

HKBU-ITM -Comp 7940

Group project- Line bot Group I

Milestone 4

Q1: How is your project architecture related to the theory taught in the lecture?

For the project architecture, our line chatbot is under three-tier cloud based architecture. Under this model, the first tier refers to the user view and control with accessing by end user device. In our project, the line chatbot conversation and user interface setting accessing by end user mobile devices will be first tier.

The second tier refers to application server which application logic for the solution. In our project, Heroku considered as a cloud application server for performing the application logic and chatbot responds according to our source code for the line chatbot.

The third tier refers to the database server which is database manager for the solution. In our project, Redis will be the database server and database manager. It has contented the coronavirus info for example the mask inventory level, measure video and the updated numbers of coronavirus case.

Beside the three tier architectures, we have to define the roles and responsibilities. There are two architectures styles helping us to define the role and responsibility, which are client-server and peer-to-peer respectively.

For client-server model, client will interact with the server for data throughout the internet. And our project is under this architecture style since end users earn the data by sending request to our server through line chatbot user interface.

For peer-to-peer model, by definition, no devices or peers consider as client or server since they will share data or resource with each peer. And peer-to-peer is not applicable to our project as there are not communication between users.

Q2: Can you demonstrate, with some screen cap, how to increase capacity of your chat bot service?

Linebot need to face customer in the front-end and require replying user as soon as possible, we need to sure customer waiting time would not increase while capacity grow. We would review our current services from front-end to back-end to figure out which part can improve to increase capacity.

Line

The most front is Linebot, consider our app design only use reply function and no active message requirement, so we can still stay at free account plan and no need to upgrade as Fig 1. Second part is review webhook between Linebot and our server. Because there is no modification place on Linebot webhook setup, we only can reduce webhook response time on service side.

LINE Official Account Subscription Plans

	Free	Light*	Standard*
Monthly Fee	Free	50 USD	150 USD
Free Messages	Up to 500	Up to 15,000	Up to 45,000
Additional Message Fee	N/A	0.05 USD	0.03 USD

*Light plan and Standard plan are not available in USA, Singapore, EU

Fig 1) Line product charge

Server

Reviewed the server side, the most time-consuming parts are python program is a single thread program which reply request one by one, connection to Redis cloud and collect data from DATA.GOV.

For the single thread problem, we can change app.run function to insert new parameters "threaded = True, processes = NumberOfProcesses" to immediately handle multiple requests at the same time. Reference Heroku fee as Fig 2, we need to select higher than Standard 1x plan to get unlimited process.

However, Standard plan is not dedicated machine which mean context-switch existed while machine changing user to us and cost time delay. Consider further capacity growth continually, we should select Performance M plan to has dedicated machine, unlimited process and horizontal scaling function, which included SSL load balance, for further usage.

There also have multiple connection problem to Redis cloud because our python change to multithread process and each child process would call Redis cloud individually. However, we are using Redis cloud free account that only cater request 30 as maximum conversation line bot can handle at the same time. Because Redis cloud connection is another delay part and we can not promise connection status and performance between Heroku datacentre and Redis cloud, we suggest using local Redis to store data in own dedicated machine memory to

save Redis cloud fee, increasing response time and completely control well-known reliable Redis database with ourselves.

Combine with third DATA.GOV problem, we can directly access a machine frequently check the version of DATA.GOV then directly store the processed message in Redis instead of each request handler connect to DATA.GOV individually. More than that, DATA.GOV did not promise work fine anytime and happened change DATA format or API command without notify in advance, so current application might cause error while try to get data from DATA.GOV. Using a single program to collect can send alert email while get data fail and keep the latest correct data in database so application still can reply to user directly.


	 Free \$0	 Hobby \$7/dyno per month	 Standard 1x \$25/dyno per month	 Standard 2x \$50/dyno per month	 Performance M \$250/dyno per month	 Performance L \$500/dyno per month
	PROFESSIONAL					
What is it good for?	Ideal for experimenting with cloud applications in a limited sandbox.	Perfect for small scale personal projects and hobby apps.	Enhanced visibility, performance, and availability for powering your production applications.		Superior performance when it's most critical for your super scale, high traffic apps.	
RAM	512MB	512MB	512MB	1GB	2.5GB	14GB
Deploy from Git	•	•	•	•	•	•
Automated OS patching	•	•	•	•	•	•
Unified logs	•	•	•	•	•	•
Number of process types	2	10	Unlimited	Unlimited	Unlimited	Unlimited
Always on	Sleeps after 30 mins of inactivity, otherwise always on depending on your remaining monthly free dyno hours.	•	•	•	•	•
Custom domains	•	•	•	•	•	•
Free SSL on custom domains		•	•	•	•	•
Automated Certificate Management on custom domains		•	•	•	•	•
Horizontal scaling			•	•	•	•
Preboot			•	•	•	•
Application metrics		Past 24 hours, with 10 minute resolution	2 hours at 1 minute resolution, 24 hours at 10 minute resolution, 3 days with 1 hour resolution, 7 days at 2 hours resolution			
Language runtime metrics			•	•	•	•
Threshold alerting			•	•	•	•
Autoscaling					•	•
Dedicated	No	No	No	No	Yes	Yes
Combining multiple Dyno types	Cannot combine with other dyno types	Cannot combine with other dyno types	Can combine with Performance dynos	Can combine with Performance dynos	Can combine with Standard dynos	Can combine with Standard dynos

Fig 2) Heroku service charge

Back to local Redis, table 1 show currently our application store data in Redis which only need approximately 150 MB and 1GB of 10,000 and 100,000 customers respectively. Heroku also provide internal Redis addons service as Fig 3 but has few problems. First problem is \$200 per month is very expensive while we can assign 1GB memory to store data in 2.5GB memory local machine of Performance M plan. Second problem is Redis service is not related on your machine region but unique datacentres and only in US East and Europe, its quite far to Line popular area Asia. The final problem is Heroku Redis use URL to connect service which mean potential weakness point because Redis is practiced reliable database, but network connection is not, and performance would lower than local machine version. Even our customers grow over 10,000, we can still upgrade to Performance L plan to 14GB memory which only increase \$250 per month but Heroku Redis need \$1450 per month or \$1750 per month with private link network. Therefore, we still suggest that install Redis in dedicated machine to reduce delay, connection limitation and lower cost.

Item	Data Type (bytes)	Usage
Each customer usage		
Line User_id	Char (1)	Store user in which question
User State	Integer (8)	Store user in which step of question
Fixed global usage		
Measure video URL and thumbnail list	Set (65 x 2 x 4 videos = 520 bytes)	Store video and thumbnail URL in set which store in Google Drive
Mask list	Set { code: Address code: Number of Mask code : Full name } ((255 + 8 + 255) * 102 items = 51 MBytes)	Store mask store holding of each distinct
Latest Message from DATA.GOV	Char(400)	Store processed reply message of from DATA.GOV data

Table 1) Estimate Redis data usage

Google Drive

Video and thumbnail store in Google Drive might call limit by free edition only support 100 user access at the same time. Even the chance is low because user take time to watch the video, we still can use Business Google Drive as \$12 per month to get unlimited connection as Fig 4.

Plans & Pricing

Hobby Dev	Free	Memory	1 GB
Premium 0	\$15/mo	Connection Limit	1,000
Premium 1	\$30/mo	High Availability	✓
Premium 2	\$60/mo	Persistence	✓
Premium 3	\$120/mo	Performance Analytics	✓
Premium 5	\$200/mo	Metric Logs	✓
Premium 7	\$750/mo	Heroku Redis via PrivateLink	
Private 7	\$900/mo		
Premium 9	\$1450/mo		
Private 9	\$1750/mo		
Premium 10	\$3500/mo		
Private 10	\$4000/mo		

Install Heroku Redis

```
$ heroku addons:create heroku-redis:premium-5
```

To provision, copy the snippet into your CLI or use the install button above.

Fig 3) Heroku Redis service charge

Basic

\$6

USD / user / month

START FREE TRIAL

INCLUDED APPLICATIONS

Gmail Business email

Hangouts Meet Video and voice conferencing (for up to 100 participants per call)

Hangouts Chat Secure team messaging

Calendar Shared calendars

Drive 30 GB cloud storage

Docs Word processing

BEST VALUE Business

\$12

USD / user / month

START FREE TRIAL

INCLUDED APPLICATIONS

Gmail Business email

Hangouts Meet Video and voice conferencing (for up to 150 participants per call)

Hangouts Chat Secure team messaging

Calendar Shared calendars

Drive **Unlimited** cloud storage (or 1TB per user if fewer than 5 users)

Enterprise

\$25

USD / user / month

START FREE TRIAL

INCLUDED APPLICATIONS

Gmail Business email

Hangouts Meet Video and voice conferencing (for up to 250 participants per call)

Hangouts Chat Secure team messaging

Calendar Shared calendars

Drive **Unlimited** cloud storage (or 1TB per user if fewer than 5 users)

Fig 4) Google Drive charge

In the conclusion, we suggest applying python multithread server handling, \$250 per month Heroku machine, install local Redis, individual process to collect DATA.GOV data to Redis for global access and \$12 per month business Google Drive service to handle increase capacity.

Q3: Can you identify if your bot is one of the examples of PaaS, IaaS, SaaS? Explain your answer.

By definition, SaaS (Software as a Service) only allowing users access to front-end interface and configuring cloud service in based on the usage and usage-related configuration. For examples: Salesforce, Slack, DocuSign etc.

And for PaaS (Platform as a Service), it has including the consumer activities of SaaS. Meanwhile, it is allowing users not only access to front-end interface, also end user can develop, test, deploy under a pre-configures platform with limited administrative features. For example: Force.com, Heroku etc.

And for IaaS (Infrastructure as a Service), it allows end users to set up and configures bare infrastructure which basically provisioning the processing, storage, networking on Cloud solution provider for example: AWS EC2, Rackspace etc.

For our line bot project, it has classified as SaaS from an end user prospective. The reason of our product considers as SaaS, is that it only allowing end users requesting coronavirus information which is our main feature of the line bot and access to front-end user interface. Also, by configuring user setting, end users can use extra features or configurations, for example recommending the chat bot to other users, set remainder when there is message from line bot, or even changing the background of the conversation history in the line bot.

Also, from line bot developer prospective, line chatbot/line developer is considered as PaaS since it has provided a platform for us to develop, design, deploy and test the customized application under a pre-configures platform, and a specified API to user.

And our product cannot classify as IaaS since end user will not involve in any set up on the infrastructure. And even from line bot developer prospective, our line bot is not allowing developer to provisioning the storage networking etc., so for sure it cannot consider as PaaS from developer prospective as well.