Cloud Distribution Network -Tunesheap

Victor Hawley

Robert Lis

- Concurrently serve "large" files to large number of users who request them.
- Every popular service on the internet requires an infrastructure that can deal with such loads.
- Provide an easy-to-use API (that also scales) to request these files and their metadata.
- Build a system that can easily scale even further during periods of high usage.

Problem

- Single servers cannot easily handle an influx of requests simultaneously.
- We have a working API to interface with the data.
- How do we scale the API to work at a larger scale (if our service becomes popular)?
- Single server averages:
 - 7.3 secs/successful request at 1000 simultaneous requests (100% success rate)
 - 25.7 secs/successful request at 10000 simultaneous requests (52.5% success rate)
 - This isn't good enough!

Our project - Tunesheap

- Music streaming service (similar to Spotify)
- RESTful API (Ruby on Rails) for clients to interface with songs and their metadata (JSON objects).
- Amazon (EC2, S3, CloudFront, RDS) to scale the simple implementation to something that can handle an influx of requests.
- iOS client as a proof-of-concept client.











Setup (in detail)

- EC2 load balancer to split up web requests among multiple web servers.
- Web servers configured with nginx to communicate with the app servers
- Another load balancer for the app servers.
- Ruby on Rails API running on for app servers.
- S3 for song storage
- CloudFront to speed up retrieval of songs.
- Can add more systems as needed.
- Puppet and custom OS images used for deployment of machines.

Example API call

GET: /api/v1/artists

Response:

```
"artists": [
      "id": 4,
      "name": "Jay-Z",
      "country": "USA",
artist-picture"
```

			0.0	0.0	Ċ			۵	
student Timetable Scopes (Part ng-news	letter MPEG compres.	n technique 🕨 iOS	Programrd) - YouTube	CloudKit Dashboard	CAPPTIVATE.cUI Anima	tions Move it to a c	different place.		
H Type Refere rbenv: rails: c	osx - Rails 3	RDS · AWS Co	RDS · AWS Co	Rails 3 datab	Rails Applica	on rails - Pos	linux - sh: 0:	API documen	
Resources									
Albums									
Albums stored in the database									
Resource			Description						
GET /api/v1/albums		L	ist all albums						
GET /api/v1/albums/:id		c	Get album with the sp	ecified identifier					
POST /api/v1/albums		C	Create new album						
PUT /api/v1/albums/:id		ι	Jpdate existing album	with the specified io	lentifier				
DELETE /api/v1/albums/:id		[Delete existing album	with the pecified ide	ntifier				
Albums-songs									
Albom - songs relationship									
Resource			Description						
GET /api/v1/albums/:album_id/songs		L	ist all songs from the	album with the spec	ified identifier				
GET /api/v1/albums/:album_id/songs/:id		(Bet song with the spe	cified identifier, from	the album with the spe	cified identifier			
POST /api/v1/albums/:album_id/songs		C	Create new song or at	tach already exising	one to the album with	the specified ident	ifier		
PUT /api/v1/albums/:album_id/songs/:id		ι	Jpdate exising song v	vith the specified ide	ntifier from the album v	vith the specified is	dentifier		
DELETE /api/v1/albums/:album_id/songs/:id	I	F	Remove existing song	with the specified id	entifier (removes relation	onship not the son	g object itself)		
Artists									
Artists									
Artists Artists stored in the database Resource			Description						

```
"description": "New York native. ",
"dob": "1969-12-04 00:00:00 UTC",
```

```
"website": "www.rocafella.com",
```

"picture url": "https://tunesheap-content.s3.amazonaws.com/4-

Testing/Conclusion

- Python script utilizing work_queue and condor to send HTTP requests and time the results.
- Measuring scaled version of the app vs. an implementation using a single server
- Each individual request's results are used to calculate the total performance (aggregate time for all requests to finish, including overlap)
- We are testing a wide range of the amount of requests and amount of workers performing those requests
- Still finalizing the infrastructure of the system, but major improvements are expected.

What's next?

- Finalize infrastructure and gather data
- memcached
- Elastic search
- Scale the database.

Questions?