Scaling Up with AWS

Alan Vuong
Katie Quinn
Idea

- Create a scalable image sharing website
- When a website becomes popular, need to be able to handle more requests
- Amazon (S3, DynamoDB) to scale up
- Using Condor, PhantomJS, and Apache AB to measure performance of non-scaled vs. scaled up application
Goals

- To increase the storage space available for website
- To increase the number of requests/second that can be made
- To carefully plan out design and budgeting to ensure AWS services are used efficiently
Website Design
Initial Architecture
Website API

GET request to cherrypy:

- Returns object containing the list of image paths
- `{"pictures": [{"id": 1, "name": "name": "/Pictures/1460917065.jpg"}], "id": 2, "name": "name": "/Pictures/1460917075.jpg"}] "result": "success"}

POST request to cherrypy:

- Returns object acknowledging success and image path
- `{result: "success", file: "/Pictures/1460920228.jpg"}`
AWS: S3 and DynamoDB

S3
- Store objects in buckets
- Uses replication of at least 3 copies
- High availability, weak consistency

DynamoDB
- Fully managed NoSQL Database service
- Uses replication
- Optimizes availability over consistency
Scaled Up Architecture

Amazon DynamoDB

Amazon S3
Using Javascript to send requests to AWS

```javascript
var db = new AWS.DynamoDB();
var params = {
    TableName: "testing",
    Item: {
        url : {'S' : data.Location}//data.Location contains the url on S3
    }
};
db.putItem(params,function(err,data){
    if (err) console.log(err);
    else console.log(data);
});
```
Challenges

- **Browser caching javascript file**
  - Restarting the apache service with new files
  - Files not properly loaded

- **PhantomJS testing**
  - For testing, you have to have the client actually make the AJAX calls to our CherryPy server and further on S3 and DynamoDB
## Testing and Conclusions

<table>
<thead>
<tr>
<th># Users</th>
<th>Request/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33.81</td>
</tr>
<tr>
<td>2</td>
<td>32.78</td>
</tr>
<tr>
<td>3</td>
<td>30.98</td>
</tr>
<tr>
<td>4</td>
<td>30.30</td>
</tr>
<tr>
<td>5</td>
<td>28.85</td>
</tr>
<tr>
<td>6</td>
<td>28.87</td>
</tr>
<tr>
<td>7</td>
<td>28.11</td>
</tr>
<tr>
<td>8</td>
<td>27.65</td>
</tr>
<tr>
<td>9</td>
<td>27.3</td>
</tr>
<tr>
<td>10</td>
<td>26.65</td>
</tr>
</tbody>
</table>

![Test for Base Case](image)

**Graph showing the decrease in requests per second as the number of users increases.**
What’s next?

● Testing more cases such as “POST” requests
● More testing for the scaled up version
● Add more complexity to the website, more styling
Questions?