of existing monolithic code may not bring too much performance degradation.

Exploiting Parallelism. During the decomposition of MetaCompare, we observed that its implementation involves three external library function invocations, and that these invocations are independent of each other. In the previous experiment, our decomposed version had these external invocations grouped into a single instance. In this experiment, we also parallelize these external library function invocations using FaaS. We measure the execution time under this approach and compare it with our previous version. Figure 7 shows the result and breakdown of the time taken by each of the library invocations. In our first decomposed version, the library invocations are sequential, whereas in our parallel version the three can run simultaneously. As a result, the overall execution time for the parallel version is reduced—though still dictated by the longest invocation, i.e., #3—by 17%.

We note that even after the parallelization of the library invocations, the overall execution time is well over 30 minutes, which is much more than the average life cycle of a typical serverless instance [18]. This can be remedied by scaling out the system, i.e, instead of just one FaaS instance processing all of the input data, create a large number of instances each processing a fraction of data. However, this may significantly increase function-to-function communication. We plan to explore such optimizations and trade-offs therein in our future work.

6 CONCLUSION

Serverless computing offers great promise for legacy and emerging scientific computing applications. In this paper, we presented a systematic approach to decompose a scientific workflow into a set of functions that can be deployed on serverless platforms such as AWD Lambda. We demonstrate our approach in the context of a bioinformatics pipeline (CIWARS). Our decomposition goal is to preserve correctness of the original application without compromising performance. We also presented the design of an automation tool that can provide finer-grained FaaS functions. The tool employs a two-step approach of control-flow decomposition and data-flow analysis to transform a monolithic application into functions for the target CIWARS. The tool also provides the foundation for building general-purpose automatic tools for FaaS implementations of general monolithic applications. Our evaluation shows that our approach is able to preserve correctness, and incur only small (2.0%) overhead compared to the original application. In the future, we aim to fully automate our process, and test our solutions more rigorously and at scale.

ACKNOWLEDGEMENTS

This work is sponsored in part by the National Science Foundation under grants CCF-1919113 and OAC-2004751.

REFERENCES

[1] 2018. Actiavted sludge samples collected from Shatin WWTP, Hong Kong. https://www.ncbi.nlm.nih.gov/bioproject/PRJNA432264. Accessed: April 9, 2021.
[2] 2021. Docker Engine Release Notes. https://docs.docker.com/engine/release-notes/#20103. Accessed: April 9, 2021.
[3] 2021. Kubernetes. https://kubernetes.io/. Accessed: April 9, 2021.
[4] 2021. python - Docker Hub. https://hub.docker.com/_/python?tab=description. Accessed: April 9, 2021.
[5] Istemi Ekin Akkus, Ruichuan Chen, Ivica Rimac, Manuel Stein, Klaus Satzke, Andre Beck, Paarijaat Aditya, and Volker Hilt. 2018. SAND: Towards High-Performance Serverless Computing. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. Boston, MA, 923–935.
[6] Lixiang Ao, Liz Izhikevich, Geoffrey M. Voelker, and George Porter. 2018. Sprocket: A Serverless Video Processing Framework. In *Proceedings of the ACM Symposium on Cloud Computing, SoCC*.
[7] Gustavo Arango, Gargi Singh, Lenwood Heath, Amy Pruden, Weidong Xiao, and Liqing Zhang. 2016. MetaStorm: A Public Resource for Customizable Metagenomics Annotation. *PloS one* 11 (09 2016), e0162442.
[8] G. A. Arango-Argoty, E. Garner, A. Pruden, L. S. Heath, P. Vikesland, and L. Zhang. 2017. DeepARG: A deep learning approach for predicting antibiotic resistance genes from metagenomic data. *bioRxiv* (2017). https://doi.org/10.1101/149328
[9] Paul C. Castro, Vatche Ishakian, Vinod Muthusamy, and Aleksander Slominski. 2019. The server is dead, long live the server: Rise of Serverless Computing, Overview of Current State and Future Trends in Research and Industry. *CoRR* abs/1906.02888 (2019). arXiv:1906.02888
[10] Ryan Chard, Tyler J. Skluzacek, Zhuozhao Li, Yadu Babuji, Anna Woodard, Ben Blaiszik, Steven Tuecke, Ian Foster, and Kyle Chard. 2019. Serverless Supercomputing: High Performance Function as a Service for Science. arXiv:1908.04907 [cs.DC]
[11] A. Christidis, R. Davies, and S. Moschoyiannis. 2019. Serving Machine Learning Workloads in Resource Constrained Environments: a Serverless Deployment Example. In *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)*. 55–63.
[12] Peter Cock, Christopher Fields, Naohisa Goto, Michael Heuer, and Peter Rice. 2009. The Sanger FASTQ format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic acids research* 38 (12 2009), 1767–71. https://doi.org/10.1093/nar/gkp1137
[13] S. Eismann, J. Scheuner, E. van Eyk, M. Schwinger, J. Grohmann, N. Herbst, C. L. Abad, and A. Iosup. 2021. Serverless Applications: Why, When, and How? *IEEE Software* 38, 1 (2021), 32–39.
[14] Doug Hyatt, Gwo-Liang Chen, Phil Locascio, Miriam Land, Frank Larimer, and Loren Hauser. 2010. Hyatt D, Chen G-L, LoCascio PF, Land ML, Larimer FW, Hauser LJ.. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinform 11: 119. *BMC bioinformatics* 11 (03 2010), 119.
[15] Amazon Web Services Inc. 2021. Amazon Lambda. https://aws.amazon.com/lambda. Accessed: April 9, 2021.
[16] Douglas M Jacobsen and Richard Shane Canon. 2015. Contain this, unleashing docker for hpc. *Proceedings of the Cray User Group* (2015), 33–49.
[17] Cinar Kilcioglu, Justin M. Rao, Aadharsh Kannan, and R. Preston McAfee. 2017. Usage Patterns and the Economics of the Public Cloud. In *Proceedings of the 26th International Conference on World Wide Web, WWW*.
[18] Ana Klimovic, Yawen Wang, Patrick Stuedi, Animesh Trivedi, Jonas Pfefferle, and Christos Kozyrakis. 2018. Pocket: Elastic Ephemeral Storage for Serverless Analytics. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*.
[19] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. 2017. Singularity: Scientific containers for mobility of compute. *PLOS ONE* 12, 5 (05 2017), 1–20. https://doi.org/10.1371/journal.pone.0177459
[20] Dirk Merkel. 2014. Docker: lightweight Linux containers for consistent development and deployment. *Linux Journal* 2014 (03 2014).
[21] Jan Midtgaard. 2012. Control-Flow Analysis of Functional Programs. *ACM Comput. Surv.* 44, 3, Article 10 (June 2012), 33 pages.
[22] Min Oh, Amy Pruden, Chaoqi Chen, Lenwood S Heath, Kang Xia, and Liqing Zhang. 2018. MetaCompare: a computational pipeline for prioritizing environmental resistome risk. *FEMS Microbiology Ecology* 94, 7 (04 2018).
[23] William Pearson and D Lipman. 1988. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proc Natl Acad Sci USA 85: 2444-2448. *Proceedings of the National Academy of Sciences USA* 85 (05 1988), 2444–8.
[24] Simon Shillaker and Peter Pietzuch. 2020. Faasm: Lightweight Isolation for Efficient Stateful Serverless Computing. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*.
[25] Josef Spillner. 2017. Transformation of Python Applications into Function-as-a-Service Deployments. *CoRR* abs/1705.08169 (2017).
[26] Josef Spillner, Cristian Mateos, and David A. Monge. 2018. FaaSter, Better, Cheaper: The Prospect of Serverless Scientific Computing and HPC. In *High Performance Computing*, Esteban Mocskos and Sergio Nesmachnow (Eds.). Springer International Publishing, Cham, 154–168.
[27] H. Wang, D. Niu, and B. Li. 2019. Distributed Machine Learning with a Serverless Architecture. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*.
[28] Liang Wang, Mengyuan Li, Yinqian Zhang, Thomas Ristenpart, and Michael Swift. 2018. Peeking Behind the Curtains of Serverless Platforms. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*.
[29] Tianyi Yu, Qingyuan Liu, Dong Du, Yubin Xia, Binyu Zang, Ziqian Lu, Pingchao Yang, Chenggang Qin, and Haibo Chen. 2020. Characterizing Serverless Platforms with Serverlessbench. In *Proceedings of the 11th ACM Symposium on Cloud Computing* (Virtual Event, USA) *(SoCC '20)*. 30–44.