SuperArch: Optimal Architecture Design for Cloud Deployment

Kuldeep Singh
TCS, India
k.singh13@tcs.com

Chetan Phalak
TCS, India
chetan1.phalak@tcs.com

Dheeraj Chahal
TCS, India
d.chahal@tcs.com

Shruti Kunde
TCS, India
shruti.kunde@tcs.com

Rekha Singhal
TCS, India
rekha.singhal@tcs.com

ABSTRACT

The success of application migration to cloud depends on multiple factors such as achieving expected performance, optimal cost on deployment, data security etc. The application migration process starts with the architecture design, mapping technical and business specifications to the appropriate services in cloud. However, cloud vendors offer numerous services for each service type and requirement. The onus of selecting the optimal service from the pool lies with the user. Identifying an optimal service for a specific component or application requirement is a daunting task and necessitates a deep understanding of each cloud service offered.

This paper introduces SuperArch, a supervised architecture design tool designed to facilitate optimal selection and configuration of cloud services. We propose utilization of Large Language Models (LLM) for extracting information from user requirements and specifications, aiding in optimal selection of cloud services. Additionally, SuperArch maps workloads to the cloud services to generate optimal configurations of the cloud service and estimate performance and cost of the entire architecture.

CCS CONCEPTS

• **General and reference** → **Performance**; **Estimation**.

KEYWORDS

Performance and cost estimation, cloud deployment

ACM Reference Format:
Kuldeep Singh, Chetan Phalak, Dheeraj Chahal, Shruti Kunde, and Rekha Singhal. 2024. SuperArch: Optimal Architecture Design for Cloud Deployment. In *Companion of the 15th ACM/SPEC International Conference on Performance Engineering (ICPE '24 Companion), May 7–11, 2024, London, United Kingdom.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3629527.3651406

1 INTRODUCTION

Creating a resilient and efficient cloud-based system requires a delicate equilibrium among factors such as performance, cost, compliance, security, and user-specific requirements. Navigating the vast and ever-evolving design space encompassing various cloud services, new hardware, and multi-cloud deployments poses a significant challenge for the user. User-provided technical and business requirements may be fulfilled by multiple services from a cloud vendor. For example, an AWS database requirement can be served with Amazon RDS, DynamoDB, Amazon Neptune, or Amazon Aurora. In order to design an optimal architecture for complex user requirements, a conventional approach is based on manually exploiting abilities and experiences of cloud architects and iterative designs. Fortunately, the advent of state-of-the-art LLMs [8] presents an opportunity to find the optimal cloud-based services and their configuration that form the foundation of end-to-end architectures. Currently, the tools available in the market are not fully leveraging the capabilities of Large Language Models (LLMs) to automatically suggest high-performance and cost-effective optimal cloud architectures. This deficiency frequently results in the creation of inefficient designs, leading to lower performance and inaccurate cost estimates, ultimately causing cost escalations. Moreover, the hardware and software components provided by these cloud services can be configured in various ways influencing their performance and cost. A crucial factor in choosing the right cloud service and configuring it appropriately, is the anticipated workload, which is defined by factors such as expected number of users, size of data etc.

Tool	Cost		Guided design	Performance		All * vendors supported	LLM * support
	Selected services	End-to-end		Selected services	End-to-end		
Holori [6]	P	N	N	N	N	Y	N
Diagrams.net [4]	N	N	N	N	N	Y	P
Cloudskew [3]	N	N	N	N	N	Y	N
Cloudcraft [2]	P	P	N	N	N	N	N
Hava.io [5]	Y	Y	N	N	N	Y	N
Lucidchart [7]	Y	Y	N	N	N	Y	Y
Brainboard [1]	Y	Y	N	N	N	Y	N
SuperArch	Y	Y	Y	Y	Y	P	Y

Table 1: Comparison of SuperArch with cloud architecture design tools

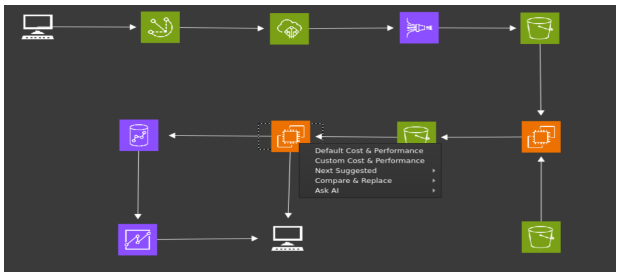
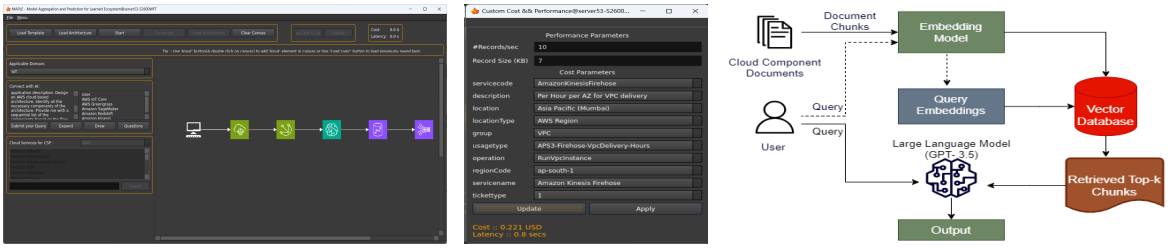


Figure 1: An example AWS cloud architecture design for IoT application using SuperArch

Most of the existing architecture design tool (see Table 1) do not map current as well as futuristic workloads to the cloud services in an architecture design often resulting in an inaccurate performance and cost estimation results in cost escalation and hence cloud repatriation.

ICPE '24 Companion, May 7–11, 2024, London, United Kingdom

ACM ISBN 979-8-4007-0445-1/24/05.
https://doi.org/10.1145/3629527.3651406



(a) (b) (c)

Figure 2: (a) Use of LLaMA for automated architecture design based on user specifications. (b) Cloud deployment cost estimation for AWS S3 service. (c) RAG design for finding architecture patterns from user specifications and vendor documents

This work aims to address these challenges and contribute to the development of a more nuanced approach to cloud-based system design using a tool called SuperArch (see Figure 1).

2 FEATURES OF SUPERARCH

Major contributions of our tool are as follows :

- **Leveraging Language Models:** The tool utilizes LLM to automatically identify and recommend cloud services based on capabilities, features, and system requirements. Additionally, it identifies optimal cloud services by recognizing architecture patterns in user's technical and business specifications. The tool automatically parses cloud services and their connections to draw architecture on the canvas as depicted in Figure 2(a).
- **Dynamic Architecture Modification**: The architecture generated by the LLM is editable which enables iterative modification of the architecture, allowing users to change and compare components at each step, for instance if user wants to compare RDS and DynamoDB it can be done within a few clicks.
- **Cost and Performance Estimation**: A repository of performance benchmarking data from various sources is maintained for popular cloud services such as storage, databases, VMs of various configurations, etc. Also, baseline cost numbers with associated parameters (geography, cost/hour, configuration etc.) are retrieved using cost calculators provided by vendors. This data is used in conjunction with workload data available in the user specification document to calculate end-to-end performance and cost of the architecture as illustrated in Figure 2(b).
- **User-Friendly Interface and multi-cloud support**: Offers an intuitive and user-friendly interface for seamless interaction, and also simplifies the architecture design process by recommending *next optimal cloud service and configuration* at each step in the design process.
- **Iterative Refinement**: Allows iterative refinement of the architecture based on user feedback, ensuring collaborative efforts between the tool's capabilities and user domain expertise. Tool provides features to save and load versions of the architecture.
- **Domain Based Templates:** The tool offers domain-specific templates, tailoring designs to applications of various industries or fields. Users can easily access and utilize pre-designed templates relevant to their application domain, streamlining the designing process and ensuring a professional and industry-appropriate architecture for their application.

3 USE CASE

Figure 1 shows an AWS architecture for an IoT application using SuperArch. User specifications for the application requires streaming on-premise data to cloud for storage, model training, performing analytics and sending results back to the user. The architecture is generated using LLM and supervised design functionalities providing assistance to architects in making optimal choices for cloud components from a vast array of available options.

4 CONCLUSION AND FUTURE WORK

We proposed the SuperArch tool that utilizes the capabilities of Large Language Models (LLMs) for optimal cloud architecture design. Furthermore, we presented a methodology for estimating the performance and cost of architectures by mapping workloads to the cloud. The current implementation of C-Arch employs LLaMA. We are currently in the process of incorporating a Retrieval Augmented Generation (RAG) pipeline using ChatGPT-4 2(c), aiming to enhance the capabilities of Large Language Models (LLMs) with relevant cloud service information sourced from the internet.

REFERENCES

[1] Brainboard. [n. d.]. The cloud is your canvas. Accessed Jan. 22, 2024. https://www.brainboard.co/
[2] Cloudcraft. [n. d.]. Visualize your cloud architecture like a Pro. Accessed Jan. 22, 2024. https://www.cloudcraft.co/
[3] Cloudskew. [n. d.]. Online Diagram, Flowchart Maker. Accessed Jan. 22, 2024. https://www.cloudskew.com/
[4] Diagrams.net(Draw.io). [n. d.]. Flowchart Maker Online Diagram Software. Accessed Jan. 22, 2024. https://app.diagrams.net/
[5] Hava.io. [n. d.]. Automated Cloud Diagrams in Minutes. Accessed Jan. 22, 2024. https://www.hava.io/
[6] Holori. [n. d.]. Cloud cost platform with infrastructure visibility. Accessed Jan. 22, 2024. https://holori.com/
[7] Lucidchart. [n. d.]. Diagram your people, processes, and systems. Accessed Jan. 22, 2024. https://www.lucidchart.com/pages/
[8] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]