d): %s" % (taskid, t.command)

        sValue *= math.exp(math.log(0.0001/2.0)/num_frames)

    print "Waiting for tasks to complete..."
    while not q.empty():
        t = q.wait(5)
        if t:
            print "task (id# %d) complete: %s (return code %d)" % (t.id, t.command, t.return_status)
            if t.return_status != 0:
                None
    print "All tasks complete!"
```

Work Queue Flowchart



Mandelbrot Fractal (.mpg)

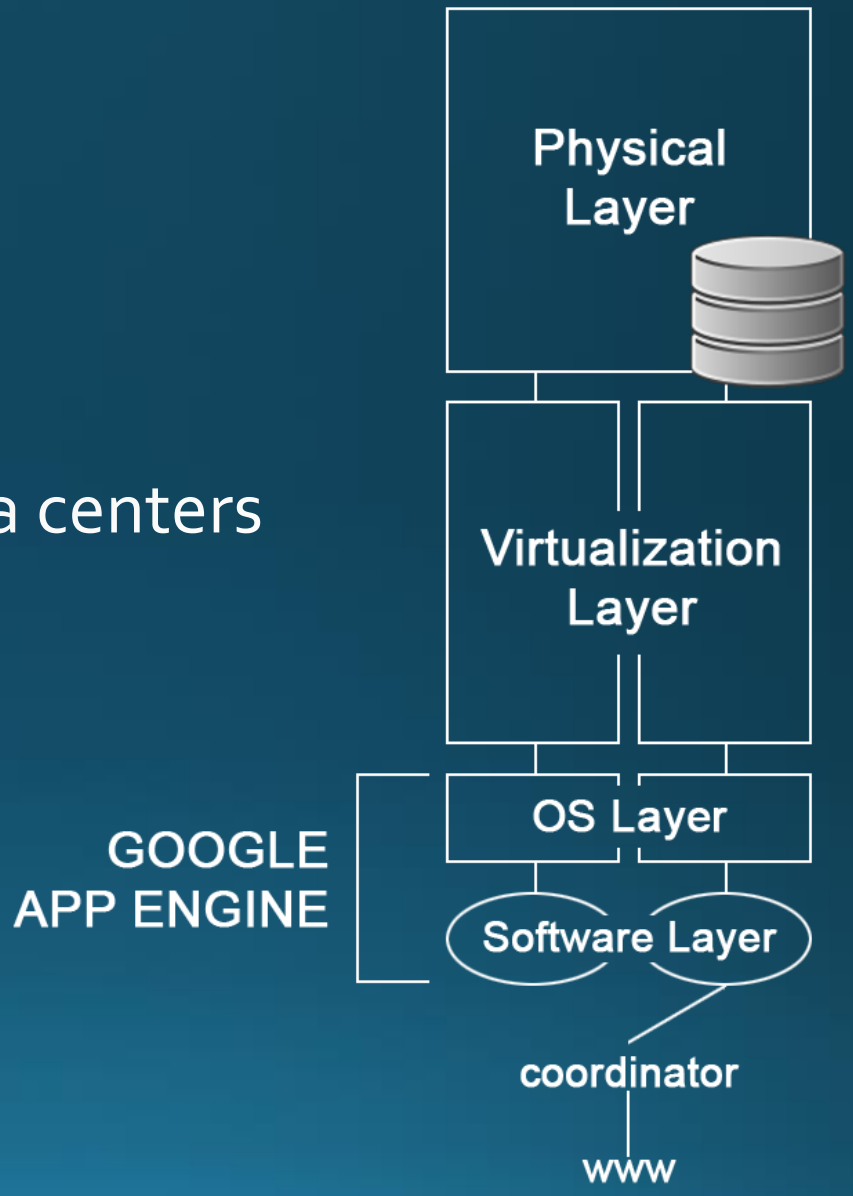


Performance Results

Number of Workers	Run Time	Speedup	Efficiency
1	6950.593529 s = 1.931 hours	1	100 %
50	231.186616 s = 3.853 minutes	30.0648612331	60.1297224662 %
100	146.842121208 s = 2.447 minutes	47.333785918	47.333785918 %
150	161.802911 s = 2.697 minutes	42.9571599549	28.6381066366 %
200	159.602332 s = 2.660 minutes	43.5494484441	21.7747242221 %

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

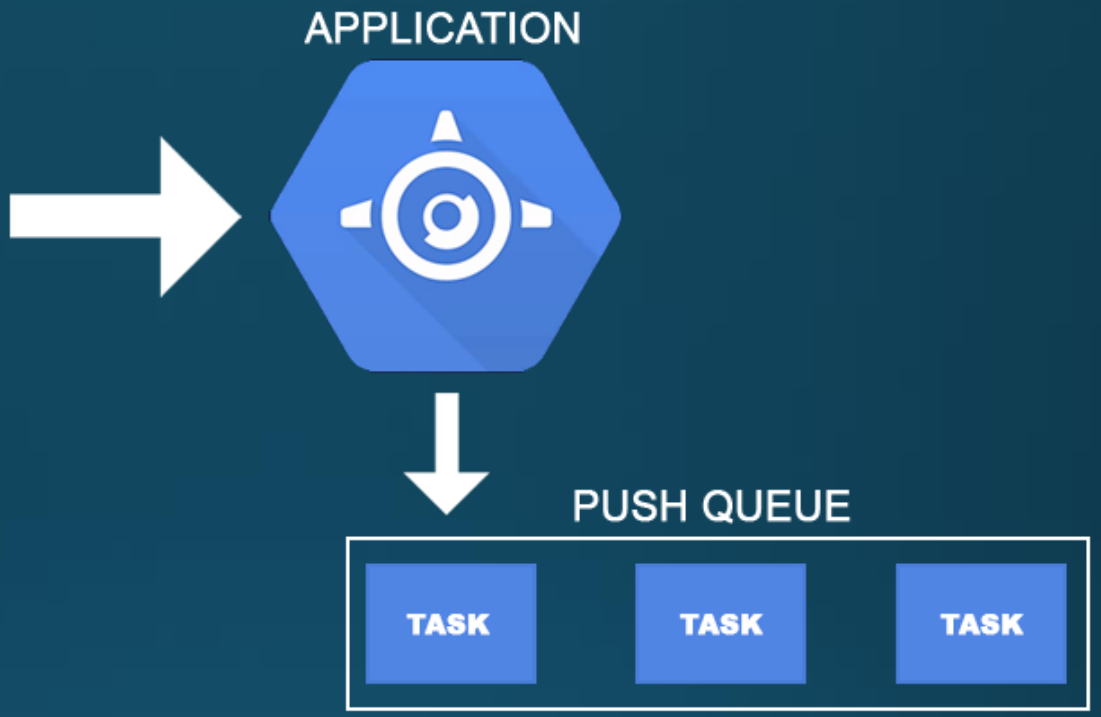
Files:

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?