The Journal of Open Source Education

StarBLAST: a scalable BLAST+ solution for the classroom

Michele Cosi[1], **J.J. Forstedt**[1], **Emmanuel Gonzalez**[1], **Zhuoyun Xu**[1], **Sateesh Peri**[1], **Reetu Tuteja**[1], **Kai Blumberg**[1], **Tanner Campbell**[1], **Nirav Merchant**[1], **and Eric Lyons**[1]

1 Data Sciences Institute, University of Arizona, Tucson, AZ 85721, USA

DOI: 10.21105/jose.00102

Software

- Review
- Repository
- Archive

Submitted: 02 October 2020
Published: 27 April 2021

Summary

Basic Local Alignment Search Tool (BLAST) (Madden, 2002) is a heuristic nucleotide and protein search tool frequently used in biological research to match sequences to those in a database. It is becoming increasingly common in classrooms, including the high school-grade level Shaffer et al. (2010) through the publicly accessible version, NCBI BLAST (Johnson et al., 2008). However, NCBI BLAST can become problematic in a classroom environment, as many job submissions from the same IP address may be throttled (IP block), response time may be slow, and users are unable to import custom databases. In addition, updates to NCBI databases may produce results incongruent with lesson plans. SequenceServer (Priyam et al., 2019) addresses some of these needs by providing a student-friendly, web-accessible interface of NCBI-BLAST+ (Camacho et al., 2009) and an expanded collection of BLAST tools, with options to use custom databases. However, SequenceServer does not address the scalability issues for large class sizes, and deploying the software is still difficult for instructors with minimal computational resources and technical expertise. To address these needs, we developed StarBLAST, a modular deployment strategy for SequenceServer that is suitable for a variety of technical skill-levels and classroom sizes. This is achieved by leveraging a master-worker framework for distributed and scalable computing using WorkQueue and Makeflow (Albrecht et al., 2012). WorkQueue manages jobs (compute tasks) to be distributed from a "Master" node to designated "Worker" nodes for completion. Its inherent scalability makes it functional for large and small classrooms. Through CyVerse VICE (Devisetty et al., 2016), an instructor can quickly deploy SequenceServer and databases on a web-based interface. Here, we describe the three deployable methods of StarBLAST that account for custom databases, classroom size, technical expertise, and computational resources.

Statement of Need

NCBI BLAST has become an increasingly popular tool, well-suited for researchers, but less accessible to a classroom environment. Multiple incoming connections to the NCBI BLAST services from the same classroom IP address can lead to IP block and slow BLAST results. Moreover, as NCBI databases are updated, results may be inconsistent with original lesson plans. StarBLAST addresses these issues by allowing its users to access a scalable, user-friendly interface, adaptable to classrooms of any size and technical expertise, enabling the use of custom databases and fast BLAST results.

Description

StarBLAST deployment methods include StarBLAST-VICE, StarBLAST-Docker, and StarBLAST-HPC. StarBLAST-VICE is hosted in the CyVerse Discovery Environment (DE) (Merchant et al., 2016), a web-based data science workbench for researchers. StarBLAST-Docker and StarBLAST-HPC integrate SequenceServer with WorkQueue and Makeflow for distributing BLAST jobs across multiple computer nodes on the Cloud or high-performance computing (HPC) environments.

StarBLAST-VICE

StarBLAST-VICE is a dockerized image of SequenceServer integrated as a VICE application in the CyVerse DE. This solution quickly launches SequenceServer on a virtual machine with up to 8 CPU cores and 16GB RAM, and is practical for small classrooms (<25 students). Launching StarBLAST-VICE requires minimal technical expertise and no supporting infrastructure except for an internet connection, and custom BLAST databases may be specified. A supporting app, *Create_BLAST_database*, is integrated in CyVerse to allow instructors to quickly create custom BLAST databases. The deployment speed of the StarBLAST-VICE app is dependent on the size of the input BLAST database, which is copied to the VM, and usually takes only a few minutes.

StarBLAST-Docker

StarBLAST-Docker uses WorkQueue and Makelfow for a master-worker framework to distribute BLAST jobs among a master Virtual Machine (VM) and one (or more) Worker VMs, allowing for scalability. Although designed for use on NSF's XSEDE Jetstream (Stewart et al., 2015), StarBLAST-Docker is compatible with other cloud computational resources such as Digital Ocean and Google Cloud. StarBLAST-Docker requires running two or more VMs and can scale to handle a moderate number of students (<100, depending on the number of worker VMs). The master VM is responsible for managing SequenceServer and sending BLAST jobs to worker VMs. Once these VMs are started and configured, students can easily connect to SequenceServer on the master VM using a web-browser. Setting up StarBLAST-Docker requires minimal technical knowledge. Prior to launching the VMs, two deployment scripts need to run, one each for the master and worker VMs. The deployment scripts are available in GitHub and a step-by-step tutorial is available in the StarBLAST documentation. BLAST databases are automatically downloaded to each worker VM from the CyVerse DataStore, and custom databases can be indexed by either using the companion app *Create_BLAST_database* or NCBI's *makeblastdb* command (available with SequenceSever) and stored in the same location as other databases on the VMs. StarBLAST-Docker is scalable depending on the size and number of worker VMs provided by the user. An additional option for StarBLAST-Docker is its ability for deployment across a local area network using the instructor's computer as the master and the students' computers as workers. This method keeps everything local and is ideal for classroom environments with limited or unreliable internet connectivity.

StarBLAST-HPC

StarBLAST-Docker workers can be deployed on HPC systems instead of VMs, enabling further scalability. Setting up StarBLAST-HPC requires some setup and moderate knowledge of the linux command-line and HPC (StarBLAST's documentation has examples for PBS-Pro.) The master node for StarBLAST-HPC, equivalent to the master VM for StarBLAST-Docker, is deployed to manage SequenceServer and send BLAST jobs to HPC

worker nodes. Workers run on nodes of the HPC using WorkQueue's *work_queue_factory*, allowing instructors to specify the number of workers run on each node, based on the amount of resources available per node. Given that many HPC systems have nodes with 16-96 cores, the number of workers can be tailored to optimize for the number of students and current workload on the HPC. Since HPC resources are shared and may not be available instantly, HPC submission requests must be done well ahead of class to ensure the worker nodes are available and accessible. Due to the computational power of the HPC, more than 100 students can be supported by StarBLAST-HPC. The script and HPC set-up tutorials are available in StarBLAST's documentation. BLAST databases are obtainable from the CyVerse DataStore but are added manually to the master VM; the instructor/IT staff are required to copy the BLAST databases to the HPC.

Availability and Requirements

StarBLAST-VICE is available on CyVerse's Discovery Environment (https://de.cyverse.org/de/). StarBLAST-Docker Master and Worker VM images are available on NSF's XSEDE Jetstream (https://use.jetstream-cloud.org/). Deployment scripts can be obtained from https://github.com/LyonsLab/StarBlast/. To set up StarBLAST-Docker on a different cloud computational resource, Docker version 19.03.5 and CCTools version 7.0.21 are required. HPC integration requires access to a High Performance Computing system. Documentation to the StarBLAST suite is available at https://starblast.readthedocs.io/en/latest/. Instructors and students may need to create a CyVerse account to access StarBLAST-VICE and an XSEDE account to access StarBLAST-Docker and HPC.

Acknowledgements

The Fall 2019 Applied Concepts in Cyberinfrastructure class at the University of Arizona. Benjamin Tovar, Douglas Thain and the CCTools team, the Sequenceserver team, Wilson Leung and the Genomic Education Alliance.

Funding

This work was supported by the National Science Foundation [grant numbers IOS–1849708, IOS–1743442, IOS–1444490].

References

Albrecht, M., Donnelly, P., Bui, P., & Thain, D. (2012). *Makeflow: A portable abstraction for data intensive computing on clusters, clouds, and grids*. 1–13. https://doi.org/10.1145/2443416.2443417

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & Madden, T. L. (2009). BLAST+: Architecture and applications. *BMC Bioinformatics*, *10*(1), 421. https://doi.org/10.1186/1471-2105-10-421

Devisetty, U. K., Kennedy, K., Sarando, P., Merchant, N., & Lyons, E. (2016). Bringing your tools to CyVerse discovery environment using docker. *F1000Research*, *5*. https://doi.org/10.12688/f1000research.8935.1

Form, D., & Lewitter, F. (2011). Ten simple rules for teaching bioinformatics at the high school level. *PLOS Computational Biology*, *7*(10), e1002243. https://doi.org/10.1371/journal.pcbi.1002243

Johnson, M., Zaretskaya, I., Raytselis, Y., Merezhuk, Y., McGinnis, S., & Madden, T. L. (2008). NCBI BLAST: A better web interface. *Nucleic Acids Research*, *36*, W5–9. https://doi.org/10.1093/nar/gkn201

Madden, T. L. (2002). The BLAST sequence analysis tool [updated 2003 aug 13]. *McEntyre J, Ostell J, Editors. The NCBI Handbook. Bethesda (MD): National Center for Biotechnology Information (US)*, 1–15. https://unmc.edu/bsbc/docs/NCBI_blast.pdf

Merchant, N., Lyons, E., Goff, S., Vaughn, M., Ware, D., Micklos, D., & Antin, P. (2016). The iPlant collaborative: Cyberinfrastructure for enabling data to discovery for the life sciences. *PLOS Biology*, *14*(1), e1002342. https://doi.org/10.1371/journal.pbio.1002342

Priyam, A., Woodcroft, B. J., Rai, V., Moghul, I., Munagala, A., Ter, F., Chowdhary, H., Pieniak, I., Maynard, L. J., Gibbins, M. A., Moon, H., Davis-Richardson, A., Uludag, M., Watson-Haigh, N. S., Challis, R., Nakamura, H., Favreau, E., Gómez, E. A., Pluskal, T., … Wurm, Y. (2019). Sequenceserver: A modern graphical user interface for custom BLAST databases. *Molecular Biology and Evolution*, *36*(12), 2922–2924. https://doi.org/10.1093/molbev/msz185

Shaffer, C. D., Alvarez, C., Bailey, C., Barnard, D., Bhalla, S., Chandrasekaran, C., Chandrasekaran, V., Chung, H.-M., Dorer, D. R., Du, C., Eckdahl, T. T., Poet, J. L., Frohlich, D., Goodman, A. L., Gosser, Y., Hauser, C., Hoopes, L. L. M., Johnson, D., Jones, C. J., … Elgin, S. C. R. (2010). The genomics education partnership: Successful integration of research into laboratory classes at a diverse group of undergraduate institutions. *CBE—Life Sciences Education*, *9*(1), 55–69. https://doi.org/10.1187/09-11-0087

Stewart, C. A., Turner, G., Vaughn, M., Gaffney, N. I., Cockerill, T. M., Foster, I., Hancock, D., Merchant, N., Skidmore, E., Stanzione, D., Taylor, J., & Tuecke, S. (2015). *Jetstream: A self-provisioned, scalable science and engineering cloud environment*. 1–8. https://doi.org/10.1145/2792745.2792774