Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 6, No. 1, February 2021, Pp. 77-82

Design and implementation LETS (Low Power Cluster Server) for sustaining SME during pandemic

Hanugra Aulia Sidharta*[1], Sidharta[2], Felix Filipi[3], Rafli Attala Jaskandi[4], Febrian Nugroho[5]
Computer Science Departement, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia[1,2,3,4,5]

Article Info
Keywords:
Cluster Raspberry Pi, K3S, Orchestration

Article history:
Received: November 30, 2020
Accepted: December 29, 2020
Published: February 28, 2021

Cite:
Sidharta, H. A., Sidharta, Filipi, F., Jaskandi, R. A., & Nugroho, F. (2021). Design and Implementation LETS (Low Power Cluster Server) for Sustaining UMKM during Pandemic. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(1). https://doi.org/10.22219/kinetik.v6i1.1162

*Corresponding author.
Hanugra Aulia Sidharta
E-mail address:
hanugra.sidharta@binus.ac.id

Abstract
Covid pandemic is spreading since end 2019 arround globe, this resulting several countries suffer economic crisis. Due people movement restriction and international border is closed, impact halting on export and import. Indonesia with 267,7 million people manage to overcome global crisis during 1998 and 2008 with encourage small and local industry. According data from BPS, SME support 60% GDP with Rp 7,7 Billion and provide 85 million employment. To sustain SME during customer shifting behavior, we propose small cluster server called as LETS. LETS is modular, and composed with several Raspberry Pi 4, also consume few energies compared with traditional server. Although Raspberry Pi 4 is small size with few limitations, with lightweight application (ansible, k3s, and nginx) it can provide web server service. LETS provide handling traffic up to 742 simultaneus requests, which sufficient of SME requirement.

1. Introduction

Currently pandemic already spread on all around the globe, which started spread since early 2020 from Wuhan as ground zero. This accelerated by many modern transportations, such as: airplane, train, and bus. Public transportation is one of reason this virus is spread globally, this virus easily contagious from person to person. According to worldometer data, on November 2020 there is 54 million cases with death 1,3 million and recovered 38,140 million people [1]. This pandemic caused by corona virus, named by its structure similar with outer ring of sun that have crown. Covid-19 is a unique virus, with crown structure and surrounded with lipid layer, which make difficult detect with human antibody [2]. Since mid year 2020, World Health Organization (WHO) classified this virus as airborne that cause massive lock down on major cities on every country [3]. Every local government restrict people movement, this aim to reduce number of transmissions. To suppress virus spreading, since early this year almost all country closed its border, include airport and port, which made all international transaction is hold. Many country that reliance on tourism, export import, global trading have most impact, with economic slowdown up to 7% and income oer capita is shrink to 3,6 % [4].Due this reason, almost on all country have negative impact on economic scale. World Bank predict global economic will slow down by 5,2% in 2020, this would be a deepest recession since World War two.

Although Indonesian president declare that Indonesia will not suffer due to global recession, on 3rd quarter 2020 Sri Mulyani as Minister of Finance declare that recession is unavoidable. Indonesian economic will suffer contraction by 2,9% to 1,1%, and 2021 projection is 4,5% from this year 5,5% [5]. To overcome recession impact, government should make a mitigation strategy. Faisal Basri and Enny Sri Hartati as senior economist make recommendation aside from priotizing economic impact, government need make strategy to handling pandemic. Without people faith, it will be difficult to overcome this issue [6][7].

During lockdown on major cities, government restrict people movement and forced to minimize physical contact. This caused traditional market is closed, and many SME (Small Medium Enterprise) or UMKM (Usaha Mikro, Kecil dan Menengah) suffer great loss due many people afraid made transaction. On other side, e-commerce transcation is rise due many costumer shift to online transaction. In this paper we try to design a small but efficient server with low power consumption. Many SME have difficulties to adapt to costumer shifting, especially SME (Usaha Mikro Kecil dan Menengah) in rural area with traditional model business.

According data from BPS (Badan Pusat Statistik), Indonesia economic survive during crisis moneter in 1998 due SME with providing employment for 85 million people. During that era, many company difficulties to maintain business that impact massive lay out. SME is one of economic pillar in Indonesia, each year number of SME increase inline with Indonesia Gross Domestic Bruto (GDB). Since 2012 number of SME increase to 62,98 million business unit, with vast amount of business unit involved 107 million workers. Comparing with large scale business model, SME support 60%

GDP with Rp 7,7 Billion [8]. To support Indonesian economic during pandemic, government must give attention and make special policy for SME.

To support SME during pandemic, we propose build small cluster server on SME to provide digital access. Due customer shifting behavior, SME need to adapt digital marketing ecosystem. To solve these problem SME need to adapt with creating value add to digital ecosystem. We named it as LETS, a cluster server composed several Raspberry Pi 4 with low power consumption for various digital needs. This server equipped with modular architecture, that can adapt to specific needs on each SME, such as: website server for e-commerce, as a startup server platform, Point of Sale server to rebranding SME, and many other functions. LETS equipped with minimal 4 unit of Raspberry Pi 4, with cloud capabilities, High Availability scenario. To make system ease to maintain and deploy, in this research use ansible as config management, Kubernetes as docker container management, and trafeic as system monitoring.

Raspberry pi is a single board computer with small size (just a size of credit card), with lot of potential. The latest version is Raspberry Pi 4 equiped with quad core processor and several RAM option (from 1GB through 8GB). With its small size but with load of computing capabilities, there is many projects utilizing it. There is tons of DIY tutorial for home scale project through major company like Oracle build supercomputer with 1060 unit Raspberry Pi [9]. Idea lies beneath it is making a powerful HPC (High Power Computing) for specific needs. Such as for parallel computing [10][11], as a cloud server [12][13], and make a small but powerful computing on biomedic [14][15]. Build Raspbery Pi cluster is cheaper option rather than build traditional server, with low power consumption. On each Raspbery Pi 3 model B+ is consume around 6,95 Watt during idle, and 7,61 Watt during top peak usage [16]. With LETS prototype we hope SME can adapt through customer shifting behavior, and sustain with reachable independent resources.

2. Research Method

In this research we propose network topology as Figure 1, LETS cluster server constructed with 4 unit of Raspberry Pi 4, with several function such as: Node 1 act as node and a master, master node act to distribute command and load to and from another node, load balancing traffic and CPU load on each node. There is 3 node act slaves, which is node 2, node 3 and node 4. But there is differentiation for Node 4, where this node act as slave and database storage. We choose this method due Node 4 is equipped with SSD storage, meanwhile another node just equipped only SD card. SSD used to provide vast amount of DB storage but on this research, we didn't use RAID mode. This option appears due Raspbery Pi 4 already equipped with 2 port USB 3.0 that theoretically support transfer rate up to 620 MBps.

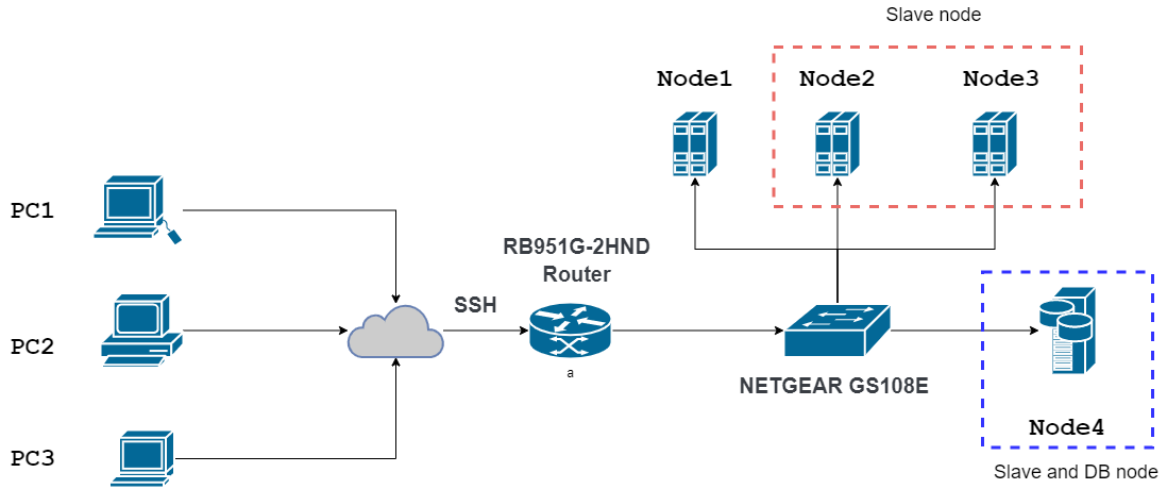


Figure 1. LETS Network Topology

Router RB951 is used to create policy to limit inbound and outbound traffic, only master node can reach outsite network and another node is prohibited by any user or administrator directly. Those strategy is made to provide protection on all node, if there are security issue such as DDOS attack that only make 1 casualities, which is node number one. Router RB also act as port filtering and forwarding, RB only open common port regarding service offered by server. In this research we configure open port related are: HTTP, HTTPS and SSH services only, HTTP is for provide website service, and SSH is used to accessing master node. With this criteria, user and administrator can access through master node using SSH. SSH is old technology yet effective to accessing device remotely and provide enough security.

Ansible is an open sources program run at Linux environtment, developed by Michael DeHand for Fedora Unified Network Controler. On 2015 Ansible.Inc acquired by Red Hat, since then ansible available on all Fedora distribution,

such as: Red Hat Enterprise Linux, CentOS, openSUSE, Debian, Ubuntu, and Oracle Linux [17]. Ansible aim to develop and manage multiple devices or server, provide ease to maintain server on same network or on different network using SSH. Many developer and system manage servers by logging onto each each server via SSH, making adjustment or done several configs, dan log off after their work done. Followed with documenting action already done on those servers, and then move to another server. If system administrator needs to grab data from bunch of servers, they need to manually logging on each server and create log file. This action is exhausting and sometimes leads to human erorr due repetitive manual labor.

Ansible is a nimble software, can run on every machine remotely using SSH using multiple platforms as long there is shell command console. Ansible provide automation tools as a conductor on orchestra, one conductor can manage multiple person with different music tools just using hand movement [18]. Ansible use YAML (Yet Another Markup Language), YAML is simple due easly read by human, and computer ease to understand due its machine-perseable syntax. As Figure 2 below, user interact with server through YAML using SSH. Meanwhile Playbook provide organizing task, including define host, variables files been used, task to each node. Using ansible, system administrator can reach and manage n number of hosts [18][19].

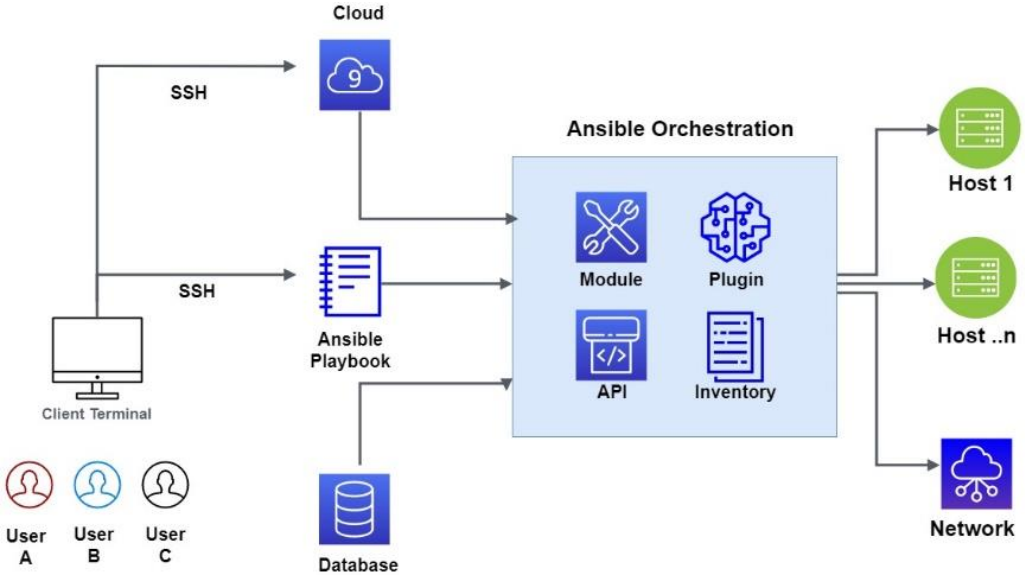


Figure 2. Ansible Architecture on LETS

Raspberry Pi 4 with magnificent hardware specification (compare with its size), it still has many disanvantages. Raspberry Pi 4 is equipped with ARM based processor, which limit number of operating system and software that can be used. Although we can use many linux distro on Raspberry Pi 4, not all software and tools ported to arm architecture, one of them is Kubernetes. Currently Kubernetes is rising star due its feature and rapid grows (either end user or enterprise). To answer Kubernetes compability issue on Raspberry Pi, Rancher LAB created K3S, a lightweight version from Kubernetes that run smooth on arm platform. K3s is simplified and miniature version from Kubernetes, however K3S equipped with API and functionality. Almost all tools and ecosystem that work on Kubernetes also work on K3S.

K3S packages is design with simplicity yet powerful, deploy with small size (~100MB) and ease to running and deploy. Although its small size, K3S capable to run all Kubernetes component, such as: API server, scheduler, and controller. Regarding Figure 3, in K3S cluster environment, server is a node equipped with kubelet and control plane, while agent is a node that just run kubelet. Both server and agent have running container and a kubeproxy, kubeproxy used to manage network traffic and tunneling between cluster [20].

3. Results and Discussion

On this research we used four Raspberry Pi 4 model 2019, one Raspberry Pi with largest memory (8GB) is act as a master with assumption master node will consume more memory than slave. Folllowed three Raspbery Pi 4 with lower memory (4GB) as slave node. On each Raspberry Pi equipped with class 10 micro SD card with 32 GB capacity, and all node using arm ubuntu server only CLI support to minimize memory usage to unnecessary program. The fourth node connected with SSD as database server, due SSD transfer rate is almost 2 times faster than micro SD card All node connected with switch that support Gigabit ethernet using Cat5e UTP cable, to minimize bottleneck issue on network connection. Mikrotik router used to manage traffic inbound and outbound stub, to limit direct connection through slave.

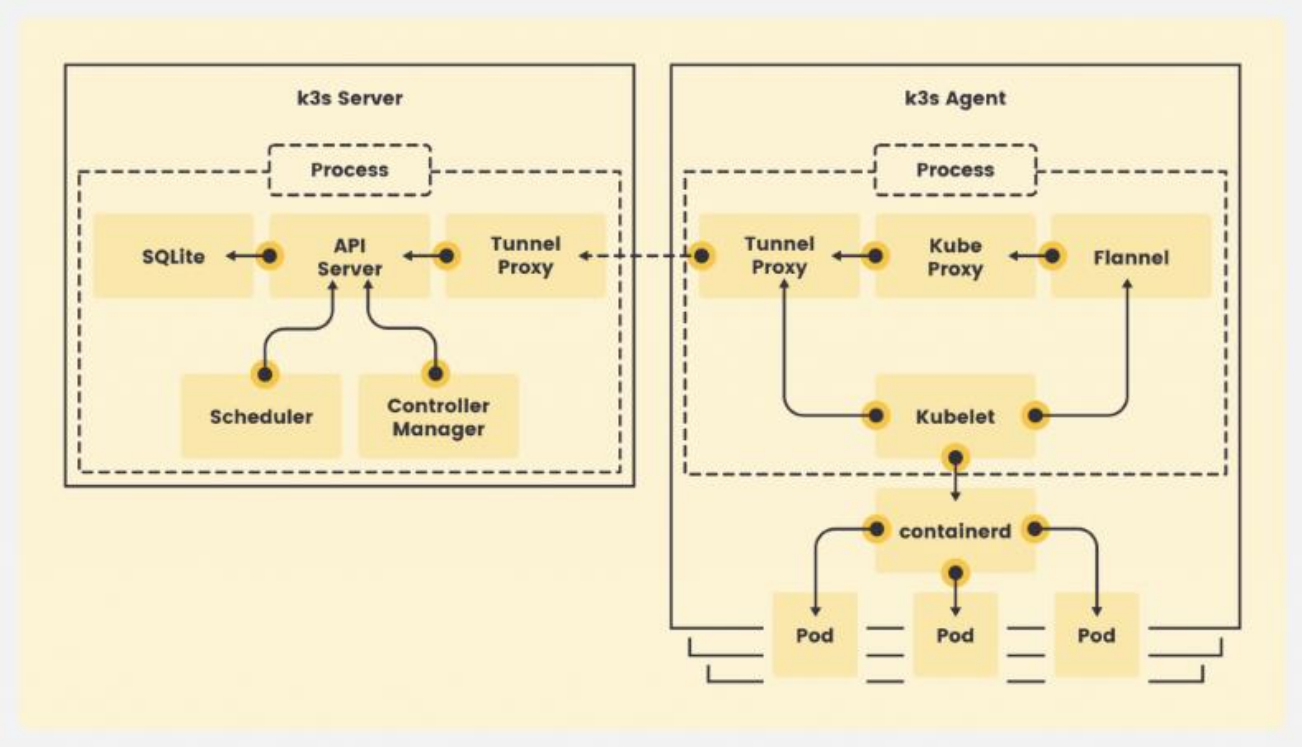


Figure 3. K3S Architecture on Raspberry Pi 4

3.1 Overheat and bandwith Issue

In this research, we used first model of Raspberry Pi 4 that released in 2019. First impression using using Raspberry Pi 4 is overly heat while been used, even though on idle condition. To overcome this issue, we conduct trial with several operating system, such as: Ubuntu MATE for Raspberry Pi, Raspberry OS (Previously named as Raspbian OS) and Ubuntu server. Based our observation MATE desktop is consuming most memory and caused excessive heat, this suspect because Raspberry Pi work harder to render MATE environment. The lightest operating system is Raspberry OS followed with Ubuntu Server; this is our reason choose Ubuntu server.

We can examine heatmap from thermal imaging camera on Figure 4, most heat source came from CPU (right picture number 3), new power chipset to handle USB-C power supply (right picture number 1). There is two opinion regarding overheat issue, first is larger power supply using USB-C with specification capacity 3.0 A. Second, is related with large and stable power supply is needed due Raspberry Pi 4 equipped with large memory and there is 2 micro HDMI with 4K support output [21]. Also, with new two USB3 make Raspberry Pi 4 become power voracious and lead to overheating. This issue broadens with operating system policy, when CPU hitting overheat threshold (85 ℃) system will throthled down CPU with limiting power current to CPU. Apart from CPU, throttled down also effect on GPU which result degradation of rendering quality.

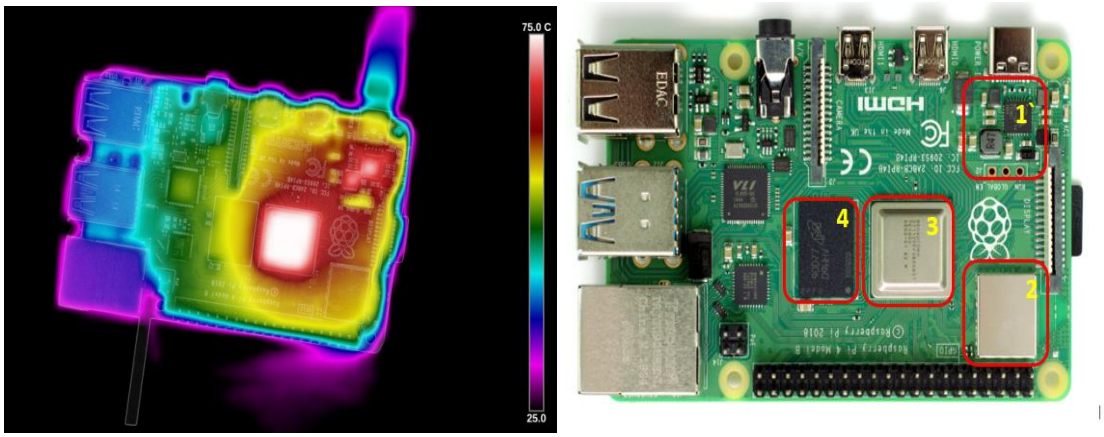
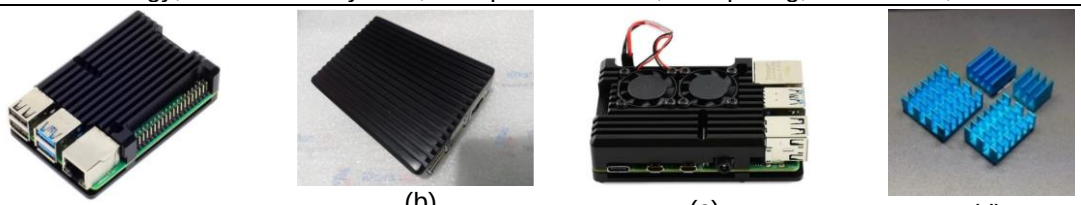


Figure 4. Thermal Imaging on Raspberry Pi 4 with Heavy Load [22]



(a) (b) (c) (d)

Figure 5. Active and Pasive Cooling used for Raspberry Pi 4

As we can see on Figure 5, to overcome overheat issue we conduct experiment using two type of heatsink, 3 model passives heatsink (model a, b, and d), and active cooling (model c). Type a, b and c heatsink made of CNC allumunium alloy, meanwhile type c is aluminium. Heatsink type a, b, and c covered Raspberry Pi 4 from upside and bottom side, to overcome heat problem on back side, meanwhile heatsink type d covered CPU, memory modul, and WLAN modul.

Table 1 resume our experiment regarding overheat, all passive heatsinks succeed reduce temperature by 10°C, and active heatsink by 12 – 13 °C. From data in Table 1, we can conclude most effective cooling is use type c, with dual fan 5v using internal Raspberry Pi power supply (Using pin 2 and 3), but this option is most noisy due high RPM combined with small fan size (around 5 cm). Most subtle solution is type c, with lowest price can gain effect as expensive one. During our experiment we found that passive cooling type b have poor impact on network communication. Raspberry Pi 4 is equipped with dual band wireless connection, but when type b implemented cause reduction on wireless network reception, that impact low throughput and unstable connection. We recommend not using passive cooling type b, unless network connection is using cable connection.

Table 1. Temperature Comparison on Raspberry Pi 4

Model	Idle (°C)	Average load (°C)	Blocking wifi reception
With out heatsink	54	65	No
a	44	54	Yes
b	45	55	Yes
c	43	51	Yes
d	45	54	No

3.2 Web Server Configuration

LETS is a cluster server contain four Raspberry Pi 4, hardware configuration can see on Figure 1. After all node installed ubuntu server arm, then all node configured using ansible. Ansible was chosen to distribute software requirement and version due ease to manage on cluster environment, config done with yaml file. While working on cluster node, we focus to make high availability server. Master nodes have task to balancing traffic to all agent node by creating node pool of several workers. All node that act as agent node is registered through websocket initated by k3s agent, then master is in charge to maintain client site load balancer [23].

To provide webserver services, in this research we used NGINX. Nginx is a open source software to provide several service, such as: web server, reverse proxy, cache, load balancing, media streaming [24]. While Raspberry Pi 4 is superior to it predecessor, it still single board computer with limited computation power. Nginx impeccable answer those issue, with lightweight support and consume only small computing resources also able to handle large number of connections with low memory usage.

In addition, as webserver nginx also perform as HTTP load balancer, to handle and balance multiple application instances. HTTP load balancing intent to optimize resource utilization, maximize traffic throughput, reduce latency, and fault tolerant configuration [25]. Nginx support several load balancing algorithms, such as round robin, least connection, and IP hash. On this experiment we select round robin algorithm, all incoming request distributed evenly on all server with server weight consideration.

To verify cluster performance, we done benchmark cluster using simultaneous request from several computer, this aim to simulate real life user request. Simulation done to master node (192.168.100.31:30080) by GET HTTP method request from 4 laptop. On each laptop equipped with test script utilize concurrency method, with objective abuse server with 250 thread. With that scenario will load server with 1000 thread simultaneously on one time, test done for one hour to get various data. All data is log on master node, due master node act as load balancer onto another node. Benchmark result is summarized on Table 2, while on our simulation make 1000 request simultaneous, request handled average is 481 request, and maximal 742 request. From all incoming request, Raspberry Pi cluster can handle average 451 request with average 32 warning label. Latency measured average is 3120,25 ms, with minimal 38,16 ms and

maximal 96581 ms, this suspect occur due high traffic that make server takes time to respond request. Traffic measured by server is max 7.485 bytes with average 7.102 bytes.

4. Conclusion

Covid pandemic that spread during 2020 have been conduce crysis on many aspects, include economy. According data from Minister of Finance predict Indonesian economic will suffer contraction from 2,9% to 1,1%. SME as low and middle business model that proven supporting Indonesian economic during global crysis in 1998. SME is proven agile and effective absorb manpower, government must observe and care with SME. To support SME self sustain during pandemic, we propose small cluster with low cost server. This server aims to provide web service server to improve customer engagement. LETS able to provide website service and could handle traffic up to 742 simultaneous requests but need to improve on latency issue. We suggest optimizing Raspberry pi cluster to improve latency and add more capacity that can handle by cluster.

Table 2. Benchmark Result

Remark	Min	Avg	Max
Req generate	204	481	742
Req success handled	185	452	742
Req with warning	1	32	69
Latency (ms)	48,16	3120,25	96581
Bytes send	6.564	7.102	7.485

Acknowledgement

Thanks for Kementerian Riset dan Teknologi Republik Indonesia for grand research PDP (Penelitian Dosen Pemula). This research is funding from hibah PDP (*Penelitian Dosen Pemula*) single year 2020 with contract number: 068/SP2H/LT-MONO/LL7/2020.

References

[1] "Coronavirus Update (Live): 54,818,483 Cases and 1,324,530 Deaths from COVID-19 Virus Pandemic - Worldometer."
[2] G. Jain, D. Mittal, D. Thakur, and M. K. Mittal, "A deep learning approach to detect Covid-19 coronavirus with X-Ray images," *Biocybern. Biomed. Eng.*, vol. 40, no. 4, pp. 1391–1405, 2020. https://doi.org/10.1016/j.bbe.2020.08.008
[3] W. H. O. Health *et al.*, "Transmission of SARS-CoV-2 : implications for infection prevention precautions," no. July, pp. 1–10, 2020, Accessed: Nov. 16, 2020.
[4] The World Bank, "COVID-19 to Plunge Global Economy into Worst Recession since World War II," *The World Bank*, 2020.
[5] T. Pangastuti, "Indonesia Is Unlikely to Escape Recession: Finance Minister."
[6] A. W. Akhlas, "Indonesia at 'near-certain recession' as recovery depends on virus control, economists warn - Business - The Jakarta Post."
[7] N. Ibrahim and Verliyantina, "The Model of Crowdfunding to Support Small and Micro Businesses in Indonesia Through a Web-based Platform," *Procedia Econ. Financ.*, vol. 4, no. Icsmed, pp. 390–397, 2012. https://doi.org/10.1016/S2212-5671(12)00353-X
[8] Nur Islami, "Bukalapak dan Kemkominfo Gelar SME Go Online," 2017.
[9] G. Venzl, "Building the world's largest Raspberry Pi cluster." 2020.
[10] J. Kiepert, "Creating a Raspberry Pi-Based Beowulf Cluster," *Boise State Univ.*, pp. 16, 2013.
[11] I. C. Donca, C. Corches, O. Stan, and L. Miclea, "Autoscaled RabbitMQ Kubernetes Cluster on single-board computers," *2020 22nd IEEE Int. Conf. Autom. Qual. Testing, Robot. - THETA, AQTR 2020 - Proc.*, 2020. https://doi.org/10.1109/AQTR49680.2020.9129886
[12] P. Abrahamsson *et al.*, "Affordable and energy-efficient cloud computing clusters: The Bolzano Raspberry Pi cloud cluster experiment," *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 2, pp. 170–175, 2013. https://doi.org/10.1109/CloudCom.2013.121
[13] S. Dokic *et al.*, "Angular and Jenkins Based System for Remote Execution of Tasks on Raspberry Pi Cluster," *2020 19th Int. Symp. INFOTEH-JAHORINA, INFOTEH 2020 - Proc.*, no. March, pp. 18–20, 2020. https://doi.org/10.1109/INFOTEH48170.2020.9066299
[14] F. Habibie, Afiahayati, G. B. Herwanto, S. Hartati, and A. Z. K. Frisky, "A Parallel ClustalW Algorithm on Multi-Raspberry Pis for Multiple Sequence Alignment," *Proc. - 2018 1st Int. Conf. Bioinformatics, Biotechnol. Biomed. Eng. BioMIC 2018*, 2019. https://doi.org/10.1109/BIOMIC.2018.8610612
[15] J. Singer *et al.*, "Next generation single board clusters," *IEEE/IFIP Netw. Oper. Manag. Symp. Cogn. Manag. a Cyber World, NOMS 2018*, pp. 1–3, 2018. https://doi.org/10.1109/NOMS.2018.8406120
[16] P. J. Basford *et al.*, "Performance analysis of single board computer clusters," *Futur. Gener. Comput. Syst.*, vol. 102, pp. 278–291, 2020. https://doi.org/10.1016/j.future.2019.07.040
[17] M. DeHaan, "The Origins of Ansible," 2013.
[18] J. Geerling, *Ansible for DevOps : Server and configuration management for humans*. Midwestern Mac, LLC; 1st edition, 2015.
[19] M. Duffy, *DevOps Automation Cookbook*, 1st editio. Birmingham: Packt Publishing, 2015.
[20] A. MSV, "How Rancher Labs' K3s Makes It Easy to Run Kubernetes at the Edge – The New Stack," 2020.
[21] J. Geerling, "The Raspberry Pi 4 needs a fan, here's why and how you can add one | Jeff Geerling," 2019.
[22] A. Bate, "Thermal testing Raspberry Pi 4 - Raspberry Pi," *2019*.
[23] Rancher, "Rancher Docs: Documentation | Rancher Labs."
[24] V. Kholodkov, *Nginx Essentials*, 1st editio. Birmingham: Packt Publishing, 2015.
[25] "NGINX Docs | HTTP Load Balancing."