

An Open-source Azure Solution for Scalable Genomics Workflows

Fan Yang-Turner[†], Lawrence Gripper^{*}, Jeremy Swann[†], Trien Do[†], Dona Foster[†], Denis Volk[†], Anita Ramanan^{*}, Marcus Robinson^{*}, Tim Peto[†] and Derrick Crook[†]

[†]Nuffield Department of Clinical Medicine, University of Oxford, Oxford, United Kingdom

^{*}Microsoft CSE, London, United Kingdom

Abstract—We present an open-source Azure solution for running scalable genomics workflows. It benefits from state-of-art distributed workflow framework, container and cloud technologies and allows users to create a cluster that is scaled to suit their workload in minutes. We describe the design decisions, solution testing and automation options to support a variety of users for their genomic data analytics. The solution demonstrates a generic and customizable approach to run genomic data analytics workflows on a cloud environment.

Keywords—scientific workflow, container, cluster system, genomic data analytics, bioinformatic pipeline, cloud computing

I. INTRODUCTION

Running bioinformatic pipelines on cloud infrastructure adds scalability, accessibility, potential cost gains and collaborative research opportunities [1]. There are scientific workflow solutions executed on the cloud, which utilize containers to solve the software deployment problems and resource utilization inefficiencies [2][3]. With cloud and containers as ideal technologies, genomics workflows can still be in a variety of design with diverse behaviors, resources requirements (CPU, memory, disk and network) and file system demands to share input and output files between instances in a distributed cluster. To execute different genomics workflows that used container technology, we developed an open-source, generic and customizable solution to scale genomic data analytics on Azure cloud (source code available at [4]).

II. AZURE SOLUTION OF SCALABLE GENOMICS COMPUTE

Data analytics requires three core resources: CPU, memory and storage. A cluster distributes work across a set of machines, adding a fourth resource, network. Our solution uses cloud computing to allow users to create a cluster in minutes that suits their workload while abstracting the resources. Resources appropriate to a specific workflow can be allocated and the scale of the cluster can be dynamically changed at deployment and during use, in response to workload.

We used a Domain Specific Language (DSL), Azure Resource Manager (ARM) [5], to automate cluster formation on the Azure cloud. Our solution sets up the required resources (CPU, memory, storage and network) and provides software to be configured on machines. The cluster is created via the ARM template using an API call or from the web portal GUI. After

this, ARM monitors the deployment progress, outputting the resulting connection details.

We utilize a higher order concept called ‘Virtual Machine Scale Sets’ (VMSS) [6] to simplify the cluster. VMSS handles the management of N cluster nodes as a single resource and allows users to focus on key tasks, such as increasing processing speed, rather than considering configuration and administration of individual machines. The deployment consists of a master node with persistent storage and N worker nodes. Worker node speciation is set independently of the master, which allows for a larger master node to account for its additional orchestration role.

We use Nextflow [7] as a workflow framework which requires a Portable Operating System Interface (POSIX) compatible storage layer, therefore each node in the cluster is provided with two shared storage volumes mounts – Network File System (NFS) and Common Internet File System (CIFS). The NFS share is hosted by the master node within the private cluster network. Storage of the CIFS share is outside the cluster and allows synchronization of tasks during setup, as well as initialization of files before cluster creation.

Nextflow and its dependencies are configured by an ARM extension that executes a bash script on each node. The solution also provides a customized script feature allowing users to pass further bash scripts to perform extra configuration steps for their specific needs.

III. SOLUTION TESTING

We tested our solution with two genomics workflows, the first runs the ARIBA (Antimicrobial Resistance Identification By Assembly) resistance prediction tool [8] and the second runs Tuberculosis (TB) Whole Genome Sequencing (WGS) analysis.

A. Running ARIBA and scaling the cluster

The ARIBA pipeline has three major steps: 1) *bam_to_paired_fastq*, which converts the files to a text-based format; 2) *run_ariba*, which runs ARIBA tool and 3) *run_ariba_summary*, which produces a summary file of processed samples. A compute-optimized virtual machine, Standard_F16s (8C/16T 2.4 GHz Intel Xeon E5-2673 v3 CPU, 32GB RAM, 64GB SSD) was selected as head node and four memory-optimized virtual machines, Standard_E4s_v3s (four

cores of a 10C/20T 2.30GHz Intel Xeon E5-2673 v4 CPU, 32GB RAM, 64GB SSD) were set as processing nodes. Cluster scaling was investigated by processing the data on 4-node and 30-node clusters with the same specifications as those above. The resulting execution times and a reduction in the time taken for the *bam_to_paired_fastqs* processes from 16m 43s to 8m 12s are shown in TABLE I.

TABLE I. AVERAGE RUNTIMES OF DIFFERENT CLUSTER ARRANGMENTS AND PROCESSES PER SAMPLE

Process	4-Node	30-Node
<i>bam_to_paired_fastqs</i>	16m 43s	08m 12s
<i>run_ariba</i>	02m 17s	02m 51s
Total	18m 00s	10m 06s

B. Running a TB WGS Pipeline

We used a TB WGS pipeline previously developed using Nextflow with Singularity and ran in a High Performance Computing (HPC) environment. To execute the pipeline on Azure, we need to prepare the artefacts on Azure storage before cluster deployment. To achieve this, we tested the accessibility of CIFS storage in a different Azure resource group and used the customized script feature of the solution to mount those CIFS storage to all the nodes of the cluster. We successfully completed test runs of this pipeline with following settings summarized in TABLE II.

TABLE II. SUMMARY OF TB WGS PIPELINE RUNNING

No. of Samples	Worker Nodes Set up and Scale	CPU Cores	NFS Disk Usage	Execution Time
20	Start 2, scale to 5	20	90G	1h 35 m
40	Start 2, scale to 10	40	187G	1h 45 m
90	Start 4, scale to 12	48	618G	3h 35 m
205	Start 4, scale to 15	60	1800G	8h 10 m

These successful test runs indicate that we can execute similar workflows on Azure by selecting fit-for-purpose resources. The performance data collected will help us set out directions for further investigation, such as NFS storage requirements, the scale strategy and the cost and performance implications of different worker nodes etc.

IV. AUTOMATION OF GENOMICS WORKFLOWS ON AZURE

Azure provides APIs and protocols for developing applications to automate genomics workflows. These applications would provide non-technical users in medical research with more effective and friendly user interfaces to operate the pipelines on cloud. Azure APIs in various programming languages (e.g. .NET, Java, Python) enable organizations to develop applications that interact with Azure's resources (e.g. storage or computing nodes). These applications hide the complexities of Azure's technical interactions.

To support global distributed users using a variety of operating systems, an ASP C#.Net Web application was prototyped to orchestrate the operation of genomics workflows on Azure. This application uses Azure Service Principal to authenticate, Fluent APIs to interact with Azure's resources, VMSS template (file *azuredploy.json* of the solution [4]) to

deploy clusters, and SSH to run commands and scripts on the head node of clusters to control genomics workflows.

V. DISCUSSION AND FUTURE WORK

We have presented an open-source, generic and customizable solution for genomic data analytics on Azure. Our paper makes three contributions: we introduce an end-to-end solution for running genomics workflows on Azure with state-of-art technologies; we evaluate the scalability of the solution with two genomics workflows; we have a web user interface prototype to prove the concept of automating the process, which hides the complexity of scalable compute from end users.

We are aware that our current solution has limitations. It works only on a fixed technical stack: Nextflow framework, Docker or Singularity containers and Azure cloud. We would like to work on different cloud infrastructures to improve performance, usability and flexibility. To support genomic data analytics, scalable compute is only one part of the whole story. In future, we will work towards a comprehensive evaluation and integration of our solution with distributed meta-data management to serve our users with a complete genomic data analytics experience using the cloud.

ACKNOWLEDGEMENT

This research was funded by the National Institute for Health Research (NIHR), Oxford Biomedical Research Centre (BRC) and the National Institute for Health Research Health Protection Research Unit (NIHR HPRU) in Healthcare Associated Infections and Antimicrobial Resistance at University of Oxford in partnership with Public Health England (PHE). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR, the Department of Health or Public Health England. We thank Microsoft CSE for their Azure technical assistance and anonymous reviewers for their helpful feedback.

REFERENCES

- [1] J. Leipzig, A review of bioinformatic pipeline frameworks. *Briefings in bioinformatics*, 18(3) pp. 530–536, 2017.
- [2] C. Zheng and D. Thain, Integrating Containers into Workflows: A Case Study Using Makeflow, Work Queue, and Docker, in *Proceedings of the 8th International Workshop on Virtualization Technologies in Distributed Computing*, Jun. 2015.
- [3] W. Gerlach, W. Tang, K. Keegan et al., Skyport: Container-based Execution Environment Management for Multi-cloud Scientific Workflows, in *5th International Workshop on Data-Intensive Computing in the Clouds*, Nov. 2014.
- [4] Nextflow on Azure, <https://github.com/Azure/azure-quickstart-templates/tree/master/nextflow-genomics-cluster-ubuntu> (accessed on 5/15/2018)
- [5] M. Eijk van, Getting Started with Azure Resource Manager, Microsoft Azure and Microsoft Azure Stack White Paper, Sep. 2016.
- [6] What are virtual machine scale sets in Azure? <https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-overview> (accessed on 5/15/2018)
- [7] Di Tommaso P, Chatzou M, Floden EW et al. "Nextflow enables reproducible computational workflows." *Nat Biotechnol* 35, pp. 316–319. 2017.
- [8] M. Hunt, A. E. Mather, L. Sánchez-Busó, A.J. Page, J. Parkhill, J.A. Keane et al. ARIBA: rapid 272 antimicrobial resistance genotyping directly from sequencing reads. *Microb Genomics*. Sep. 2017.