Performansi Pemrosesan Big Data pada Virtualisasi Berbasis Container dan Hypervisor

Muhammad Rashief[1] , Sidik Prabowo[2], Siti Amatullah Karimah[3]

[1,2,3] Fakultas Informatika, Universitas Telkom, Bandung
[1]rashiefm@student.telkomuniversity.ac.id, [2]pakwowo@telkomuniversity.ac.id,
[3]karimahsiti@telkomuniversity.ac.id

Abstrak

Hadoop merupakan tools yang sangat membantu dalam pemrosesan data berukuran besar secara terdistribusi. Hadoop MapReduce merupakan model pemrograman yang memproses dataset yang besar, karena Hadoop berhubungan dengan data yang besar, hal ini mempengaruhi kekuatan pada hardware yang digunakan. Dengan adanya teknik virtualisasi, kinerja Hadoop dapat dioptimalkan. Pada penelitian ini membandingkan dua jenis virtualisasi yaitu Container dan Hypervisor dimana keduanya memiliki arsitektur yang berbeda. Dengan melihat CPU Utilization pada Container dimana 6.9 kali lipat lebih tinggi dibanding dengan Hypervisor, membandingkan Disk I/O yang dimana Container lebih optimal dari Hypervisor yang mempengaruhi execution time dalam menjalankan wordcount job yang dimana Container mendapatkan waktu 2.48 kali lebih cepat dari Hypervisor. Container(Docker) mendapatkan performa yang lebih baik ketimbang Hypervisor(VMware) pada pengujian yang diberikan.

Kata kunci : hadoop, container, hypervisor, wordcount, CPU utilization, TestDFSIO

Abstract

Hadoop is a tool that is very helpful in processing large data in a distributed manner. Hadoop MapReduce is a programming model that processes large datasets, because Hadoop deals with large data, this affects the strength of the hardware used. With the virtualization technique, Hadoop's performance can be optimized. In this study comparing two types of virtualization, namely Container and Hypervisor where both have different architectures. Looking at CPU Utilization in Container where ten times higher than the Hypervisor, compares Disk I / O where Container is more optimal than Hypervisor which affects the execution time in running a wordcount job where Container gets four times faster than Hypervisor. The container (Docker) gets better performance than the Hypervisor (VMware) in the test given.

Keywords: hadoop, container, hypervisor, wordcount, CPU utilization, TestDFSIO

1. Pendahuluan

Big Data merupakan istilah yang mulai digunakan untuk menggambarkan proses penerapan kekuatan komputasi yang serius dan untuk kumpulan informasi yang sangat besar dan seringkali sangat rumit[1]. Kebutuhan akan kecepatan data yang besar membebankan tuntutan pada infrastruktur komputasi yang mendasarinya[2]. Hadoop adalah framework populer yang digunakan untuk analisis data tak terstruktur yang cepat dan hemat biaya[3]. Dimana dari beberapa ecosystem Hadoop, HDFS dan MapReduce merupakan inti dari komponen Hadoop, dimana HDFS dan MapReduce merupakan komponen untuk penyimpanan dan pemrosesan Hadoop[4]. Karena Hadoop digunakan untuk pemrosesan dan penyimpanan dataset yang sangat besar dan dirancang untuk bekerja pada node fisik, deployment dan maintenance-nya memiliki biaya yang mahal[5]. Dimana dengan mem-virtualisasi akan memberikan flexibility dan mengurangi biaya.

Teknologi virtualisasi yang ada dibebankan secara kasar antara solusi berbasis container dan hypervisor[6]. Virtualisasi berbasis container memberikan kinerja yang mendekati aslinya, dimana container bekerja pada tingkat sistem operasi dan berbagi kernel yang sama dengan host[7].

Latar Belakang

Saat ini hampir semua Hadoop cluster berjalan diatas virtual machine, semua perusahaan seperti Yahoo, dan Facebook menggunakan virtual machine untuk Hadoop cluster information. Dimana ketika meng-install OS pada virtual machine diharuskan meng-install keseluruhan pakage termasuk kernel, aplikasi dasar, dan lainnya[8]. Untuk mempresentasikan kedua virtualisasi tersebut, penelitian ini menggunakan Docker untuk container dan VMware untuk hypervisor dimana Docker saat ini adalah solusi container yang paling populer dan VMware adalah salah satu pimpinan di pasar hypervisor[6].

Pada penelitian terkait[6], peneliti mengatakan masih sedikitnya pembuktian kuantitatif untuk hipotesis yang diberikan dengan cara ”apple to apple”. Atas dasar tersebut, penelitian ini dilakukan dengan skenario yang berbeda dengan menjalankan Hadoop diatasnya, dan tingkat dataset yang bereda.

Topik dan Batasannya

Beberapa batas yang terdapat pada tugas akhir ini adalah :

- Hadoop yang dijalankan diatas container hanya menggunakan Docker.
- Hadoop yang dijalankan diatas hypervisor hanya menggunakan VMware.
- Hanya menggunakan satu physical machine.
- Menggunakan tiga jenis pengujian yaitu pengujian untuk execution time, disk I/O, dan CPU utilization.

Tujuan

Untuk tujuan penelitian yang bisa diambil dari latar belakang adalah :

- Melakukan perbandingan performansi Hadoop yang dijalankan diatas Container dan Hypervisor
- Mendapatkan hasil setelah melakukan Job yang diberikan untuk dilakukan komparasi

Organisasi Tulisan

Urutan penulisan laporan ini adalah sebagai berikut : Bagian 2 menunjukkan penelitian-penelitian terkait dengan tugas akhir ini. Sistem yang diajukan untuk analisis adalah Hadoop pada Hypervisor-Based Virtualization dan Hadoop pada Container-Based Virtualization di bagian 3. Pada bagian 4 akan didiskusikan mengenai hasil pengujian dan evaluasi sistem. Akhirnya, kesimpulan akan dipaparkan pada bagian 5.

2. Studi Terkait

2.1 Big Data

Big Data sebagian besar adalah penyimpanan data dan analisis data. Meski dalam Big Data, konsep-konsep ini jauh dari baru. ”Besar” atau ”Big” menyiratkan arti, kompleksitas, dan tantangan[1]. Dapat dikatan bahwa data didefinisikan menjadi lima karateristik berikut[2]:

1. Volume.

   Dimana jumlah data yang akan disimpan dan di analisis cukup besar sehingga memerlukan pertimbangan khusus.

2. Variety.

   Dimana data terdiri dari beberapa jenis data yang berpotensi dari berbagai sumber, perlu mempertimbangkan data terstruktur yang disimpan dalam tabel atau objek yang metadatanya terdefinisi dengan baik, data semistruktur yang disimpan sebagai dokumen atau serupa dimana metadata secara internal, atau tidak terstruktur yang dapat berupa foto, video, atau bentuk data biner lainnya.

3. Veracity.

   Dimana kebenaran dapat dinilai.

4. Velocity.

   Dimana data diproduksi dengan kecepatan tinggi(High Rate) dan beroperasi pada data ’stale’ yang belum mempunyai nilai.

5. Value.

   Dimana data telah diberi manfaat secara kuantitatif bagi perusahaan atau organisasi yang menggunakannya.

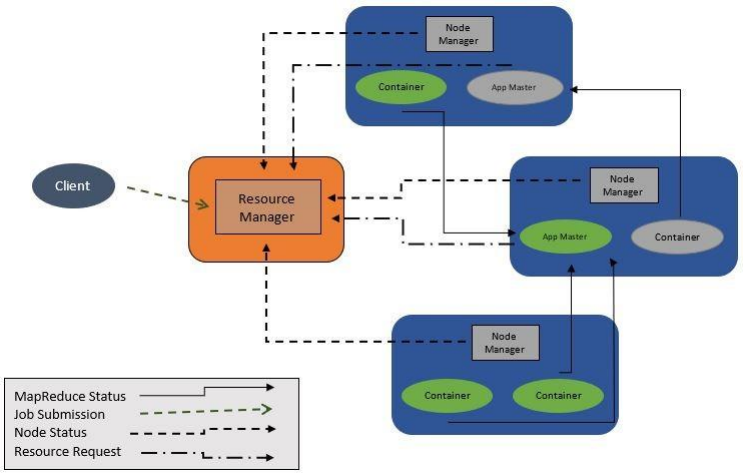
2.2 Hadoop Cluster

Hadoop adalah salah satu perangkat lunak open source, Hadoop digunakan untuk penyimpanan dan pemrosesan big data pada cluster yang besar. Hadoop adalah salah satu proyek tingkat atas dari Apache yang dijalankan dan digunakan oleh komunitas penyedia layanan dan pengguna layanan tingkat global. Menyediakan penyimpanan yang scalable untuk semua varitas data, dan meningkatkan pengolahan daya dan memberikan kelincahan yang lebih besar untuk menangani tugas bersamaan secara virtual[8]. Hadoop dirancang untuk bekerja dengan node fisik dan memberikan skalabilitas tinggi, fleksibilitas, sistem toleransi kesalahan. Proyek Hadoop meliputi Hadoop common, Hadoop Distributed File System atau HDFS, dan Hadoop MapReduce[8].

Multi-node terdiri dari lebih dari satu node yaitu master node, sleeve node dan sleeve node. Pada multi-node untuk master node, layer MapReduce hanya memiliki JobTracker dan pada layer HDFS hanya memiliki NameNode. Sedangkan pada sleeve node pada layer MapReduce hanya memiliki TaskTracker, sedangkan pada layer HDFS hanya memliki DataNode.

2.3 Hadoop Yet Another Resource Negotiator(YARN)

Yarn merupakan Hadoop MapReduce generasi kedua atau biasa disebut MRv2. Yarn terdiri dari ResourceManager dan NodeManager. Dimana ResourceManager memiliki otoritas tertinggi yang menegahi sumber daya diantara semua aplikasi dalam sistem. Dan untuk NodeManager dimana agen kerangka kerja pada sleeve node yang bertanggung jawab untuk kontainer, memantau penggunaan sumber daya(cpu, memori, disk, jaringan) dan melaporkan hal yang sama ke ResourceManager/Sceduler[9][10].



Gambar 1. YARN Architecture

2.4 Hadoop Distributed File System (HDFS)

Hadoop Distributed File System atau HDFS adalah sistem file terdistribusi, ini memiliki banyak kesamaan dengan sistem file terdistribusi yang ada. HDFS sangat toleran terhadap kesalahan dan dirancang untuk digunakan pada perangkat keras yang biayanya rendah. HDFS menyediakan akses throughtput tinggi ke data aplikasi dan cocok untuk aplikasi yang memiliki dataset yang besar[7]. HDFS memiliki dua tipe node yaituNameNode dan DataNode[11].

1. NameNode

   NameNode menjaga file metadata system, yang mencakup informasi tentang files dan directory tree dimana setiap blok data disimpan secara fisik.

2. DataNode

   DataNode menyimpan blok data sendiri, dimana blok data disimpan secara fisik setelah didistribusikan.

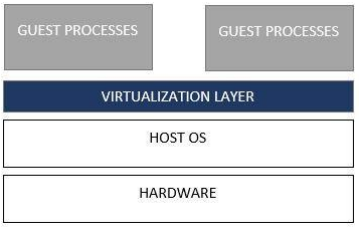
2.5 Virtualization

Ketika Hadoop cluster yang dikerahkan secara fisik bertambah besar, developer seringkali bertanya : bisakah kita memvirtualisasinya?[3]. Node adalah virtual machines (VMs) yanf ditunjuk dengan peran master atay pekerja/sleeve. Setiap VM dialokasikan komputasi tertentu dan penyimpanan sumber daya tertentu dari physical host,

dan sebagai hasilnya, dapat mengkonsolidasikan kelompok Hadoop mereka ke physical server yang jauh lebih sedikit[3].

2.5.1 Container Based-Virtualization

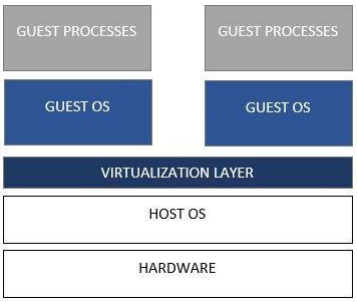
Container berbasis virtualisasi adalah metode virtualisasi yang bekerja pada tingkat sistem operasi dengan host kernel untuk membuat satu atau beberapa container[12], dimana dapat dilihat pada gambar 2.



Gambar 2. Container Architecture

2.5.2 Hypervisor Based-Virtualization

Hypervisor-Based Virtualization telah banyak digunakan selama dekade terahir untuk mengimplementasikan virtualisasi dan isolasi. Bertentangan dengan container, hypervisor beroperasi di tingkat perangkat keras, sehingga mendukung mesin virtual mandiri yang independen dan terisolasi dari host system. Karena hypervisor mengisolasi VM dari host system yang mendasarinya[13]. Dapat dilihat pada gambar 3.



Gambar 3. Hypervisor Architecture

2.6 Penelitian Terkait

Pada penelitian sebelumnya[7], analisis performansi ini dilakukan pada OpenNebula Cloud dimana menggunakan KVM dan OpenVZ sebagai masing-masing hypervisor dan container, pada penelitian ini hadoop dijalankan pada OpenNebula Cloud.

Pada kasus berikutnya[13], perbandingan performa dilakukan dengan membandingkan hypervisor dengan lightweight virtualization dimana adalah container. Hypervisor pada penelitian ini menggunakan KVM sedangkan untuk containernya sendiri menggunakan Docker. Untuk benchmark tools mengukur CPU, Memory, Disk I/O, dan Network I/O performance, dimana setiap pengukuran yang berbeda menggunakan benchmark tools yang berbeda pula. Sehingga hasil dari benchmark dapat dijadikan acuan untuk pemilihan tools virtualisasi maupun becnhmark tools yang akan digunakan lebih lanjut. Pada penelitian ini KVM dikelola menggunakan API libvirt Linux Standar dan tools chain(virsh), OSv dijalankan diatas KVM, LXC dan Docker dijalankan langsung pada host OS.

Penelitian sebelumnnya[14] ,hanya menggunakan VMware dengan empat node yaitu satu master node dan tiga sleeve node. Dimana melakukan benchmarking berupa MrBench, wordcount, TestDFSIO dan terasort. Dimana pada penelitian ini fokus pada tingkat data yang digunakan.

Pada penelitian berikutnya[6], perbandingan performance overhead antara hypervisor dengan container menggunakan Docker dan VMware. Dengan menggunakan ECS yang menyeddiakan Domain Knowlodge-driven Methodology(DoKnowMe), penelitian ini secara eksperimental mengekplorasi performance overhead dari solusi virtualisasi yang berbeda

3. Sistem yang Dibangun

3.1 Sistem yang Dibangun

Pada gambar 4 memperlihatkan alur pengambilan dampai pengolahan data, dimana dataset menggunakan data twitter yang telah di-crawling dengan menggunakan phyton code sedaharna, setelah itu data dimasukkan ke masing-masing virtualisasi yang akan dijelaskan di subbas selanjutnya.

Gambar 4. Arsitektur Sistem

3.2 Hypervisor-Based Virtualization

VMware digunakan untuk representasi dari Hypervisor, dimana pembuatan virtual machine dilakukan sebanyak tiga kali dengan menggunakan Ubuntu 18.04 sebagai OSnya. Tiga virtual machine ini digunakan untuk tiga node yang akan digunakan untuk Hadoop cluster dimana satu virtual machine untuk Master Node dan dua lainnya untuk Sleeve Node.

3.2.1 Hadoop On Hypervisor

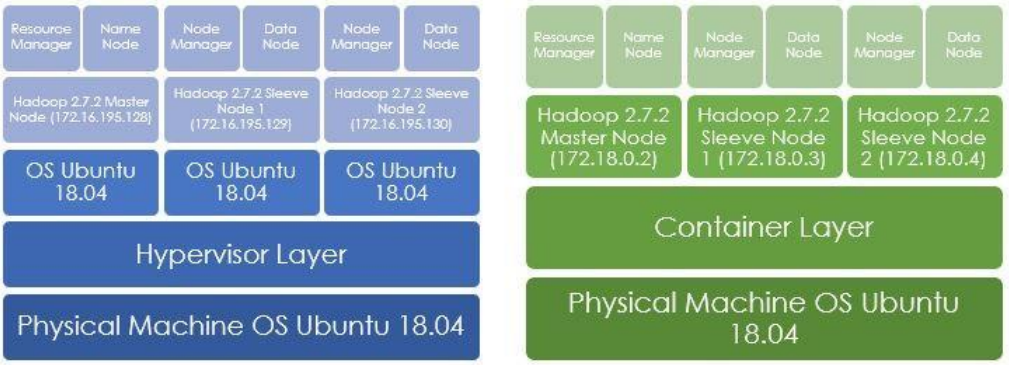
Hadoop yang digunakan adalah versi 2.7.2 dimana versi ini merupakan Hadoop versi 2 yang jobtracker pada master node menjadi resource manager dan tastracker pada sleeve node menjadi node manager dimana disebut sebagai YARN.

3.3 Container-Based Virtualization

Penelitian ini menggunakan Docker untuk Container dengan menggunakan image milik kiwenlau dimana image ini sudah dipull sebanyak lebih dari sepuluh ribu kali. Image ini sudah menyediakan tiga buah container dimana satu container untuk Master Node dan dua lainnya untuk Sleeve Node. Pull dilakukan di github untuk mendapat code atau command yang digunakan untuk image milik kiwenlau.

3.3.1 Hadoop On Container

Pada image kiwenlau sudah termasuk container dengan tiga node yang sudah diset sebagai Master Node dan dua Sleeve. Dengan men-create hadoop network menggunakan command ”sudo docker network create –driver=bridge hadoop”, dimana semua node yang telah di-create terhubung dengan network hadoop yang telah dibuat.Command ”sudo ./start-container” untuk menjalankan tiga node tersebut dan ”./start-hadoop.sh” untuk menjalankan HDFS dan Yarn(Mapreduce).



Gambar 5. Arsitektur yang Dibangun

3.4 System Specification

Tabel 1 menunjukkan spesifikasi pada physical machine yang akan digunakan sebagai host pada setiap virtualisasi Hypervisor maupun Container.

Komponen	Spesifikasi
Processor	AMD Ryzen 7 2700X Eight-Core Processor
RAM	DDR4 8GB + 8GB
HDD	SATA 1 TB

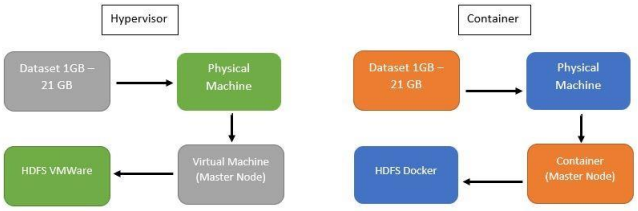
Tabel 1. Spesifikasi Physical Machine

3.5 Skenario Pengujian

Pada penelitian ini mempunyai tiga skenario pengujian yaitu adalah CPU Utilization untuk mengukur utilisasi kedua jenis virtualisasi, dimana menggunaka TOP sebagai alat ukur utilitas pada CPU saat menjalankan job maupun dalam kondisi idle atau tidak menjalankan job sama sekali. Kedua yaitu Disk I/O dimana untuk mengukurnya menggunakan benchmark TestDFSIO, TestDFSIO merupakan benchmark bawaan dari Hadoop itu sendiri, dimana TestDFSIO melakukan MapReduce job dan men-generate data random pada HDFS secara bersamaan dan ketiga yaitu execution time dimana job yang dilakukan adalah wordcount untuk melihat berapa lama waktu yang dibutuhkan untuk menyelesaikan job wordcount tersebut.

3.5.1 Dataset

Dataset pada penelitian ini menggunakan data hasil crawling dari twitter dengan menggunakan hashtag yaitu #jokowi, #prabowo dan #pilpres2019. Dengan memnggunakan code phyton untuk proses crawlingnya. Hasil dari crawling dataset terbagi menjadi delapan dataset yaitu 1GB, 3GB, 6GB, 9GB, 12GB, 15GB, 18GB, 21GB. Setelah data diperoleh, depalan dataset tersebut dimasukkan di masing-masing virtualisasi sebelum data tersebut masuk ke HDFS masing-masing virtualisasi yang dapat dilihat di gambar 6



Gambar 6. Flow Dataset Sampai Ke HDFS

3.5.2 CPU Utilization

Pengujian CPU Utilization dilakukan pada saat running job wordcount dilakukan. Dengan men-capture menggunakan tools TOP tiap 30 detik selama running job wordcount dilakukan dengan nilai yang didapatkan berupa persentase penggunaan CPU pada physical machine ditiap jenis virtualisasi. Capture dilakukan 30 detik sebelum running job dan 30 detik setelah running job untuk mendapatkan nilai persentase idle tiap dataset sebelum dan sesudah running job.

3.5.3 Disk I/O

Pada pengujian Disk I/O dilakukan dengan menggunakan tools benchmark TestDFSIO dimana TestDFSIO ini akan melakukan stress testing terhadap HDFS dengan melakukan write dan read dengan mendefinisikan dataset sesuai dengan jumlah dataset yang ada. Pada saat akan running TestDFSIO data yang didefinisikan akan mengikuti jumlah data pada pengujian executon time yaitu 1GB, 3GB, 6GB sampai 21GB. pendefinisian data dilakukan pada saat melakukan TestDFSIO write dan TestDFSIO read dengan jumlah nFiles sama dengan 1. Dan hasil yang digunakan adalah throughtput dan average I/O rate dengan satuan mb/sec.

3.5.4 Execution Time

Untuk menghitung execution time pada kedua jenis virtualisasi, wordcount job dilakukan sebanyak delapan kali menggunakan dataset yang telah dijelaskan pada subsection dataset . Execution time dihitung dari awal menjalankan job sampai job selesai dilakukan dengan hasil yang didapatkan dalam satuan menit.

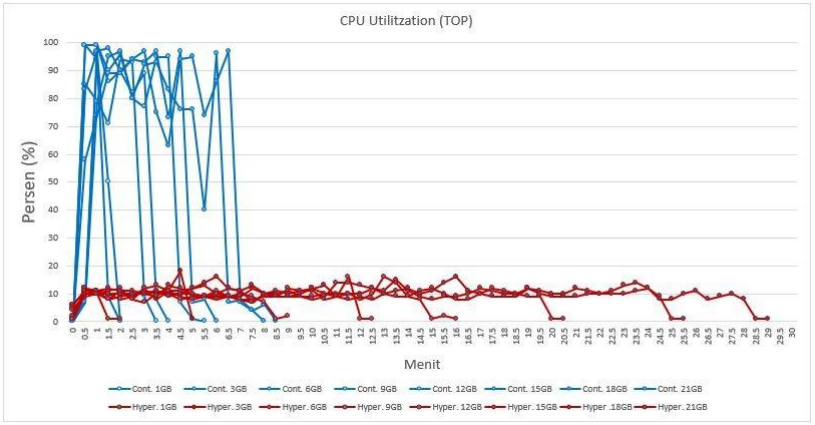
4. Evaluasi

Bagian ini menampilkan hasil analisis penulis. Seperti yang dijelaskan sebelumnya, parameter yang dipilih untuk diukur adalah persentase penggunaan CPU saat melakukan job menggunakan TOP, write throughtput, read throughtput, write avg I/O rate, read avg I/O rate pada saat melakukan TestDFSIO dan terakhir execution time saat melakukan wordcount job. Hasilnya disusun dalam tiga subbagian berbeda. Subbagian 4.1 menjelaskan CPU Utilization. Pengukuran throughtput dan avg I/O rate ditunjukkan dalam subbagian 4.2. Dan, subbagian 4.3 menjelaskan perbedaan execution time antara Hypervisor dan Container.

4.1 CPU Utilization

Terlihat pada gambar 7 grafik untuk Hypervisor, persentase penggunaan CPU rata-rata di angka 9.57%, dimana penggunaan resource pada saat menjalankan job tidak mencapai 100% bahkan dibawah 50%. Dimana pada grafik untuk Hypervisor, maksimal persentase hanya mencapai 18%. Jika melihat arsitektur pada Hypervisor, physical machine harus memabgi resource seperti CPU, RAM dan HDD kepada OS untuk menjalankan node dimana saat bersamaan job dilakukan.

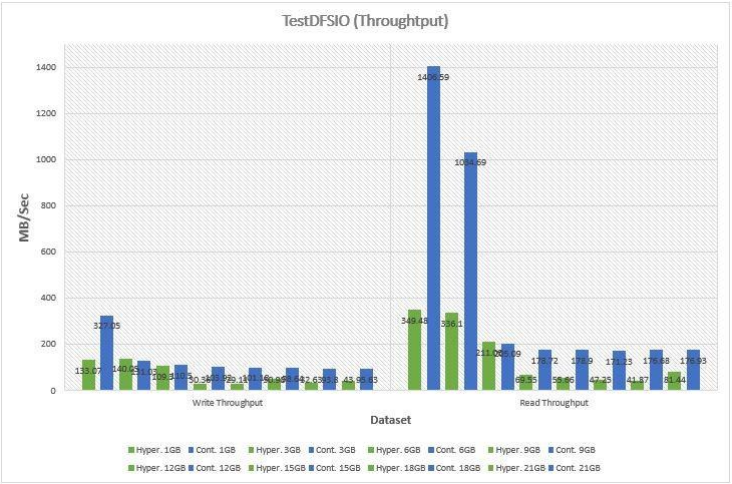
Sedangkan pada gambar 7 pada grafik Container, data 1GB sampai 21GB memperlihatkan bahwa CPU usage rata-rata 66.97%. Jika dilihat lagi pada gambar grafik pada Hypervisor dan dibandingkan dengan gambar grafik pada Hypervisor memperkuat pembuktian bahwa penggunaan resource pada CPU berhubungan langsung dengan waktu penyelesaian job, yaitu jika penggunaan resource maksimal maka waktu penyelesaian job akan semakin cepat, begitu pula sebaliknya.



Gambar 7. CPU Utilization Menggunakan tools TOP pada Container dan Hypervisor

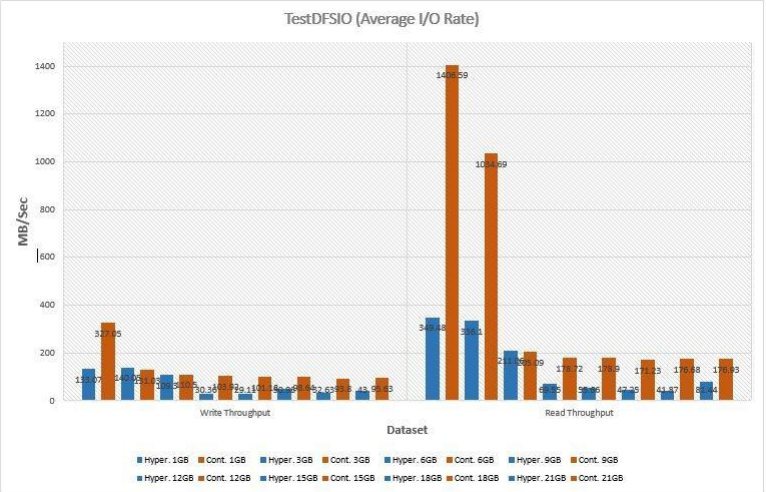
4.2 Disk I/O

Dari hasil pengujian TestDFSIO terhadap Hypervisor dan Container, dapat dilihat dataset 1GB pada write throughput menunjukkan nilai 133.07 mb/sec pada Hypervisor dan 327.05 mb/sec untuk Container yang memperlihatkan perbedaan cukup jelas pada grafik pada gambar 8. Khusus Container terlihat dataset 3GB sampai 21GB pada write throughput tidak menunjukkan interfal nilai hasil yang begitu signifikan, sedangkan Hypervisor mengalami-'nya' pada dataset 9GB sampai 21GB, tetapi kenaikan terjadi dari dataset 18GB ke 21GB yaitu 32.63 mb/sec ke 43 mb/sec. Dikarenakan manajemen harddisk terikat secara langsung dengan waktu penyelesaian eksekusi beban kerja.



Gambar 8. Hasil TestDFSIO (Throughput)

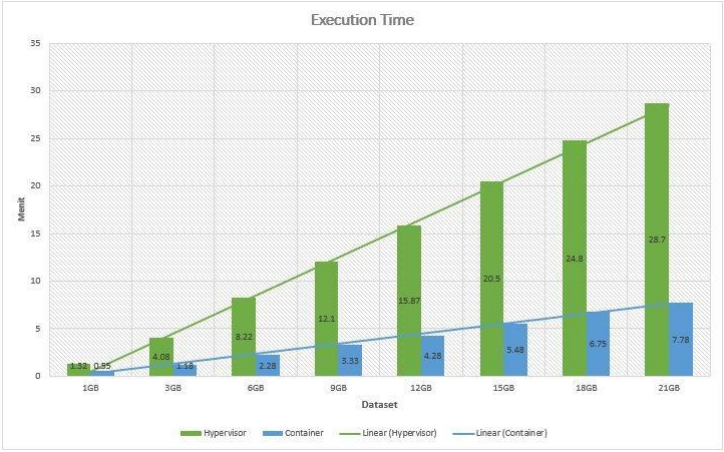
Pada gambar 9 menunjukkan hasil yang hampir sama dengan hasil throughput. Dimana perbedaan hanya dapat dilihat jika nilai ditampilkan sembilan 0 dibelakang koma, sebagai contoh data 'raw' 1GB hasil testDFSIO yaitu untuk throughput 327.052060044 mb/sec sedangkan untuk avg I/O rate yaitu 327.052062988 mb/sec. Dimana I/O rate itu sendiri merupakan kecepatan writing dan reading tempat penyimpanan data.



Gambar 9. Hasil TestDFSIO Average I/O Rate

4.3 Execution Time

Berdasarkan hasil pada pengujian execution time dengan melakukan wordcount job dimana dataset yang digunakan merupakan hasil dari men-crawling data dari twitter. Pada gambar 10 dapat dilihat execution time Container lebih rendah dibandingkan Hypervisor dari percobaan menggunakan data 1GB sampai 21GB, dimana pada data 21GB Hypevisor hampir mencapai 30 menit yaitu 28.7 menit sedangkan Container tidak melebihi 10 Menit yaitu 7.78 menit. Jika melihat dua pengujian sebelumnya, hal itu berdampak langsung pada cepat tidaknya pengeksekusian job diselesaikan.



Gambar 10. Hasil WordCount Job

5. Kesimpulan

5.1 Kesimpulan

Hasil analisis dari seluruh pengujian yang dilakukan dalam penelitian tugas akhir ini dapat menarik kesimpulan sebagai berikut:

1. Dengan penggunaan resource atau penggunaan CPU, Container memperoleh kurang lebih 6.9 kali lebih tinggi dibanding hypervisor. Dimana rata-rata CPU Utilization pada Container 66.97%, sedangkan Hypervisor hanya mendapat 9.57%, membuktikan bahwa Container menggunakan resource secara maksimal.
2. Mengukur Disk I/O menggunakan benchmark TestDFSIO memperlihatkan bahwa management pada harddisk terpengaruh secara langsung dengan waktu penyelesaian eksekusi beban kerja. Dimana setelah membandingkan hasil dari kedua virtualisasi, Container(Docker) menunjukkan throughput dan avg I/O rate yang

lebih baik dibanding Hypervisor(VMWare) dengan nilai rata-rata 131.71 mb/sec untuk write throughput dan 441.10 mb/sec untuk read throughput. Membuktikan bahwa Container(Docker) memiliki management harddisk yang lebih baik dibandingkan Hypervisor(VMWare).

3. Execution time antara Container(Docker) dengan Hypervisor(VmWare) pada penelitian ini menunjukkan bahwa Container(Docker) memiliki waktu yang lebih cepat dalam mengeksekusi wordcount job dengan melihat rata-rata execution pada Container(Docker) diselesaikan dalam waktu 3.95 menit sedangkan Hypervisor(VmWare) membutuhkan 14.44 Menit, dimana Container 2.48 kali lebih cepat di bandingkan Hypervisor. Makin cepat penyelesaian job makin baik.
4. Dari semua pengujian yang dilakukan, berpengaruhnya CPU Utilization dalam Execution time saat menjalankan job dan baiknya manajemen harddisk meningkatkan performa virtualisasi.

5.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian berikutnya adalah:

1. Menggunakan skenario lain seperti arsitektur container orchestration.
2. Menggunakan file system lain.

Daftar Pustaka

[1] Jonathan Stuart Ward and Adam Barker. Undefined By Data: A Survey of Big Data Definitions. 2013.

[2] Afifi-Sabet Keumars. What Is Big Data Analytics? Locowise Blog, pages 1–3, 2018.

[3] Courtney Webster. Hadoop Virtualization: VMware, Inc. 2015.

[4] Kalpana Dwivedi and Sanjay Kumar Dubey. Taxonomy and comparison of Hadoop distributed file system with Cassandra file system. ARPN J. Eng. Appl. Sci., 10(16):6870–6876, 2015.

[5] U Z Edosio. Big Data Paradigm- Analysis , Application , and Challenges Big Data Paradigm- Analysis , Application , and Challenges. (April), 2014.

[6] Zheng Li, Maria Kihl, Qinghua Lu, and Jens A. Andersson. Performance overhead comparison between hypervisor and container based virtualization. Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA, pages 955–962, 2017.

[7] Pedro Roger Magalhães Vasconcelos and Gisele Azevedo De Araújo Freitas. Performance analysis of Hadoop MapReduce on an OpenNebula cloud with KVM and OpenVZ virtualizations. 2014 9th Int. Conf. Internet Technol. Secur. Trans. ICITST 2014, pages 471–476, 2014.

[8] Avanish Singh, P Gouthaman, Shivankit Bagla, and Abhishek Dey. Comparitive Study of Hadoop over Containers and Hadoop Over Virtual Machine. 13(6):4373–4378, 2018.

[9] Apache Hadoop YARN, 2018.

[10] Rizki Rizki, Andrian Rakhmatsyah, and M. Arief Nugroho. Performance analysis of container-based hadoop cluster: OpenVZ and LXC. 2016 4th Int. Conf. Inf. Commun. Technol. ICoICT 2016, 4(c):4–7, 2016.

[11] Sidik Prabowo. Implementasi Algoritma Penjadwalan untuk pengelolaan Big Data dengan Hadoop. Indones. J. Comput., 2(2):119, 2017.

[12] Miguel G. Xavier, Marcelo V. Neves, Fabio D. Rossi, Tiago C. Ferreto, Timoteo Lange, and Cesar A.F. De Rose. Performance evaluation of container-based virtualization for high performance computing environments. Proc. 2013 21st Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. PDP 2013, pages 233–240, 2013.

[13] Roberto Morabito, Jimmy Kjällman, and Miika Komu. Hypervisors vs. lightweight virtualization: A performance comparison. Proc. - 2015 IEEE Int. Conf. Cloud Eng. IC2E 2015, pages 386–393, 2015.

[14] Jingxian Xu, Jianhong Guo, and Chunlan Ren. Implementation and performance test of cloud platform based on Hadoop. IOP Conf. Ser. Earth Environ. Sci., 108(5), 2018.