Insert Your Paper Title Here∗

Insert Any Paper Subtitle Here

FirstName Surname†
 Department Name
 Institution/University Name
 City State Country
 email@email.com

FirstName Surname
 Department Name
 Institution/University Name
 City State Country
 email@email.com

FirstName Surname
 Department Name
 Institution/University Name
 City State Country
 email@email.com

ABSTRACT

Please provide an abstract for your paper. The abstract should introduce the serverless application your team has implemented. Describe the key features provided by the application as well as the design tradeoffs you studied. (e.g. service composition, application flow control, alternate back-end databases, etc.) The abstract can include one or two key performance, throughput, or cost results your group measured after implementing the application.

CCS CONCEPTS

• To generate the ACM Computing Classification for the paper, see instructions here: <https://www.acm.org/publications/class-2012> • Insert CCS text here • Insert CCS text here

KEYWORDS

Provide a list of keywords for the paper separated by commas. Word1, Word2, Word3

ACM Reference format:

FirstName Surname, FirstName Surname and FirstName Surname. 2018. Insert Your Title Here: Insert Subtitle Here. In *Proceedings of ACM Woodstock conference (WOODSTOCK’18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1 Introduction

For the report title, please adjust the title above to describe your project. In the introduction, please introduce the serverless application you’ve implemented. If your group did not implement the standard “TLQ” application, please include background to fully describe the application from the perspective of someone who is unfamiliar with it. Be sure to define acronyms and include references where appropriate. If your group implemented the “TLQ” application, first describe your group’s interpretation of the TLQ application, and describe anything that may be different or unique regarding your group’s implementation. (e.g. languages used, tools, alternate cloud services, databases, etc.)

Example of display a Formula with a Formula Number:

  (1)

Example of a Formula without Number:



Here is an example of a figure with a **figure caption**. Tables should also use the same caption style for **table captions**:



Figure 1: Figure Caption and Image above the caption [In draft mode, Image will not appear on the screen]

1.1 Research Questions

For your paper, write two or more “research” questions which you will answer through the performance experiments with respect to the design tradeoffs studied by your group. These research questions ask about how the tradeoffs of alternate implementations impact key indicators such as: performance, throughput, and/or cloud hosting costs ($). (e.g. switchboard vs. full service isolation, etc.) Your research questions may also consider the performance and cost implications of the memory reservation size on FaaS platforms. Since CPU power is coupled to memory size on AWS Lambda, the memory setting will impact performance, and also cost due to the FaaS pricing policy. Your research questions may also want to consider the freeze-thaw lifecycle of serverless computing. The switchboard architecture pattern, for example, is designed specifically to minimize infrastructure initialization overhead.

We investigate the following research questions: (Replace with your specific research questions)

**RQ-1:** How does service composition of application microservices impact the application performance when hosted using Function-as-a-Service platforms?

**RQ-2:** How does Function-as-a-Service memory reservation size impact application performance, throughput, and hosting costs for the “TLQ” pipeline application?

Please write or adapt the sample research questions and tailor them to your project. The paper should then present data that tries to answer the questions. Through the written narrative refer to the research questions by number (RQ-1), (RQ-2), etc. This helps tie together the presentation of your research questions and research approach with your experimentation and analysis. Do not state research questions that are entirely unexplored by your group project. Given time constraints in TCSS 562, it is entirely possible that research questions may not be entirely answered by the study. However, there should be at least some data, and an attempt to explore the question(s). Remember the experimental results simply provide data. They do not provide “right” or “wrong” answers or results. It is the role of the researcher to correctly design and execute experiments and report on the results regardless if the data is what we expected or not!

2 Case Study

Section II of your paper should describe the serverless application implemented by your group. The first paragraph can provide additional detail not already in the introduction regarding the application.

One paragraph could describe why it is interesting to implement your application as a FaaS application case study. What is in particular challenging or interesting, (e.g. machine learning, data processing pipeline) where having access to the features of FaaS platforms (e.g. access to many CPUs in parallel for on-demand scaling, high availability, 24/7, pay only for resources actually used, no idle servers) is paramount?

2.1 Design Tradeoffs

Sub-section 2.1 should describe the design tradeoffs studied by your group. (e.g. alternate service compositions, application flow control, alternate back-end relational databases, etc.) Include diagrams, tables, or textual descriptions to depict your alternate implementations. Consider labeling the alternate implications for easy reference throughout the text for abbreviation in other tables, graphs, etc. A brief description on the expectations of your alternate designs could be included. For example, what do you expect to see when testing alternate flow control implementations? In the end, it will be interested to see how expectations stand up to real world tests.

2.2 Serverless Application Implementation

Sub-section 2.2 should describe the details for your application implementation. Describe any special languages, development libraries, tools, databases, external cloud services, and/or other technologies used in the implementation of your application. The description of your application implementation should help the reader understand how your application has been designed. When you report on the performance, throughput, and hosting costs of your tests, a thorough description of your application can help the reader understand what system you actually tested. This helps the reader understand if your testing results may be relevant to applications in their problem domain. Diagrams and figures can be included if helpful.

Please describe how state information is tracked and maintained in your implementation. For the “TLQ” application, how was data passed among the services? Are there separate SQL-Lite databases for each user session? If so, this would allow your TLQ pipeline to process many independent “flows” of data in parallel. If there is just one-database, do you differentiate between users? Has the application been designed to be multi-user?

If a user calls the “load” service to populate a database, is the application flow control among services asynchronous, synchronous, or does your application support both schemes? How does your application support asynchronous flow control? How are messages persisted by the server, and retrieved by the client?

2.3 Experimental Approach

Sub-section 2.3 should describe the experimental approach to evaluate design tradeoffs of the case study. How were the experiments designed to evaluate the case study topic(s), for example: service composition, application flow-control, memory reservation size, etc. Describe the client-side infrastructure used for testing. Described the server-side infrastructure configurations for testing. For example, How much memory was allocated to Lambda functions? Was the timeout adjusted? Were functions assigned to run in a VPC? A single availability zone? Which cloud region(s) were used? How did the client(s) communicate with your serverless application?

Section 2.3 should describe the test configurations, and tests that were performed so in the future someone could repeat/reproduce your experimental results. Repeating experiments can help to verify if the results and conclusions were correct. Repeating tests can also help evaluate how public cloud service quality and implementations are changing over time. Are cloud providers addressing issues in class that we’ve talked about regarding performance variability? Does performance of the cloud platform improve over time?

Groups may wish to write custom BASH/ Python scripts to implement all tests to make it easier to repeat/reproduce test results later on. If test scripts were developed, these could be included in a git repository and the project report could refer to them. SAAF and FaaS Runner provide tools to help make it easier to create the scripts.

3 Experimental Results

Section III should present the results of your study. This should include data tables and graphs with captions where appropriate. Here are some simple examples of captions:

4 Conclusions

Section 4 should summarize the results of your project and case study. It should reiterate what the research questions were, and restate the key conclusions from section 3 based on the experimental results. Ideally, there are one or two key results or highlights that can be summarized for each research question. Please identify the research question by number such as (RQ-1) when summarizing key conclusions from the experimental results. Someone reviewing your paper will look to see if in your conclusion section, you’ve answered the research questions proposed in the introduction. Good research papers do not hide key results from the reader. Good papers also do not present tables with raw data and make the reader perform the analysis. The authors of papers should use visualizations such as tables and figures to show the results, and then craft a narrative which describes the key observations from the experiments. The authors should not present data to the reader and only “discuss” passively what it might mean. Good research papers make assertive conclusions based on facts and data. The core components of the research paper should be accessible in the introduction (Section 1) and the conclusion (section 4). A reader wanting more detailed information can then read the intermediate sections if they find your paper of interest.

ACKNOWLEDGMENTS

Acknowledgements may be included but are not required. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

REFERENCES

This section describes the format for the references. Papers should have at least 5!!! The template will number citations consecutively within brackets [1]. When using a reference in a sentence, the punctuation mark will follow the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

If using MS Word, the Mendeley Reference Manager provides a free tool which can automatically generate a bibliography and manage references throughout the paper. This tool can be downloaded here:

<https://www.mendeley.com/download-reference-manager/>

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

[1] Patricia S. Abril and Robert Plant, 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan, 2007), 36-44. DOI: <https://doi.org/>10.1145/1188913.1188915.

[2] Sten Andler. 1979. Predicate path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226-236. DOI:https://doi.org/10.1145/567752.567774

 [3] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:https://doi.org/10.1007/3-540-09237-4.

[4] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY..

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00