Characterizing Performance Variation of Genomic Data Analysis Workflows on the Public Cloud

David Perez, Ling-Hong Hung, Sonia Xu, Ka Yee Yeung, Wes Lloyd
School of Engineering and Technology
University of Washington
Tacoma, Washington USA
daperez, lhhung, sxu253, kayee, wlloyd@uw.edu

Abstract—**Public Infrastructure-as-a-Service (IaaS) clouds abstract the physical hardware implementation of resources provided to users. Users are not informed about the exact physical location of their virtual machines (VMs), the specific hardware used, the number of co-resident VMs they reside with, or the workloads that co-resident VMs are running. Detecting when VMs underperform can help identify resource contention from co-resident VMs to spur their replacement. In addition, resource utilization metrics may help classify performance of runs for use in VM performance model datasets that sample the distribution of performance outcomes. VM performance models are key to optimizing the cost of bioinformatics analyses in the public cloud. In this paper, we investigate performance variation of running big data genomics workflows in the public cloud. We examine causes of performance variation including VM provisioning, CPU heterogeneity, and resource contention. We leverage Amazon Elastic Compute Cloud placement groups, a feature designed to help influence VM placement on Amazon EC2 to help examine how VM placement impacts performance variation. As a use case, we investigate the performance of a multi-stage bioinformatics RNA sequencing (RNA-seq) analytical workflow consisting of four distinct phases, executing in ~90 minutes on average on 8-core public cloud VMs. In addition, we investigate whether Linux resource utilization metrics collected by profiling workflow runs can help identify performance variations.**

I. Introduction

In public clouds, *provisioning variation* refers to the random nature of VM placement across physical hosts that occurs as a result of the load balancing of VM launch requests by the cloud provider. Where VMs are hosted on public clouds is abstracted, and is considered a challenge to infer in real-time [1][2][3][4][5][6][7]. Public clouds support features including availability zones, virtual private networks, and placement groups to help consolidate VMs. These features can help influence VM placement relative to other user VMs for application hosting to help improve performance. Though user VMs may be controlled, shared physical hosts can still be busy with other co-resident resource hungry VMs that consume an unusual share of CPU, memory, disk, or network resources. *Resource contention* has been shown to degrade performance of scientific applications hosted in public clouds [8][9][10].

CPU heterogeneity occurs in the public cloud where cloud providers implement the same VM type using more than one CPU type. Farley et al. first discovered CPU heterogeneity on Amazon EC2 VMs of the same instance type [11]. Farley's work focused on the m1.small instance type demonstrating cost savings by discarding VMs with lower performing CPUs. Ou et al. identified heterogeneous VM implementations on multiple public clouds and found at least four different Intel Xeon CPUs used to implement the m1.large EC2 instance type producing performance variation of 20% for operating system benchmarks [12]. We developed a "trial-and-better" approach where the CPU type of VMs are checked upon launch, and those with lower performing CPUs are terminated and replaced. Lloyd et al. tested 12 different EC2 VM types and found that 25% were implemented with more than one CPU [10]. By leveraging the trial-and-better approach, Lloyd demonstrated potential for up to 14% performance improvement for RESTful environmental modeling web service workloads.

In this paper, we investigate the implications of VM provisioning variation and CPU heterogeneity on the performance of a multi-stage bioinformatics RNA sequencing (RNA-seq) workflow. We investigate the performance of running concurrent instances of this workflow across c5.2xlarge EC2 instances equipped with 8 vCPUs. Running the RNA-seq workflow concurrently on the cloud is a common scenario for exploratory investigations over genomics data. We leverage EC2 placement groups to control VM placement as much as possible and study runtime implications. Our empirical experiments show that c5 instances, considered the current generation of compute optimized VMs in us-east-2 (Ohio), exhibit CPU heterogeneity. Nearly half of the instances used are the Intel Xeon Platinum 8124M CPU and the other half are the Intel Xeon Platinum 8275CL CPU. This CPU heterogeneity produced a difference between min/max performance of 19.5% for RNA-seq spanning from a minimum of 82m 28s (8275CL) to a maximum of 98m 35s (8124M). As workflows are deployed thousands of times, this performance variation translates to performance losses and cost increases for big data analyses.

When genomics workflows underperform on the public cloud, we are interested in developing techniques to automatically identify underperforming VMs in real-time so they can be replaced. Additionally, when profiling resource utilization of workflows to train VM performance models there is a desire to adequately sample the entire input space to capture the full spread of possible runtimes for a workflow (e.g. 19.5% for c5.2xlarge). To ensure training data adequately covers the input space, we aim to develop techniques that can suggest where a profiling sample lies across the distribution before knowing the distribution. We investigate Linux profiling metric relationships with workflow runtime to identify relationships to spur this effort.

A. *Research Questions*

This paper investigates the following research questions:

RQ-1: What is the performance variation of running genomics data analytical workflows on the public cloud? How much do factors such as provisioning variation, CPU heterogeneity, and resource contention contribute to performance variation? How does performance compare to analyses on isolated hosts?

RQ-2: What relationships exist between Linux resource utilization metrics (CPU, memory, disk, and network) and workflow runtime? Which metrics trend negatively or positively with runtime? Can these relationships help infer where a workflow's runtime lies along the distribution of runtimes for a particular VM?

II. Background

A. *CPU Heterogeneity*

Public cloud providers largely have chosen to offer distinct types of VMs to cloud users to simplify the task of resource allocation to users. By fixing VM resources to have distinct quantities of virtual CPUs (vCPUs), memory, storage capacity, and network bandwidth, cloud providers can focus on optimizing hardware to deliver these resources in a highly available and scalable manner. For example, the Amazon, Microsoft, and Google public clouds presently offer more than 265, 204, and 35 fixed VM types each with predefined hardware specifications for the number of vCPUs, RAM size, storage type and capacity, and network bandwidth. As cloud hardware ages, however, cloud providers are forced to replace aging hardware to implement existing VM types with new CPUs. This CPU heterogeneity has been shown to produce performance variation for a variety of application workloads [10][11][12].

TABLE I. EC2 C5.2xlarge Heterogenous Processors: Intel Xeon Platinum 8124M vs intel Xeon Platinum 8275 CL

	Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz	**Intel(R) Xeon(R) Platinum 8275CL CPU @ 3.00GHz**
Family/microns/yr	Skylake/14nm/2017	Cascade Lake/14nm/2019
vCPUs/host	72	96
Physical CPU cores/host	36	48
Base clock MHz	3000	3000
Burst clock MHz (single/all)	3400 / 3500	3600 / 3900
L1 cache:	1.125 MiB (½ data, ½ code)	1.75 MiB (½ data, ½ code)
L2 cache:	18 MiB	24 MiB
L3 cache:	24.75 MiB	35.75 MiB
Total Freq.	53% (16 VMs)	47% (14 VMs)
Standard Freq.	13% (4 VMs)	20% (6 VMs)
Cluster Freq.	13% (4 VMs)	20% (6 VMs)
Spread Freq.	27% (8 VMs)	7% (2 VMs)

In this paper we focus on performance analysis of the RNA-seq workflow on c5.2xlarge Amazon EC2 instances. These VMs are equipped with 8 vCPUs, 16 GB RAM, EBS storage, and up to 10 Gigabit network throughput. For this work we created ~30 VMs, where 16 were randomly implemented with the Intel Xeon Platinum 8124M CPU, and 14 with the Intel Xeon Platinum 8275CL CPU. Comparison of these two CPUs and their occurrence rates observed with different VM placement groups appears in Table I.

B. *VM Placement Groups*

Amazon EC2 offers VM placement groups to help influence placement of VMs in the public cloud [13][14]. Options include spread, cluster, and partition placement groups. With spread placement, AWS places instances on distinct hardware using distinct racks, where each rack has its own network and power source to maximize dispersion. Spread placement is limited to 7 VMs per availability zone forcing us to use two availability zones to obtain 10 distinct VM placements in the us-east-2 Ohio region for our experiments. Spread placement guarantees that no two VMs will be co-located with each other. Given that genomics workflows are both CPU and I/O intensive, co-locating all concurrent runs on the same hardware will result in interference between runs producing resource contention. Spread placement guarantees user VM's won't interfere with each other, but it does not guarantee resource isolation from other user's VMs.

Partition placement is similar to spread placement but allows for more than one VM to exist in each partition allowing for distinct destinations for VMs. Users are limited to 7 partitions per availability zone. We do not investigate partition placement groups here because RNA-seq workflows run standalone on individual VMs and we do not want to co-schedule concurrent runs on the same hardware.

Cluster placement packs instances close together inside an Availability Zone, to ensure the lowest possible network latency and the highest possible network throughput up to 10 Gbps for TCP/IP traffic. Instances in a cluster placement group are placed on the same rack, or on racks close to one another in the cloud data center. Cluster placement for concurrent jobs of the same type may increase resource contention and reduce performance when VMs that share the same host run identical workflows

III. Methodology

A. *UMI RNA-seq Workflow*

As a use case we study the performance of the multi-stage bioinformatics analytical workflow (RNA sequencing using unique molecular identifiers) [15]. To reduce computation time and cost as we performed 330 workflow runs, we used a partial dataset generated by excluding all but the first million reads from the original FASTQ files. The workflow consisted of 4 distinct phases each requiring different computational resources to execute. The first phase is a download phase where the workflow downloads input data (8 GB of FASTQ files). The second phase is a split step where data demultiplexing is performed. Data is sorted using a sequence barcode to identify the originating sample. The third phase aligns the reads to a human reference genomic sequence to identify the gene that produced the transcript. The final stage is the "merge" phase which counts all the aligned reads to identify the number of transcripts produced by each gene. RNA-seq was deployed on EC2 instances using a Docker container with Ubuntu 16.04 LTS as the host operating system. A VM image was created which included Docker and all required software dependencies for use in launching VMs.

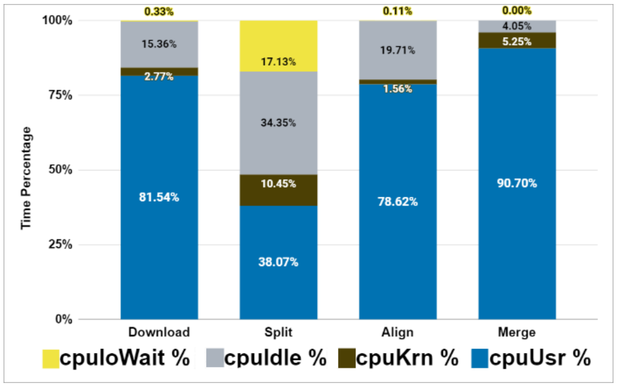

Fig. 1. CPU utilization graph for the four phases (download, split, align, and merge) of the RNA-seq workflow. The graph depicts % CPU time in each CPU mode: cpuIOWait, cpuIdle (CPU idle time), cpuKrn (CPU kernel mode), and cpuUsr (CPU user mode) [16].

Figure 1 summarizes CPU time of the different RNA-seq phases by distinct CPU mode: user mode time, kernel mode time, I/O wait time, soft interrupt service time, and idle time. The Download phase is limited by the network bandwidth and the split phase by disk I/O. These phases bound by I/O exhibit high cpuIdle time. The align and merge steps are CPU-bound and most of the CPU time is accounted for in CPU user mode. For our experiments, we profile resource utilization of RNA-seq for the entire workflow (all four phases), and for the alignment phase, the longest phase (73% of runtime).

B. Container Profiler

To profile resource utilization of the RNA-seq workflow in our experiments, the Container Profiler tool was used [16]. The Container Profiler measures and records resource utilization of any containerized task capturing over 50 individual metrics to characterize CPU, memory, disk, and network utilization at the VM, container, and process levels. All experimental data were obtained using the Container Profiler including the runtime of each workflow phase.

C. Cloud Infrastructure for Experiments

In experiment #1 (**RQ-1**), we profiled RNA-seq using 30 x AWS ec2 c5.2xlarge instances using three different Amazon EC2 placement groups to test for performance variation. We launched 10 VMs using each placement group: standard placement (i.e. no strategy, standard VM launch), spread, and cluster. We leveraged these 30 instances to run the multi-stage UMI RNA-seq workflow 3 times each, for a total of 90 different runs. For these instances we received 16 with the Intel Xeon 8124M processor, and 14 with the Intel Xeon 8274CL processor. Processor breakdowns by placement group are described in Table I.

For experiment #2 (**RQ-2**), to test for relationships between Linux resource utilization metrics and RNA-seq workflow runtime we launched 16 x AWS ec2 c5.2xlarge instances. 9 instances were created using standard placement, and 7 instances were created with cluster placement. For standard placement we received 77.7% Xeon 8124M CPUs, and 22.3% 8275CL CPUs. For cluster placement we received 71.4% Xeon 8124M CPUs, and 28.6% Xeon 8275CL CPUs for a total of 12 x 8124M CPUs and 4 x 8275CL CPUs. Each instance ran the RNA-seq workflow 15 times over a 24-hour period for a total of 240 runs. By running 15 consecutive iterations of the RNA-seq on each VM we sought to observe if workflow performance was constant or variable over a 24-hour period. Persistently slow VMs, once identified, can be replaced to improve throughput and runtime while lowering cloud computing costs.

IV. Experimental Results

A. RNA-seq Public Cloud Performance Variation

To determine performance variation for RNA-seq (**RQ-1**), 30 runs were profiled using each VM placement strategy (e.g. standard, spread, and cluster). An additional 3 runs were completed on a c5.2xlarge ec2 dedicated host, a private isolated cloud server not shared by any other users to benchmark performance when there is no resource contention.

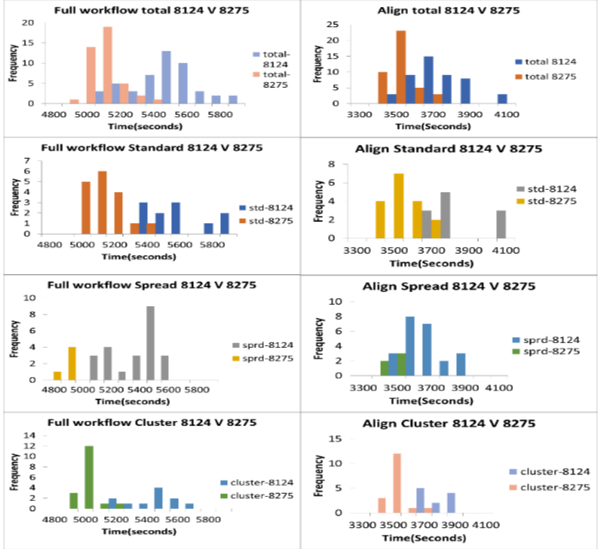

Fig. 2. RNA-seq runtime distribution graphs for c5.2xlarge instances by VM placement group and processor type. Graphs on the left depict runtime distributions for all four stages of the UMI RNA-seq workflow. Graphs on the right depict runtime distribution for the alignment phase of the workflow.

Figure 2 depicts the runtime distributions of RNA-seq for each placement group and processor for the entire workflow, and just the alignment phase. Most runs on the 8275CL processor outperform the 8124M. Figure 2 depicts the challenge of capturing training data for VM performance models. CPU heterogeneity increases the sample space of the distribution. Statistically, instance types with heterogeneous CPUs will be best handled by separating data for each CPU into separate distribution curves. The probability of obtaining a particular CPU should be considered. Figure 2 also highlights the distribution of processors for each VM placement strategy.

Tables II and III detail runtime statistics for workflow runtime on the 8124M and 8275CL CPUs. The tables include the percent runtime variation which captures the difference between the minimum and maximum. We also calculate the coefficient of variation (CV), which is equal to the standard deviation over the mean. Table II also details runtime on an EC2 dedicated host. Using an isolated host reduced runtime over standard VMs by 10.16% on average, and by 16.44% in the extreme case.

TABLE II. c5.2xlarge (Xeon 8124M)
RNA-seq Workflow Runtime Summary

	Standard	Cluster	Spread	Dedicated Host
Max Runtime (sec)	5915	5813	5667	5091
Min Runtime (sec)	5383	5334	5238	5065
Average (sec)	5596.33	5577.83	5396.83	5080.67
(%) Runtime Variation	9.51%	8.59%	7.95%	0.51%
Coefficient of Variation	3.47%	2.68%	3.28%	0.27%

TABLE III. c5.2xlarge (Xeon 8275CL)
RNA-seq Workflow Runtime Summary

	Standard	Cluster	Spread
Max Runtime (sec)	5375	5289	5040
Min Runtime (sec)	4983	5013	4948
Average (sec)	5129.39	5110.67	4995.33
(%) Runtime Variation	7.64%	5.40%	1.80%
Coefficient of Variation	2.13%	1.36%	0.61%

For both processors, the runtime distribution is the greatest when creating an instance with standard VM placement in the public cloud, and smallest with spread placement. CV is also greatest for standard cloud placement (3.47%). Spread placement provided the lowest average runtime for both CPUs. These results demonstrate up to a 19.5% performance variation for RNA-seq on the c5.2xlarge EC2 instance type with differences explained by CPU heterogeneity (8124M vs. 8275CL), resource contention (standard vs. dedicated host), and VM placement (standard vs. spread).

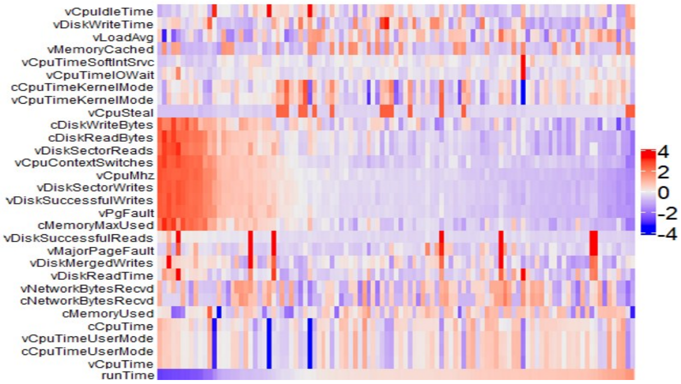

Fig. 3. Resource utilization heatmap (8124M CPU) for the alignment phase with clustered (ordered) rows. Negative correlations between profiling metrics and runtime can be seen. Columns depict 180 individual workflow runs from left to right sorted by increasing runtime.

B. Resource Utilization Relationships with Workflow Runtime

We next investigated relationships between Linux resource utilization metrics collected by the Container Profiler with RNA-seq workflow runtime (**RQ-2**) over 240 runs. 75% of the runs ran on the 8124M CPU, while 25% ran using the 8275CL. We normalized metrics using per minute averages to investigate correlations with workflow runtime. Several metrics had statistically significant negative correlations with workflow runtime (p<.01). Correlations include, VM metrics: disk sector reads, CPU context switches, disk sector writes, # of successful disk writes, and page faults; Container metrics: disk read bytes, and max memory used.

Figure 3 provides a heatmap that visualizes relationships between workflow runtime and Linux resource utilization metrics. A cluster of inverse relationships with runtime is seen including *(container metrics):* cDiskWriteBytes, cDiskReadBytes, cMemoryMaxUsed, and *(VM metrics):* vDiskSectorReads, vCpuMhz, vCpuContextSwitches, vDiskSectorWrites, vDiskSuccessfulWrites, and vPgFaults. As future work, we will investigate machine learning classifiers to characterize VM performance.

Acknowledgments

This research is supported by the US National Science Foundation Advanced Cyberinfrastructure Research Program (OAC-1849970), NIH grants R01GM126019, and R01GM126019-02S2.

References

[1] Z. Yang, H. Fang, Y. Wu, C. Li, B. Zhao, and H. H. Huang, “Understanding the effects of hypervisor I/O scheduling for virtual machine performance interference,” in *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, 2012, pp. 34–41.

[2] T. Ristenpart and E. Tromer, “Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds,” *... Conf. Comput. ...*, pp. 199–212, 2009.

[3] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, “On detecting co-resident cloud instances using network flow watermarking techniques,” *Int. J. Inf. Secur.*, 2014.

[4] M. S. İnci, B. Gulmezoglu, T. Eisenbarth, and B. Sunar, “Co-location detection on the cloud,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[5] J. O’Loughlin and L. Gillam, “Sibling virtual machine co-location confirmation and avoidance tactics for Public Infrastructure Clouds,” *J. Supercomput.*, 2016.

[6] V. Varadarajan, T. Kooburat, B. Farley, T. Ristenpart, and M. M. Swift, “Resource-freeing attacks: Improve your cloud performance (at your neighbor’s expense),” in *Proceedings of the ACM Conference on Computer and Communications Security*, 2012.

[7] X. Han, R. Schooley, D. Mackenzie, O. David, and W. J. Lloyd, “Characterizing Public Cloud Resource Contention to Support Virtual Machine Co-residency Prediction,” in *2020 IEEE International Conference on Cloud Engineering (IC2E)*, 2020, pp. 162–172.

[8] M. S. Rehman and M. F. Sakr, “Initial findings for provisioning variation in cloud computing,” in *Proceedings - 2nd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2010*, 2010, pp. 473–479.

[9] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, “Runtime measurements in the cloud: observing, analyzing, and reducing variance,” *Proc. VLDB Endow.*, vol. 3, pp. 460–471, 2010.

[10] W. Lloyd, S. Pallickara, O. David, M. Arabi, and K. Rojas, “Mitigating resource contention and heterogeneity in public clouds for scientific modeling services,” in *Proceedings - 2017 IEEE International Conference on Cloud Engineering, IC2E 2017*, 2017.

[11] B. Farley, V. Varadarajan, K. D. Bowers, A. Juels, T. Ristenpart, and M. M. Swift, “More for your money: Exploiting performance heterogeneity in public clouds,” in *Proceedings of the 3rd ACM Symposium on Cloud Computing, SoCC 2012*, 2012.

[12] Z. Ou *et al.*, “Is the Same Instance Type Created Equal? Exploiting Heterogeneity of Public Clouds,” *IEEE Trans. Cloud Comput.*, vol. 1, pp. 201–214, 2013.

[13] “Placement Groups - Amazon Elastic Compute Cloud.” [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html#placement-groups-spread.

[14] “Reducing Latency with EC2 Placement Groups.” [Online]. Available: https://awsinsider.net/articles/2017/06/12/ec2-placement-groups.aspx.

[15] Y. Xiong *et al.*, “A comparison of mRNA sequencing with random primed and 3′-directed libraries,” *Sci. Rep.*, vol. 7, no. 1, pp. 1–12, 2017.

[16] H. Deng *et al.*, “Profiling Resource Utilization of Bioinformatics Workflows,” *arXiv Prepr. arXiv2005.11491*, 2020.