MODULE CODE: ISYS40061

MODULE NAME: Service-Oriented Cloud Technologies

NAME: Dayeeta Das

NTU ID: N0830182

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

CONTENTS Pg. No.

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

Chapter 1 – Self Evaluation of achieved implementation

1.1. Schematic Diagram

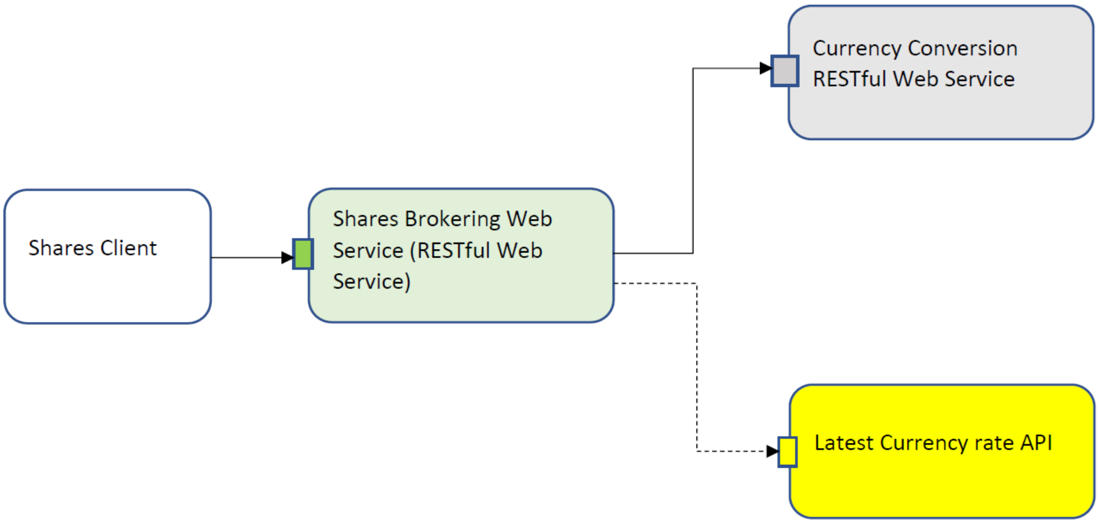


Figure 1: The image above shows the schematic diagram representing the achieved implementation.

Chapter 2 – Challenges faced by SOA (798 words)

SOA, or service-oriented architecture, defines a way to make software components reusable and interoperable via service interfaces. Services use common interface standards and an architectural pattern so they can be rapidly incorporated into new applications. This removes tasks from the application developer who previously redeveloped or duplicated existing functionality or had to know how to connect or provide interoperability with existing functions (IBM, 2019).

2.1. Challenges Faced by SOA

A true SOA application should scale easily as far as the application architecture is concerned. A SOA application has two components: service components and client applications. The client application may be a Web application, another service or any other application that relies on the SOA service components to do its job (kexugit, n.d.).

A score concept of SOA is to disintegrate an application into smaller chunks so that these can be run as independent services over multiple servers. These services should ideally be stateless meaning that they can run on multiple computers without retaining any previous data. Due to this reason, the architecture of a SOA is inherently scalable. However, data access can lead to scalability bottleneck.

Bottlenecks typically involve application data which is stored in some database like a relational database. In case of this coursework, the application data has been stored in a json file which contains a json object containing an array with the details of different companies. Storage of session data can cause problems with scalability. One SOA application relying on another can cause issues with the performance increasing response time and affecting scalability. In the coursework the Shares Brokering RESTful Web Service that has been created and called by the Shares Client depends on the external Currency Conversion RESTful Web Service for getting the conversion rates and for implementing the currency conversion features. The external Currency Conversion RESTful Web Service would have in turn depended on an external API for getting the latest currency rates. Such data trips can be costly. If these services need to be called repeatedly, it makes it challenging to effectively scale the application leading to scalability bottlenecks. Increasing concurrent client connections under such situations can lead to long response time. A client should be to maximize the number of concurrent messages that it sends to the broker without overloading it. For the resource units if there are four nodes in the cluster with four cores each, then 16 messages should be sent. If the messages are short, consider sending additional messages so that there are always messages available to the broker in the queue (docs.microsoft.com, n.d.). If the messages are sent asynchronously, the client will receive a TimeoutException exception.

2.2. Addressing the issues

Resource Provisioning is a major challenge. Service providers struggle to maximize the resource utilisation while minimising the financial cost for delivering them to the users for the execution of a particular service. Cloud computing can provide a good solution for it. By considering SLA, resource provisioning providing services to the cloud users. This is preliminary concurrence between the cloud users and cloud service providers which ensures Quality of Service (QoS) parameters like performance, availability, reliability, response time etc. Based on the application needs Static Provisioning/Dynamic Provisioning and Static/Dynamic Allocation of resources must be made in order to efficiently make use of the resources without violating SLA and meeting these QoS parameters (Narale and Butey, 2015).

Based on application requirements, static provisioning can be used for unchanging workloads for applications. Dynamic provisioning can be used when the provider needs to add or remove resources based on the needs of the application. Users can use self-provisioning for purchasing resources from the cloud provider directly through a webform. The buy and sell feature of the coursework can be a feature of self-provisioning as well as dynamic provisioning.

2.3. Challenges of migration to Cloud and Containerisation

Major challenges faced when migrating legacy applications to cloud are the lack of good strategy, transition from physical servers to cloud makes data vulnerable, moving large data from physical servers to cloud can cause data outages and data loss if it is not backed up properly, all applications might not fit well with cloud after migration and initial migration can incur a high financial value.

Containers have become a key technology for many organisations migrating to the cloud. Used to package an application with its configuration and OS dependencies, they ensure software runs reliably when moved from one computing environment to another. As well as improving portability, they make it easier to leverage cloud benefits such as cost-optimisation, enhanced operability and scalability on-demand (DevOpsGroup, n.d.). Containers have lower infrastructure costs and security optimised node kernel.

Physical QoS was performed on the developed APIs using JMeter. The screenshots of the results have been provided in Appendix A.

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

Chapter 3 (600 words)

An Integrative Semantic Framework for Image Annotation and Retrieval

Semantically enabled image annotation and retrieval engine uses methodically structured ontologies for image annotation, gets accurate results for matchmaking queries. Defines ontology as the shared conceptualisation of a domain consisting of comprehensive set of concept classes, the relationships among them and information showing how they are populated in the application domain. Caption-based semantic annotation, semantic annotation of images allows retrieval engines to make smarter decisions about similarity of an image retrieved to a given query. Content-based semantic annotation, image recognition is used. It is more complex than caption-based image retrieval hence is partially successful. Also, it is easier to use text as a query than image. For the case study, two goals were set: investigating semantic technology to build a classification and indexing system critically unifying the annotation infrastructure for all the sources of incoming photos and conducting a feasibility study aiming to improve the end-user experience for searching images using the engine. For retrieval process, images used were not from web, images needed to be relayed to the customers quickly and large number of images were annotated to a schema. For domain analysis, two trees were used: - image knowledge and image attributes. Bottom-up approach was used for populating the lower trees of the ontology class. Protégé ontology editor that adopts OWL was used to construct the sports domain ontology. Considering the dynamic motion nature of sports domain, "Actor Action Object" structure was used for motion and emotion annotations. Action/Emotion was used as search query, for multiple terms the main term was specified. Semantic annotator annotated new images and transformed user query to OWL format. Semantic reasoning engine got images with matching annotations and ranked them based on user preference. Nearest neighbour matchmaking algorithm was used for ranking.

Linked Data - The Story So Far

Linked Data is a set of design principles for sharing machine-readable interlinked data on the Web. When combined with Open Data (data that can be freely used and distributed), it is called Linked Open Data (LOD). An RDF database such as Ontotext's GraphDB is an example of LOD (Ontotext, n.d.), should be published alongside several types of metadata. Some applications are: Linked Data Browsers and search engines. Research is being carried out on Microformats, web APIs, dataspaces and semantic web.

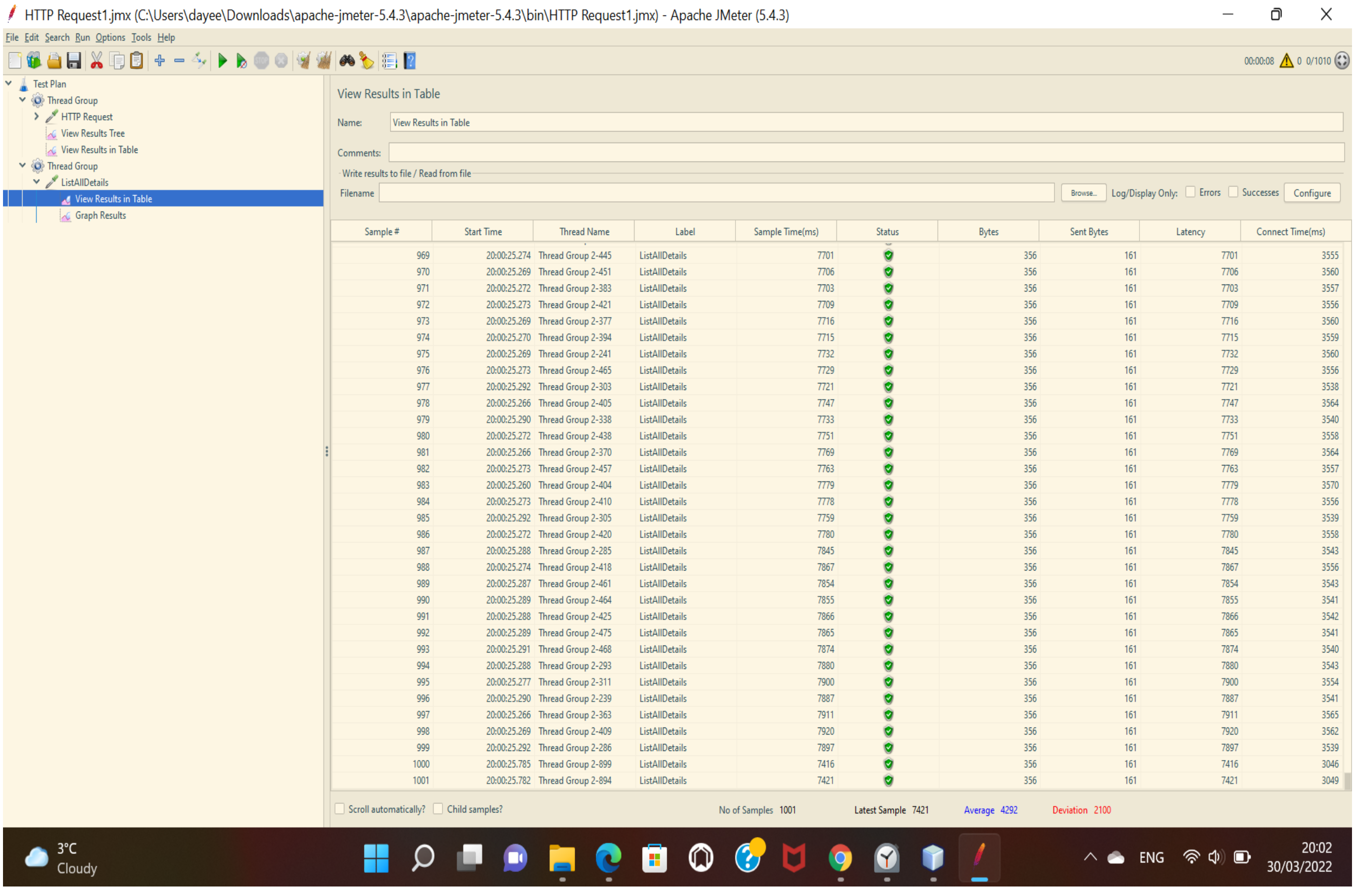
Berners-Lee (2006) outlined a set of 'rules' for publishing data on the Web in a way that it all eventually becomes a single global data space. The rules were using URIs for naming things, HTTP URIs for looking up those names, provide useful information, using standards (RDF, SPARQL) when looking up using those URIs and include links to other related URIs. The Linking Open Data Project aims to bootstrap the Web of Data by identifying existing data sets that are available under open licenses, converting these to RDF according to the Linked Data principles. Data providers add their data to a global data space by publishing data on the Web according to the Linked Data principles. They choose between two HTTP URI usage patterns to identify entities: 303 URIs and hash URIs. RDF links are used by client applications to navigate and discover additional data.

Utilising the concepts from Paper 1 and 2, a semantically enabled caption-based search engine can be used to extend the coursework functionality. "Name Symbol Currency" would be search query that would retrieve details of similar companies (example American/British company) from the Linked Open Data. Nearest Neighbour matchmaking algorithm could be used for ranking the details of these companies based on their similarity to the search query.

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

Appendix

Appendix A



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
969	20:00:25.274	Thread Group 2-445	ListAllDetails	7701		356	161	7701	3555
970	20:00:25.269	Thread Group 2-451	ListAllDetails	7706		356	161	7706	3560
971	20:00:25.272	Thread Group 2-383	ListAllDetails	7703		356	161	7703	3557
972	20:00:25.273	Thread Group 2-421	ListAllDetails	7709		356	161	7709	3556
973	20:00:25.269	Thread Group 2-377	ListAllDetails	7716		356	161	7716	3560
974	20:00:25.270	Thread Group 2-394	ListAllDetails	7715		356	161	7715	3559
975	20:00:25.269	Thread Group 2-241	ListAllDetails	7732		356	161	7732	3560
976	20:00:25.273	Thread Group 2-465	ListAllDetails	7729		356	161	7729	3556
977	20:00:25.292	Thread Group 2-303	ListAllDetails	7721		356	161	7721	3538
978	20:00:25.266	Thread Group 2-405	ListAllDetails	7747		356	161	7747	3564
979	20:00:25.290	Thread Group 2-338	ListAllDetails	7733		356	161	7733	3540
980	20:00:25.272	Thread Group 2-438	ListAllDetails	7751		356	161	7751	3558
981	20:00:25.266	Thread Group 2-370	ListAllDetails	7769		356	161	7769	3564
982	20:00:25.273	Thread Group 2-457	ListAllDetails	7763		356	161	7763	3557
983	20:00:25.260	Thread Group 2-404	ListAllDetails	7779		356	161	7779	3570
984	20:00:25.273	Thread Group 2-410	ListAllDetails	7778		356	161	7778	3556
985	20:00:25.292	Thread Group 2-305	ListAllDetails	7759		356	161	7759	3539
986	20:00:25.272	Thread Group 2-420	ListAllDetails	7780		356	161	7780	3558
987	20:00:25.288	Thread Group 2-285	ListAllDetails	7845		356	161	7845	3543
988	20:00:25.274	Thread Group 2-418	ListAllDetails	7867		356	161	7867	3556
989	20:00:25.287	Thread Group 2-461	ListAllDetails	7854		356	161	7854	3543
990	20:00:25.289	Thread Group 2-464	ListAllDetails	7855		356	161	7855	3541
991	20:00:25.288	Thread Group 2-425	ListAllDetails	7866		356	161	7866	3542
992	20:00:25.289	Thread Group 2-475	ListAllDetails	7865		356	161	7865	3541
993	20:00:25.291	Thread Group 2-468	ListAllDetails	7874		356	161	7874	3540
994	20:00:25.288	Thread Group 2-293	ListAllDetails	7880		356	161	7880	3543
995	20:00:25.277	Thread Group 2-311	ListAllDetails	7900		356	161	7900	3554
996	20:00:25.290	Thread Group 2-239	ListAllDetails	7887		356	161	7887	3541
997	20:00:25.266	Thread Group 2-363	ListAllDetails	7911		356	161	7911	3565
998	20:00:25.269	Thread Group 2-409	ListAllDetails	7920		356	161	7920	3562
999	20:00:25.292	Thread Group 2-286	ListAllDetails	7897		356	161	7897	3539
1000	20:00:25.785	Thread Group 2-899	ListAllDetails	7416		356	161	7416	3046
1001	20:00:25.782	Thread Group 2-894	ListAllDetails	7421		356	161	7421	3049

Figure 1: Table showing ListAllDetails API.

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

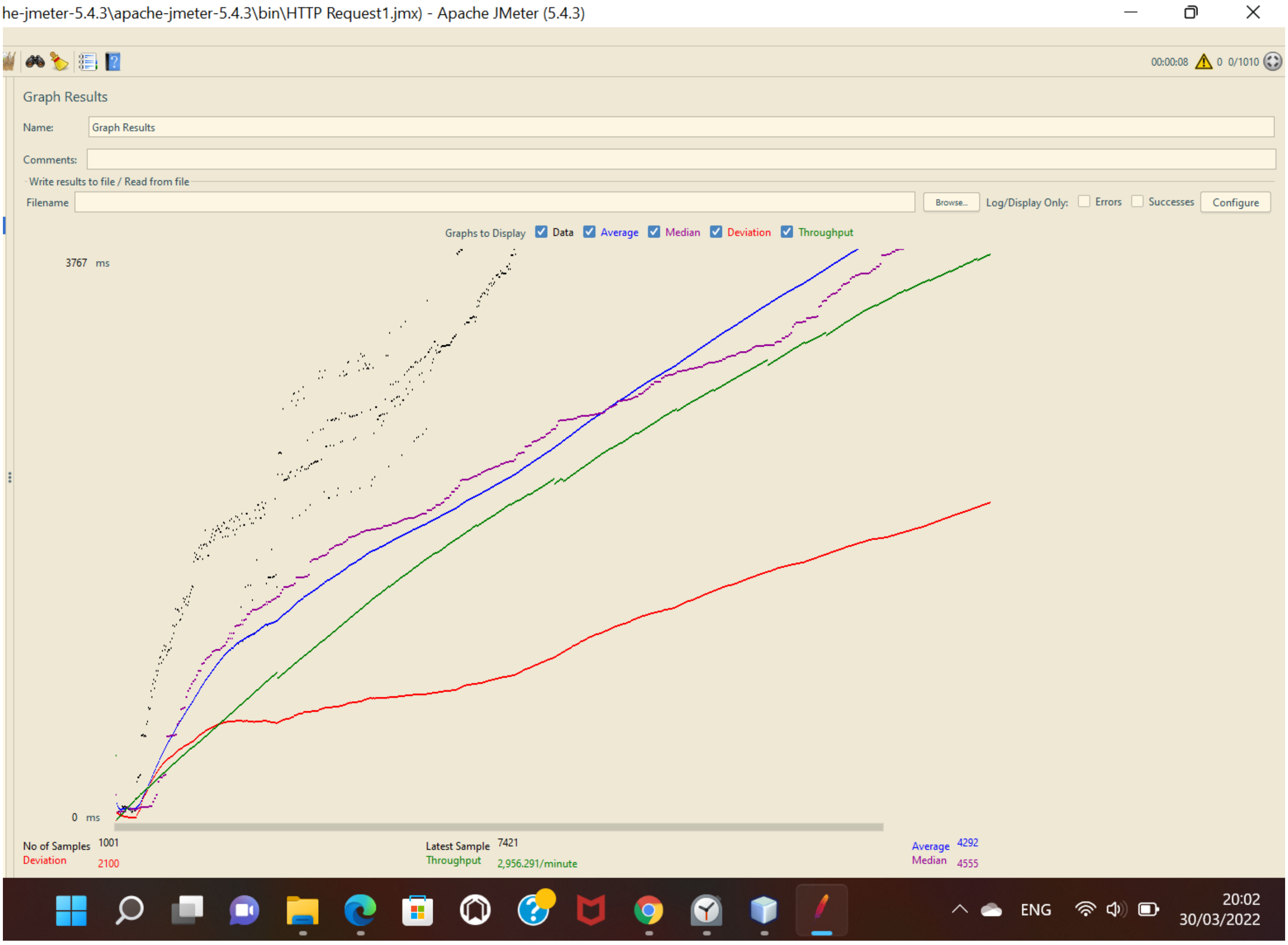


Figure 2: Graph showing ListAllDetails API.

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

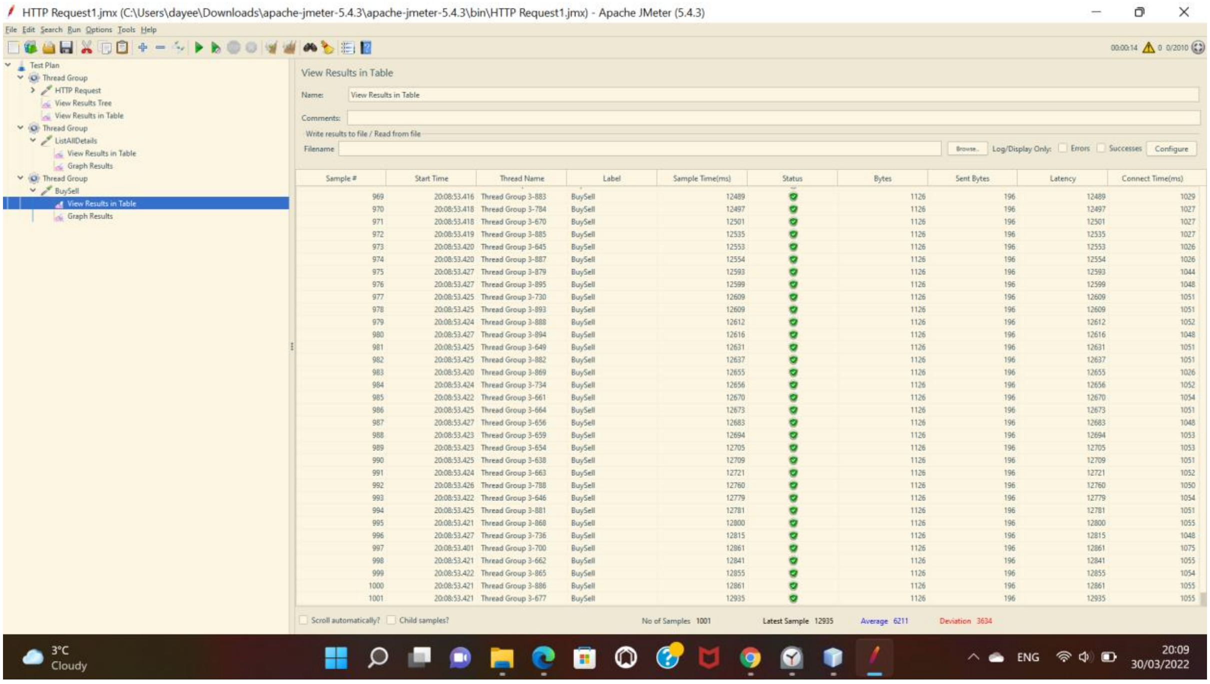


Figure 3: Table showing BuySell API.

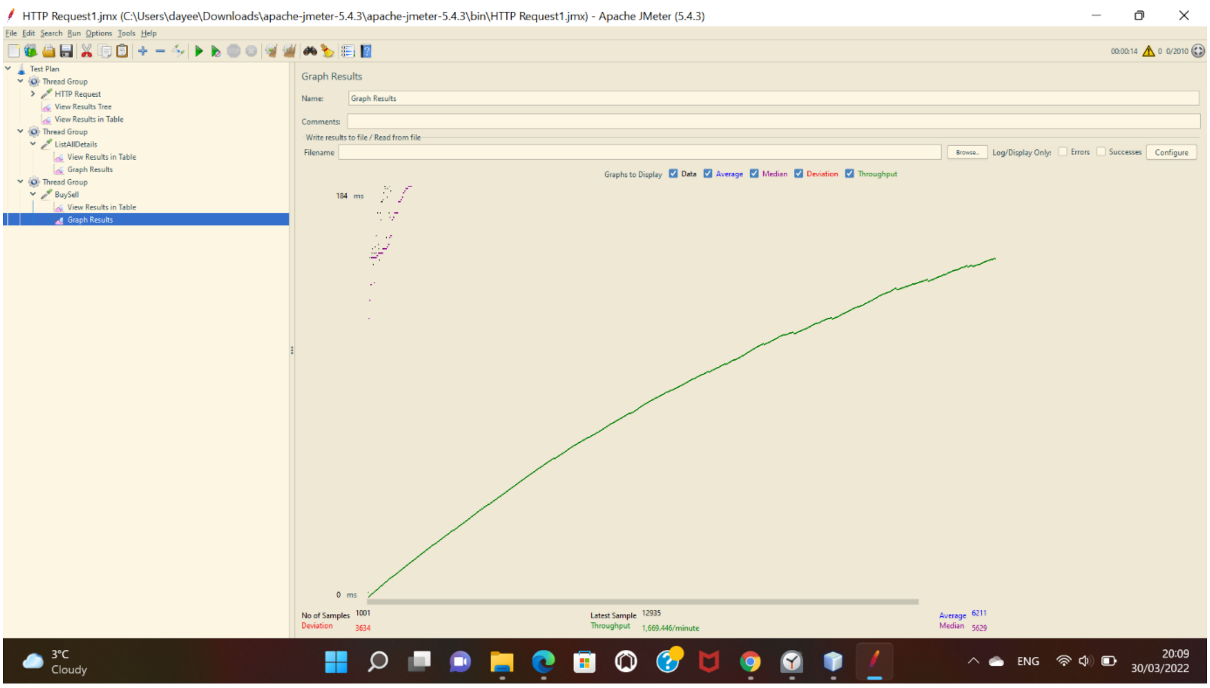


Figure 4: Graph showing BuySell API.

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

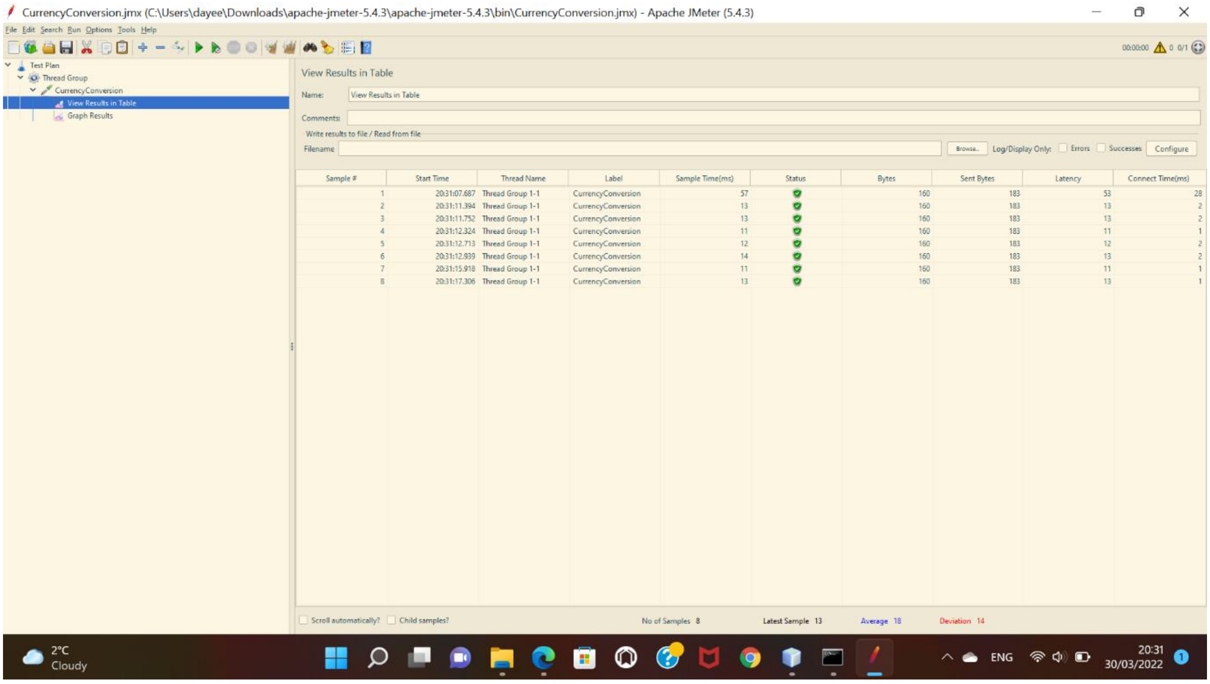


Figure 5: Table showing CurrencyConversion API.

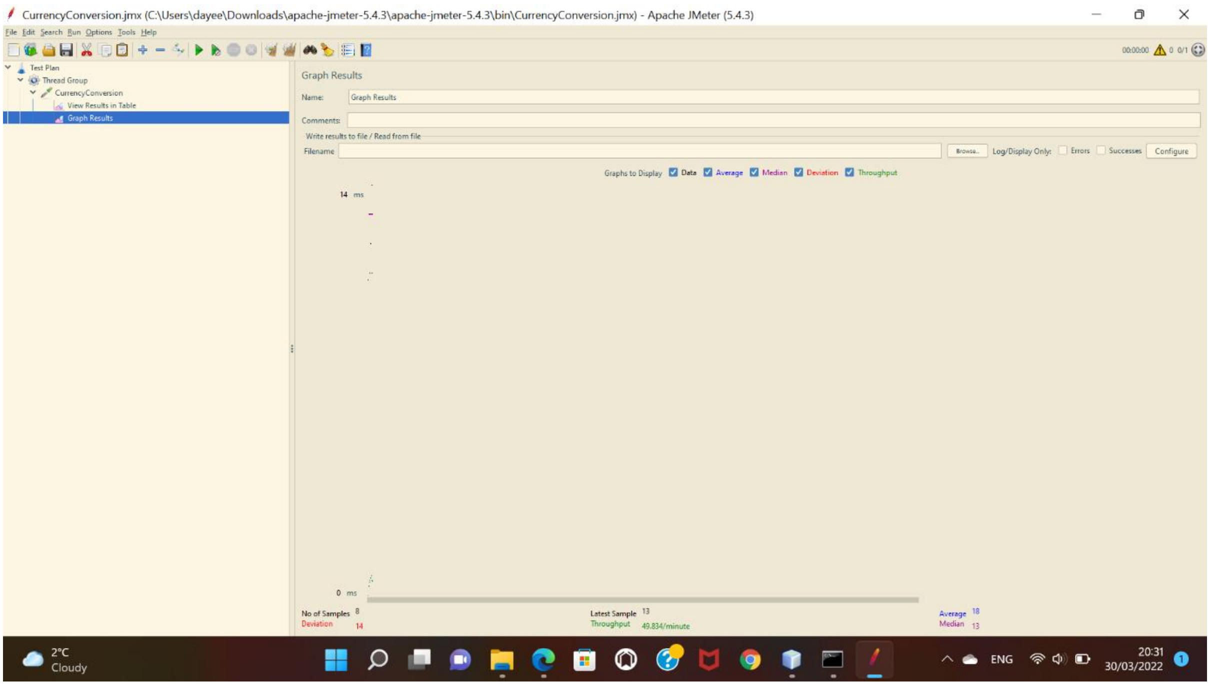


Figure 6: Graph showing CurrencyConversion API.

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

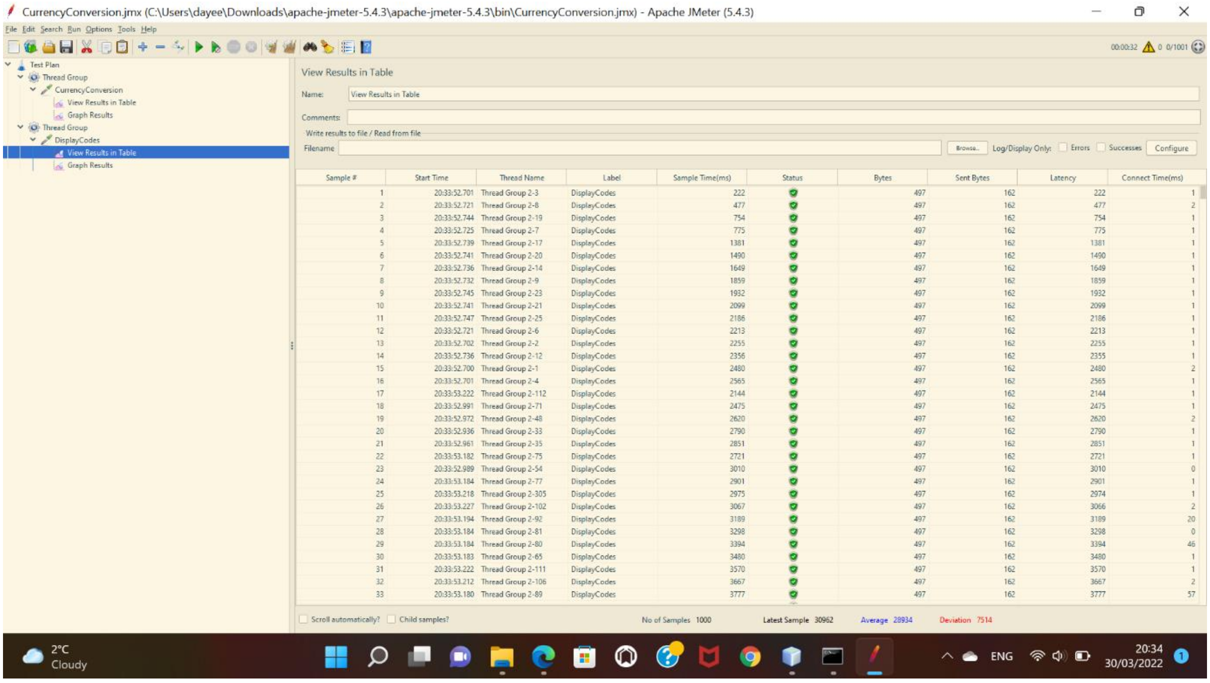


Figure 7: Table showing DisplayCodes API.

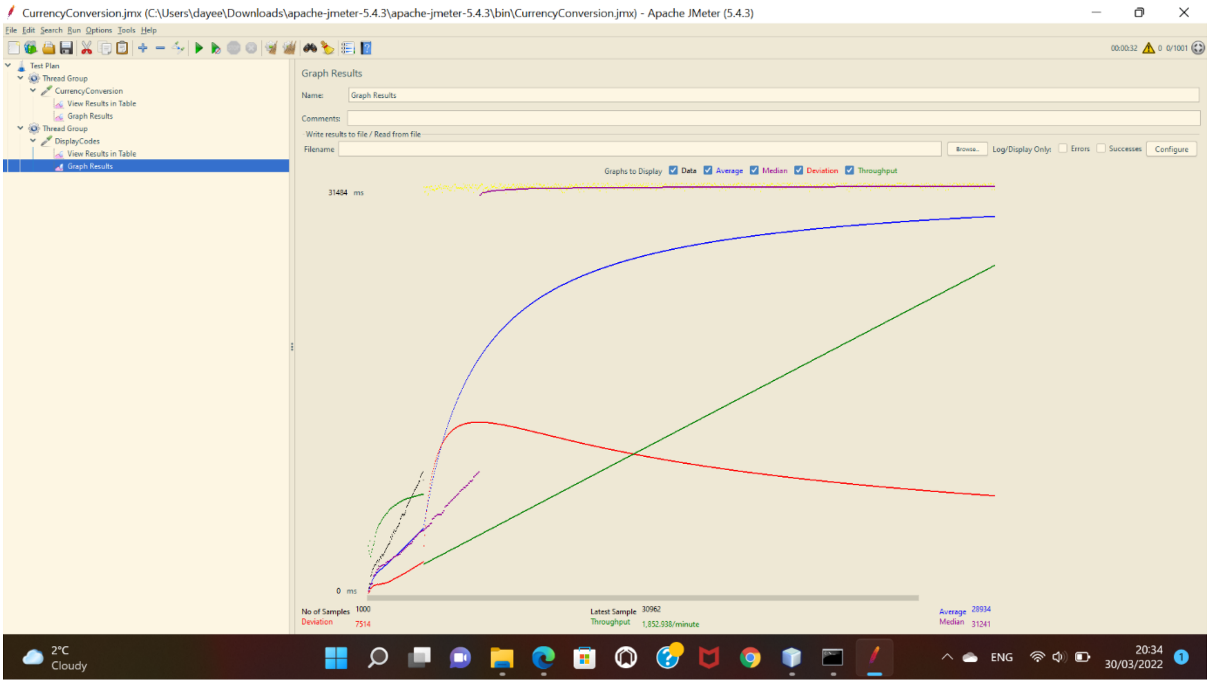


Figure 8: Graph showing DisplayCodes API.

MODULE CODE: ISYS40061
MODULE NAME: Service-Oriented Cloud Technologies
NAME: Dayeeta Das
NTU ID: N0830182

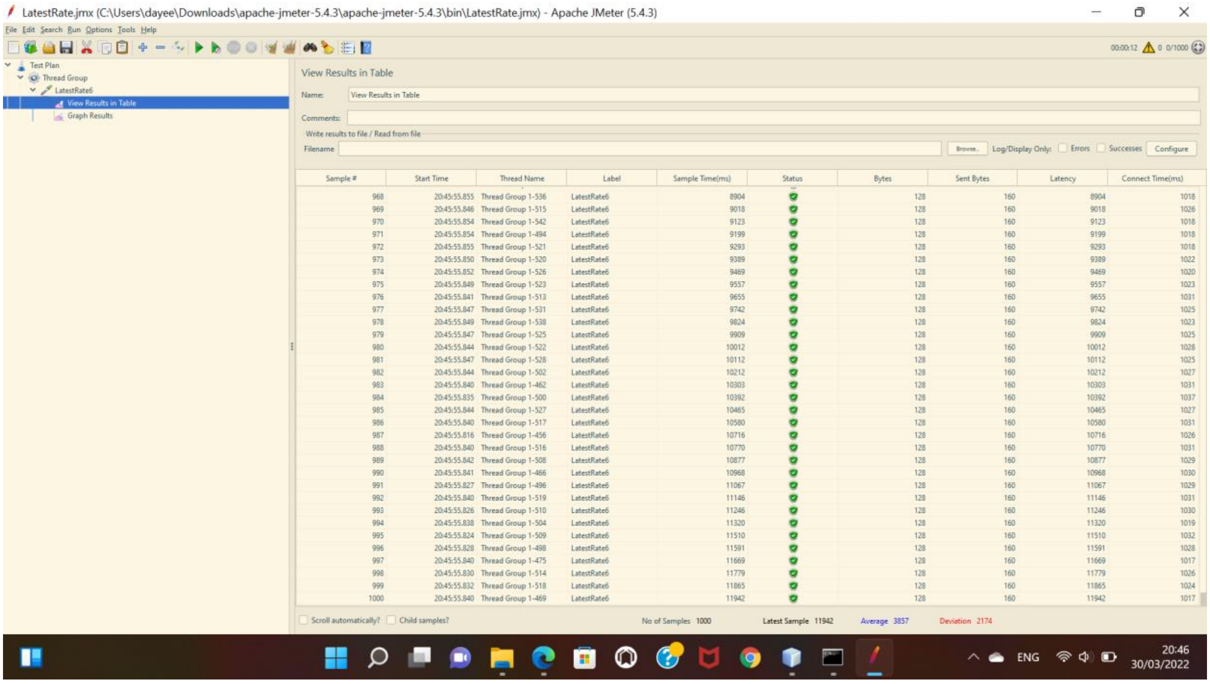


Figure 9: Table showing LatestRate API.

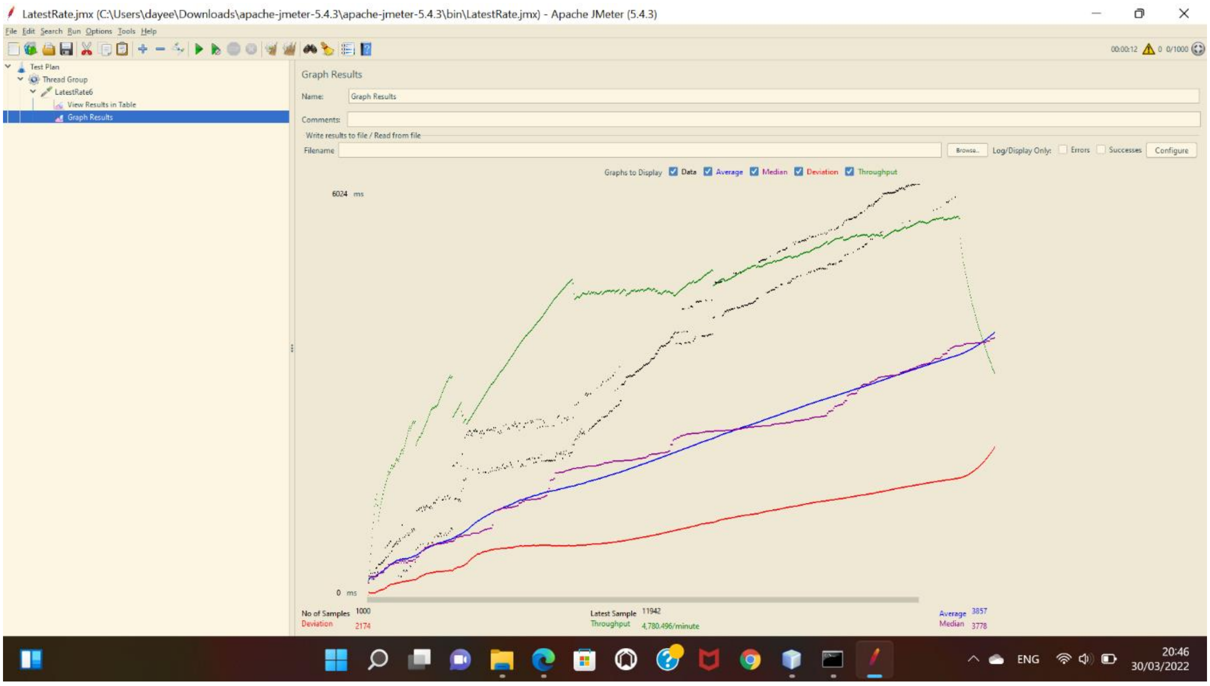


Figure 10: Graph showing LatestRate API.

Appendix B

References

1. IBM (2019). soa. [online] Ibm.com. Available at: https://www.ibm.com/cloud/learn/soa.
2. kexugit (n.d.). MSDN Magazine: SOA Tips - Address Scalability Bottlenecks with Distributed Caching. [online] docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/archive/msdn-magazine/2010/june/msdn-magazine-soa-tips-address-scalability-bottlenecks-with-distributed-caching#:~:text=As%20a%20result%2C%20the%20architecture [Accessed 30 Mar. 2022].
3. docs.microsoft.com. (n.d.). SOA Performance Considerations. [online] Available at: https://docs.microsoft.com/en-us/previous-versions/windows/desktop/cc907051(v=vs.85) [Accessed 30 Mar. 2022].
4. Narale, S. and Butey, P. (2015). CLOUD COMPUTING TECHNIQUES HELPS TO MEET QOS: A STUDY. International Journal of Advanced Computational Engineering and Networking, [online] (3), pp.2320–2106. Available at: http://www.iraj.in/journal/journal_file/journal_pdf/3-207-145018363687-90.pdf [Accessed 30 Mar. 2022].
5. DevOpsGroup. (n.d.). Containerisation | Cloud Migration Services. [online] Available at: https://www.devopsgroup.com/services/containerisation/#:~:text=Containers%20have%20become%20a%20key [Accessed 30 Mar. 2022].
6. Ontotext. (n.d.). What Are Linked Data and Linked Open Data? [online] Available at: https://www.ontotext.com/knowledgehub/fundamentals/linked-data-linked-open-data/#:~:text=Linked%20Data%20is%20a%20set [Accessed 30 Mar. 2022].