Practical Cloud Workloads for Serverless FaaS

Jeongchul Kim
Dept. of Computer Science. Kookmin Univ. South Korea.
kjc5443@kookmin.ac.kr

Kyungyong Lee
Dept. of Computer Science. Kookmin Univ. South Korea.
leeky@kookmin.ac.kr

ABSTRACT
Serverless computing is gaining popularity with the Function-as-a-Service (FaaS) execution model. Without incurring overheads involved in provisioning cloud instances and with high availability and scalability, serverless computing allows developers to focus on implementation of core application logic using other well-developed cloud services. By abstracting the complex resource management task, serverless computing opens new opportunities for the cloud service adoption even to non-cloud experts [2]. With the popularity, many research results have been published using the FaaS execution model. They include investigation of serverless computing opportunities [1], proposing new serverless applications, function run-time optimization, and public service comparison. Without a common test benchmark suite, authors in the previous work had evaluated proposed systems using fairly simple FaaS applications, such as micro-benchmarks that emphasize specific resources exclusively, e.g., CPU, disk I/O, and network. However, such simple workloads do not represent realistic FaaS system applications, and the evaluations might not compare proposed systems appropriately.

To overcome the limitation of lacking a comprehensive benchmark suite for the serverless computing and FaaS execution model, the authors create FunctionBench that provides various FaaS workloads that are ready to be executed on public cloud function execution services - AWS Lambda, Google Cloud Functions, and Azure functions. Since the inception of serving the FaaS workloads, we keep working to expand the supported applications and add scenarios in big-data processing, back-end web applications, and security. To represent big-data applications, we add a MapReduce WordCount workload, which counts the number of occurrences of each word in a given partitioned input dataset from Wikipedia. To cover web back-end applications, we add Chameleon. The application renders a template using the Chameleon module in Python PIP library to create an HTML table of N rows and M columns that are provided as input arguments. Another web-related application is JSON serialize-deerialize module. The application performs JSON deserialization using a JSON-encoded string dataset (Awesome JSON Dataset) downloaded from a public object storage service, and it serializes the JSON object again. To represent security-related applications, we add Pyaes benchmark that performs private key-based encryption and decryption. It is a pure-Python implementation of the AES block-cipher algorithm in CTR mode. We also add gzip-compression benchmark to represent realistic disk IO-heavy applications. The degree (High, Medium, Low) of resource usage characteristics of newly proposed applications are summarized in Table 1. Please refer to [3] to read the description of comprehensive applications list.

The proposed FunctionBench provides a variety of FaaS applications in multiple categories, and we are sure that it will enable fair evaluation of new research work in the relevant field with practical application scenarios.

CCS CONCEPTS
• Networks → Cloud computing; • Computer systems organization → Cloud computing.

KEYWORDS
serverless, faas, workload, benchmark, cloud computing

Table 1: Newly added workloads in FunctionBench

<table>
<thead>
<tr>
<th>Application</th>
<th>Name</th>
<th>Amount of Load</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CPU</td>
</tr>
<tr>
<td>Bigdata</td>
<td>MapReduce</td>
<td>H</td>
</tr>
<tr>
<td>Web</td>
<td>Chameleon</td>
<td>M</td>
</tr>
<tr>
<td>Security</td>
<td>pyaes</td>
<td>H</td>
</tr>
<tr>
<td>Media</td>
<td>Image</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>Video</td>
<td>H</td>
</tr>
</tbody>
</table>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM Reference Format:

ACKNOWLEDGMENTS
This work is supported by the National Research Foundation of Korea Grant funded by MSIP (NRF-2015R1A5A7037615), the IITP (2017-0-00396), and the AWS Cloud Credits for Research program.

REFERENCES

Table 1: Newly added workloads in FunctionBench

1 https://github.com/kmu-bigdata/serverless-faas-workbench

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM Reference Format:

ACKNOWLEDGMENTS
This work is supported by the National Research Foundation of Korea Grant funded by MSIP (NRF-2015R1A5A7037615), the IITP (2017-0-00396), and the AWS Cloud Credits for Research program.

REFERENCES

Table 1: Newly added workloads in FunctionBench

1 https://github.com/kmu-bigdata/serverless-faas-workbench

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM Reference Format:

ACKNOWLEDGMENTS
This work is supported by the National Research Foundation of Korea Grant funded by MSIP (NRF-2015R1A5A7037615), the IITP (2017-0-00396), and the AWS Cloud Credits for Research program.

REFERENCES