MICHIGAN STATE UNIVERSITY Project Plan Presentation AWSome Availability Zones The Capstone Experience

Team Amazon

Huang, Wynton Chak, Jung Kim, Iris Heiner, Jamison Hood, Jake

Department of Computer Science and Engineering Michigan State University

From Students... ...to Professionals Fall 2021

Functional Specifications

- Gather information on network latency between Availability Zones (i.e., data centers)
- Assist users with optimizing network latency in their applications by suggesting AZ pairs
- Interact with data through easy-to-use UI
- Filter through results based on region-based requirements (ex. no AZs in Europe, one must be in Western US, etc.)

Design Specifications

- User friendly + interactive map showing all Regions in the world. Each Region is clickable and will show more details about each Availability Zone it contains
- Filtering system where users can add, edit, and remove Region-based requirements for AZ placement
- Once a Region is selected, all other Regions are colored on main map webpage based on average network latency
- Statistics webpage for any pair of Availability Zones showing a graph of network latency tests and highlighting important features like average latency, 95th, 99th, and 99.9th percentiles

Screen Mockup: Interactive Map



Screen Mockup: Select Region



Screen Mockup: Regional AZ Matchups Page

	l • <		R 0		🚊 us-east-2.quicksight.aws.amazon.com				් අම			
I YouTube O	cogie Netti	lie Homepaga - M	L.e University Hulu	End of Class Atte	ndance Linkedin	AWS Management (Console Apple D	isney+ Movies and St	idws My Driv	e - Google Drive		
W QuickSight											kimiris@msu.edu ~	
Sundo C	Recta 💩	Reset 🛧 Te	am Amazon					📄 Export	🔂 Data	:)) Save as 🍈 Share	Alerts	
Map Latenci	es (US)	Latency stats										
			us-cast1		us-cost2			us-west1				
		118	PV15	N10	NL	NZ	N4	N.5	NG	N7		
	NB		0.46ms	0,67ms	88.60ms	86.83ms	87.38mi	131.77ms	134.40ms	129.57ms		
us-east1	NO	0.49ms		0.48ms	87.18ms	87.12ms	86.97ms	134.62ms	128.67ms	128.07ms		
	N10	0.45ms	0.50ms		87,43ms	87.66ms	87.17ms	129,49ms	129.79ms	135.58ms		
us-exit2	N1	87.24ms	87.04ms	87.52ms		0.52ms	6.54ms	71.05ms	71.06ma	71.30ma		
	N2	:87.43ms	87.30ms	07.29ms	0.45ma		0.43ins	69.37mi	67,97ms	71.06ms		
	N4	89.01ms	87.27ms	87.61ms	0.50me	0.52mi		68.01ma	69.46ms	68.09ms		
us-west1	NS	133.92ms	130.96ms	130.81ms	63.61ms	69.56ms	69.52ms		0.40ms	0.48mz		
	NG	131.46ms	130.56ma	132.06ms	71.35m\$	67.95ms	69.44ms	0.43ms		0.53ma		
	N7	133.68ms	134.59ms	131.45ms	68.37ms	71.09ms	70.01ms	0.43ms	0.47ms			

Screen Mockup: Specific AZ Pair Statistics Page







Technical Specifications

- Front End
 - QuickSight Dashboard
 - GitHub Pages
 - Angular web application
- Back End
 - AZ Matching Algorithm
 - ICMP vs. TCP
 - High Precision Timer
 - EC2 automatic deploy

System Architecture



System Components

- Hardware Platforms
 - = N/A
- Software Platforms / Technologies
 - Front End
 - Angular, Angular Materials, GitHub Pages
 - AWS Services (IAM, Lambda, API Gateway, QuickSight)
 - Back End
 - O AWS CDK, CLI, SDK
 - Amazon TimeSync
 - Chrony, C, TCP, DNS, HR Timer, NTP

Risks

- Back end/Front end Communication
 - The back end database is likely to change over time, and these changes will affect the front end
 - API interface between front and back to control when communication happens
- Timing methods and Precision
 - Latency between AZs is naturally low, so timing latency needs to be precise and consistent
 - Timing using clock ticks for precision, and consistency achieved with time
- IP Address Security
 - Deployment of our detection method requires knowledge of IP addresses, which may be inconsistent and a security risk
 - Deploy Swarm Conductor first
- Unknown QuickSight Functionalities
 - QuickSight might have/lack functionalities that will interfere with our planned display
 - Use and experiment with embedded versions on Angular to create workarounds

Questions?

