

PENINGKATAN AVAILABILITY INFRASTRUKTUR STORAGE PADA CLUSTER COMPUTING MENGGUNAKAN METODE FAULT TOLERANCE

Muhamad Ikmal Wiawan¹⁾, Agung Baitul Hikmah²⁾

¹⁾ Teknik Informatika Universitas Langlangbuana

²⁾ Manajemen Informatika AMIK BSI Tasikmalaya

email : ikmal.wiawan@gmail.com¹⁾, agung.abl@bsi.ac.id²⁾

ABSTRAK

Cluster Computing digunakan di hampir semua komputasi *multi node* seperti *grid computing*, *high performance computing*, *cloud computing*, *hypervisor* dan lainnya. Ada beberapa komponen di dalam *cluster* seperti *computing resource*, *storage resource*, dan *networking resource*. Sebuah sistem *storage cluster* tidak terlepas dari *fault* atau kegagalan sistem. Kegagalan tidak dapat dihindari tapi kita dapat mengurangi frekuensi dan menyediakan ketahanan sistem yang kuat serta mampu mengadaptasi kegagalan tersebut. Pada aspek *security triad* dikenal istilah *confidentiality*, *integrity* dan *availability*. *Availability* adalah aspek *Security* yang menjadi fokus bahasan pada penelitian kali ini. Penelitian ini harus dapat menekan nilai *Mean Time To Repair* (MTTR) dan meningkatkan nilai *Mean Time Between Failure* (MTBF) untuk mendapatkan nilai *availability* yang maksimal.

Fault tolerance menawarkan *redundancy* sebagai solusi peningkatan *robust* pada *system* yaitu *replication* dan *erasure coding*. Pada penelitian kali ini akan menghitung komposisi Teknik *replication* dan atau teknik *erasure coding* pada bagian *storage*. Selain itu juga akan menghitung waktu *recovery* yang dibutuhkan *system* atau *software*, atau protokol yang ada terhadap *block*, *file*, dan *objek storage* untuk mendapatkan nilai Kuantitas MTTR terkecil. Selain mempertimbangkan nilai *availability* terhadap sebuah sistem, yang perlu diperhatikan adalah kualitas skema *clustering* yang diuji. Eksperimen yang dilakukan ini diwakili dalam bentuk model supaya bisa diperuntukan untuk penggunaan *cluster* secara umum dan mendekati implementasi aslinya. diharapkan bahwa peningkatan *availability* dengan mengefektifkan *fault tolerance* dapat menjadi solusi permasalahan ketersediaan data pada infrastruktur cluster secara umum. Hasil akhir dari penelitian ini adalah memberikan usulan skema konfigurasi *Clustering* dalam bentuk model *storage cluster* yang *robust*.

Kata Kunci : *Cluster Computing*, *Storage*, *fault tolerance*, *Availability*, *MTTR*, *MTBF*

ABSTRACT

Cluster Computing is used in almost all *multi-node computing* such as *grid computing*, *high performance computing*, *cloud computing*, *hypervisor* and more. There are several components in the *cluster* such as *resource computing*, *resource storage*, and *networking resources*. A *storage cluster* system is inseparable from *fault* or *system failure*. *Failure* can not be avoided but we can reduce the frequency and provide strong system resilience and be able to adapt the failure. In terms of *security triad* known the term *confidentiality*, *integrity* and *availability*. *Availability* is the *Security* aspect which is the focus of discussion in this research. This research should be able to suppress *Mean Time To Repair* (MTTR) value and increase *Mean Time Between Failure* (MTBF) to get maximum *availability* value.

Fault tolerance offers *redundancy* as a *robust enhancement solution* for *replication* and *erasure coding*. In this research will calculate the composition of *replication techniques* and *erasure coding techniques* on the *storage*. It will also calculate the *recovery time* required by the *system* or *software*, or the existing protocol against *block*, *file*, and *storage object* to get the smallest *MTTR Quantity* value. In addition to considering the value of *availability* to a *system*, which needs to be considered is the quality of *clustering scheme* tested. These experiments are represented in the form of models to be used for the use of *clusters* in general and close to their original implementation. it is expected that increasing *availability* by streamlining *fault tolerance* can be a solution to the problem of data *availability* on *cluster infrastructure* in general. The final result of this research is to propose the *Clustering configuration scheme* in the form of *robust storage cluster model*.

Keywords : *Cluster Computing*, *Storage*, *fault tolerance*, *Availability*, *MTTR*, *MTBF*

PENDAHULUAN

Cloud computing muncul sebagai salah satu topik hangat di bidang teknologi informasi. *Cloud computing* didasarkan pada beberapa area penelitian komputasi lainnya seperti HPC, virtualization, computing utilities dan grid computing. *Cloud computing* berisikan infrastruktur *cluster computing* di dalamnya sehingga pembahasan *cluster* ini menjadi penting dan menjadi inti dari penelitian. Fungsinya yang sangat luas menyebabkan *cluster computing* bisa digunakan di berbagai lini bisnis secara umum dan penyimpanan data khususnya. Sesuatu yang sangat diandalkan tentunya harus terjamin keamanannya khususnya dalam segi *availability* data. Keamanan data berdasarkan *security* triad terbagi ke dalam *confidentiality*, *integrity* dan *availability*. Aset terpenting dari sebuah infrastruktur IT adalah media penyimpanan data sehingga *availability storage* pada *cluster computing* fokus penelitian

Storage Area Network (SAN) dan Network Attached Storage (NAS) adalah pilihan yang dapat diimplementasikan dalam pembangunan infrastruktur *storage* pada *cluster computing* yang tidak mungkin terlepas dari *system fault*. *System fault* atau kejadian yang sangat mungkin terjadi pada semua system khususnya system cluster. *System fault* tidak dapat dihindari namun dapat diminimalisir atau diadaptasi. Mengurangi probabilitas, mempertahankan *availability system* dan mengadaptasi *fault* adalah sifat dari *fault tolerance*. *Fault tolerance* secara alami sudah melekat pada hampir semua *software*, *protocol* dan *system cluster* namun perlu diaktifkan atau diefektifkan penggunaannya. Peningkatan *availability* harus dapat dihitung menggunakan *availability measurement*. Mean Time Between Failure (MTBF) dan Mean time to Repair (MTTR) adalah komposisi penghitungan *availability*. Metode penelitian ini adalah dengan membuat skema konfigurasi yang mengefektifkan penggunaan replika dan erasure coding untuk meningkatkan nilai MTBF dan mengurangi nilai MTTR

Hasil Akhir dari penelitian ini adalah terciptanya skema konfigurasi infrastruktur storage pada *Cluster Computing* yang memiliki nilai *availability* tinggi dan dapat diimplementasikan di berbagai macam model *Storage*.

a. *Failure Rate*

“Failure rate is the conditional probability that a device will fail per unit of time. The conditional probability is the probability that a device will fail during a certain interval given that it survived at the start of the interval [2]”.

Berdasarkan penjelasan di atas *Failure Rate* (FR) adalah probabilitas kegagalan sebuah komponen atau perangkat dalam rentang waktu. *Failure Rate* dinyatakan dalam rasio dari jumlah kegagalan atau *error* terhadap total waktu operation $\lambda = K/T$. K= jumlah total *failure*, sedangkan T adalah total waktu operation.

b. *Mean Time Between Failures (Mtbf)*

“MTBF (Mean Time between Failures) is the average (expected) time between the two successive failures of a component. It is a basic measure of a system’s reliability and availability and is usually represented as units of hours “, Hoda Rohani, Azad Kamali Roosta [3]

Berdasarkan sumber di atas MTBF adalah rata rata rentang waktu dari dua *error* terhadap sebuah komponen. Biasanya MTBF direpresentasikan dalam satuan jam.

“Which is the average time interval (normally in hours) between two consecutive component failures. Reliability is improved when the time interval spanning separate failures is extended [2]”.

Berdasarkan pemaparan di atas dijelaskan bahwa MTBF adalah rentang waktu antara dua kejadian error. Penambahan waktu interval error meningkatkan *reability* sistem atau komponen. ada dua jenis MTBF yaitu *hardware* MTBF dan *system* MTBF. *Hardware* MTBF adalah MTBF yang dilihat dari komponen komponen sistem sedangkan *system* MTBF adalah akumulasi dari MTBF setiap komponen yang mempengaruhi sistem. Sedangkan dalam penelitian ini nilai MTBF adalah media penyimpanan utama *harddisk* dan Node Server itu sendiri.

c. **MEAN TIME TO REPAIR (MTTR)**

“Mean Time to Repair (or Recover) is the average (expected) time taken to repair a failed module. This time includes the time it takes to detect the defect, the time it takes to bring a repair man onsite, and the

time it takes to physically repair the failed module. Just like MTBF, MTTR is usually stated in units of hours [3]”.

“MTTR is an arithmetic average of how fast the system is repaired and is easier to visualize than the probability value[4]”.

Berdasarkan dua pemaparan di atas dapat disimpulkan bahwa MTTR adalah waktu rata-rata yang dibutuhkan sebuah komponen untuk melakukan *recovery* / upaya perbaikan. Dalam penelitian ini MTTR adalah waktu *recovery storage cluster* pada saat ada node harddisk yang disimulasikan rusak. Satuan MTTR adalah sama dengan MTBF yaitu jam. Berikut ini adalah perhitungan MTTR

$$\text{MTTR} = \frac{\text{TOTAL RECOVERY TIME}}{\text{TOTAL NUMBER OF REPAIR}}$$

d. Konsep Availability

“Availability is the degree to which a system or component is operational and accessible when required for use [10]”

Berdasarkan penjelasan di atas dapat disimpulkan bahwa *Availability* adalah ukuran operasional setiap komponen sistem untuk selalu dapat digunakan. Menurut *Department of Defence of USA* ada dua jenis *availability* yaitu *materiel availability* dan *operational availability*. Berikut pemaparannya.

1. Materiel Availability

“*Materiel Availability* is a measure of the percentage of the total inventory of a system operationally capable (ready for tasking) of performing an assigned mission at a given time, based on materiel condition. This measure can be expressed mathematically as number of operational end items/total population. The *Materiel Availability* addresses the total population of end items planned for operational use, including those temporarily in a non-operational status once placed into service (such as for depot-level maintenance [5]”

Berdasarkan sumber di atas maka *Materiel Availability* adalah ketersediaan komponen atau perlengkapan yang bukan lingkup operasional. Sedangkan dalam penelitian ini tidak menggunakan penghitungan *materiel availability*.

2. Operational Availability

“*Operational Availability*. *Operational Availability* indicates the percentage of time that a system or group of systems within a unit are operationally capable of performing an assigned mission and can be expressed as (uptime/(uptime + downtime)). Determining the optimum value for *Operational Availability* requires a comprehensive analysis of the system and its planned use as identified in the *Concept of Operations (CONOPS)*, including the planned operating environment, operating tempo, reliability alternatives, maintenance approaches, and supply chain solutions[5]”.

Menurut sumber di atas dapat disimpulkan bahwa *Operational Availability* adalah penghitungan dalam bentuk presentasi terhadap ketersediaan sistem di lingkup operasional.

e. Fault Tolerance

“*Fault tolerance system* is a vital issue in distributed computing; it keeps the system in a working condition in subject to failure. The most important point of it is to keep the system functioning even if any of its part goes off or faulty [6].

“*Fault Tolerance is the process of finding faults and failures in a system. If a fault occurs or there is a hardware failure or software failure then also the system should work properly [7]*”.

Berdasarkan beberapa definisi di atas maka *Fault tolerance* adalah kemampuan *system* untuk menangani kegagalan baik berupa *hardware* maupun *software* dengan cara mengadaptasi kegagalan (*adaptive*) atau mempertahankan ketahanan sistem (*robust*). Ada dua metode yang sangat umum digunakan yaitu *replication* dan *redundancy*.

1. *Replikasi*
replikasi adalah metode yang digunakan di dalam *fault tolerance* untuk menjaga ketersediaan data dengan cara membuat replika atau *copy* ke tempat atau komponen lain
2. *Redundancy*
redundancy adalah metode yang digunakan di dalam *fault tolerance* untuk menjaga ketersediaan data atau komponen lain di dalam sistem dengan cara memberikan kelebihan komponen tersebut. *Redundancy* dalam *fault tolerance* diwujudkan dalam bentuk *erasure coding*
3. *Erasure Coding (EC)*
Erasure Coding (EC) adalah metode perlindungan data di mana data dipecah menjadi fragmen, diperluas dan dikodekan dengan potongan data redundan dan disimpan di satu set berbagai lokasi atau media penyimpanan. Dalam penelitian ini penggunaan EC akan melibatkan RAID

f. Parameter *Fault Tolerance*

Parameter *fault tolerance* dibutuhkan supaya menjadi bahan pertimbangan dalam implementasinya. Berikut ini adalah parameter *fault tolerance* [9]

1. *Throughput*
Mendefinisikan lebar data secara aktual dari performa *storage* yang dibangun. *Throughput* dari sebuah sistem harus tinggi.
2. *Response Time*
Waktu yang diperlukan sistem untuk menyelesaikan sebuah transaksi.
3. *Scalability*
Adalah kemampuan sistem untuk tetap menjaga kemungkinan penambahan kapasitas penyimpanan seiring dengan pertumbuhan data walau sedang dalam keadaan *fault*.
4. *Availability*
Adalah sistem tetap berjalan walaupun dalam keadaan *error*.
5. *Useability*
Sistem harus dapat diakses dengan efektif, efisien dan memuaskan. Jangan sampai teknik *fault tolerance* yang ditetapkan malah membuat fungsionalitas *storage* menjadi terganggu.
6. *Realiability*
Dalam kondisi apapun fungsionalitas sistem harus tetap berjalan dengan baik sebagaimana mestinya.
7. *Cost Effectiveness*
Efisiensi harga untuk pembangunan sistem harus murah, salah satu solusinya adalah dengan menggunakan perangkat lunak *opensource*.

g. Konsep keamanan Informasi

Secara umum hampir di semua pembahasan *information security* selalau terfokus pada *confidentiality*, *integrity* dan *availability*.

1. *Confidentiality*

“*Confidentiality: This term covers two related concepts: Data confidentiality and privacy. Data confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized individuals. Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.[1]*”

berdasarkan definisi di atas dapat disimpulkan bahwa *confidentiality* adalah upaya menjaga kerahasiaan data kepemilikan akses hanya bagi pemilik datanya.

2. Integrity

“ *This term covers two related concept. Data integrity: Assures that information and programs are changed only in a specified and authorized manner. System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system* [1].

Menurut definisi di atas dapat disimpulkan bahwa *integrity* adalah upaya untuk menjaga keutuhan dan keaslian data dari awal dikirim hingga sampai ke penerima. Selain itu *integrity* juga memungkinkan *non repudiation* berarti tidak ada sangkalan terhadap kebenaran data.

3. Availability

“*Assures that systems work promptly and service is not denied to authorized users*”, [1]

“*Availability is the guarantee that information will be available to the consumer in a timely and uninterrupted manner when it is needed regardless of location of the user. This means that the cloud infrastructure, the security controls, and the networks connecting the clients and the cloud infrastructure should always be functioning correctly. Availability is ensured by: fault tolerance, authentication and network security*[8]” Ashish Agarwal, Aparna Agarwal (2011:1)

Berdasarkan pemaparan dua sumber di atas dapat disimpulkan bahwa layanan atau produk sistem dapat diakses tanpa penghentian atau gangguan terhadap sistem itu sendiri.

METODE PENELITIAN

Penelitian tentang “Peningkatan Availability Infrastruktur Storage Pada *Cluster Computing* Menggunakan Metode Fault Tolerance”, perlu diselesaikan dengan sistematis dengan menggunakan metode penelitian yang dilakukan ini menggunakan metode Eksperimen dengan Analisa secara kualitatif dan kuantitatif. Pada tahapan penelitian yang dijelaskan di bagian selanjutnya bahwa metode kuantitatif digunakan untuk penghitungan nilai *availability* sedangkan metode kualitatif digunakan untuk memberikan legitimasi atau argumen terhadap nilai *availability* yang telah dihitung dengan metode kuantitatif. Secara garis besar tahapan penelitian ini terdiri dari studi literatur, penentuan status awal, *assessment*, dan eksperimen peningkatan *availability*. Berikut ini adalah gambaran umum tahapan penelitian.

a. Definisi status awal

Definisi status awal adalah kondisi awal sebuah sistem *storage cluster* sebelum dilakukannya peningkatan *availability*. Pada langkah ini yang perlu dilakukan adalah penentuan spesifikasi *storage cluster* yang akan dijadikan status awal. Ada tiga jenis model spesifikasi cluster yang akan digunakan pada tahapan penelitian

1. Spesifikasi *Cluster* Fisik

Spesifikasi fisik dibutuhkan untuk menentukan penghitungan *throughput*, *recovery time*, *availability* dan lainnya secara fisik yang tidak bisa dilakukan secara virtual yang akan menjadi bahan perhitungan model

2. Spesifikasi *Cluster* Virtual

Spesifikasi *cluster virtual* dibutuhkan untuk menentukan penghitungan *throughput*, *bandwidth*, *recovery time*, *availability* dan lainnya secara *virtual*.

b. *Assesment*

Assesment adalah tahapan untuk melakukan implementasi dan penilaian. Assesment dilakukan pada awal kondisi awal yang telah didefinisikan dan setelah adanya upaya peningkatan *availability*. Pada tahapan ini akan menghasilkan nilai *availability* secara kuantitatif dan juga hasil analisis pada kondisi awal dan setelah upaya peningkatan. Ada beberapa sub tahapan pada tahapan assesment ini.

a) *Storage Level*

Mengimplementasi dan menguji skema konfigurasi *fault tolerance* kepada ketiga level *storage* di bawah ini.

1. *Block Level Storage*

Implementasi *storage* di level *block*

2. *File Level Storage*

Implementasi *storage* di level *file*

3. *Object level Storage*

Implementasi *storage* di level *object*

b) Pengujian *fault tolerance* secara kuantitatif

Pengujian secara kuantitatif dilakukan untuk mengetahui nilai peningkatan *availability* dari masing masing skema

1. Nilai *Mean Time To Repair (MTTR)*

Penghitungan nilai MTTR dalam bentuk *cluster* baik di *state* awal maupun setelah upaya peningkatan

Berikut ini adalah perhitungan MTTR

$$\text{MTTR} = \frac{\text{TOTAL RECOVERY TIME}}{\text{TOTAL NUMBER OF REPAIR}}$$

2. Nilai *Mean Time Between Failure (MTBF)*

Penghitungan nilai MTBF MTTR dalam bentuk *cluster* baik di *state* awal maupun setelah upaya peningkatan.

Penghitungan MTBF adalah sebagai berikut.

$$\text{MTBF} = \frac{1}{\text{Failure Rate } (\lambda)}$$

which therefore mean

$$\text{MTBF} = \frac{\text{Operating Time}}{\text{Number of Failures}} = \frac{\text{Number of Cycles}}{\text{Number of Failures}}$$

3. Nilai *Availability*

Menghitung nilai *availability* dalam bentuk *cluster* baik di *state* awal maupun setelah upaya peningkatan. Berikut ini adalah perhitungan nilai *Availability*

$$A = \frac{MTBF}{(MTBF + MTTR)}$$

c) Analisis Kualitatif

Analisis Kualitatif Digunakan untuk mengetahui kelebihan atau kekeurangan lain diluar hasil perhitungan karena jika hanya dilakukan perhitungan kuantitatif hasilnya akan sangat berbeda tipis antara skema konfigurasi yang diusulkan. Menilai kualitas *cluster* dari penghitungan yang telah dilakukan secara kualitatif berdasarkan masalah keamanan data atau *vulnerability* yang terjadi.

HASIL DAN PEMBAHASAN

a. Objek Assesment

Objek assessment adalah Objek storage berdasarkan tipe nya yang diteliti di dalam penelitian. Berikut ini dituangkan dalam bentuk table 2.

Tabel 2. Objek Assesment

| No | Objek Penilaian | Implementasi | Redundancy | Kondisi Sebelum | Kondisi Sesudah |
|----|-----------------------------|------------------|-----------------------|--------------------|--|
| 1 | <i>Block Level Storage</i> | RAID | <i>Replication</i> | RAID 1 | RAID 1 + <i>Write Back</i> <i>Cace Battery</i> |
| 2 | <i>Block Level Storage</i> | RAID | <i>Erasure Coding</i> | RAID 5 | RAID 6 + <i>Write Back</i> <i>Cace Battery</i> |
| 3 | <i>File Level Storage</i> | <i>GlusterFS</i> | <i>Replication</i> | Replika dua | Replika tiga |
| 4 | <i>Object Level Storage</i> | CEPH Cluster | <i>Replication</i> | Replika dua | Replica tiga |
| 5 | <i>Object Level Storage</i> | CEPH Cluster | <i>Erasure Coding</i> | <i>Parity satu</i> | <i>Parity dua</i> |

b. Parameter Pengujian

Parameter pengujian dibutuhkan sebagai tolak ukur penelitian yang dilakukan. Berikut ini adalah parameter penelitian.

1. *MTBF (Mean Time Between Failure)*
 - 1) *MTBF* masing masing komponen
 - 2) *MTBF Cluster*
2. *MTTR (Mean Time to Repair)*
3. *Nilai Availability*
4. *Troughput*
 - 1) *Write Throughtput*
 - 2) *Read Throughtput*
5. Kapasitas penyimpanan yang dihasilkan

Pada penelitian ini akan menghasilkan model usulan pada skema *cluster* yang akan dibangun yang memiliki komposisi nilai *availability* yang tinggi dipadukan dengan *throughput* dan kapasitas yang tinggi juga.

c. Simulasi Perhitungan

Telah dilakukan percobaan penghitungan waktu recovery (MTTR) pada RAID sebelum peningkatan sebagai berikut :

1. Percobaan MTTR 1 = 0.61667 jam
2. Percobaan MTTR 2 = 0.60000 jam
3. Percobaan MTTR 3 = 0.60880 jam
4. Rata rata = 0.60849 Jam

Nilai MTBF = $2 \times 2000000 = 4000000$ Jam

Nilai availability nya adalah $4000000 / (4000000 + 0.60849) = 0.9999998486000$

Setelah dilakukan peningkatan dengan menambahkan redundancy nya maka nilai availability menjadi $= 0.9999998898000$

Tabel 3. Perhitungan RAID

| Sebelum (RAID 1) + Battery | | | | | | | | | Sesudah (RAID 1) + Battery | | | | | | | | |
|----------------------------------|-----|-----|-------------|---|---|--------------|-----|-----|----------------------------------|-----|-----|-------------|---|---|--------------|-----|-----|
| MTBF | | | MTTR | | | Availability | | | MTBF | | | MTTR | | | Availability | | |
| $2 \times 2.000.000 = 4.000.000$ | | | 0.61667 | | | 0.0999999848 | | | $2 \times 2.000.000 = 4.000.000$ | | | 0.6 | | | 0.0999999848 | | |
| | | | 0.6 | | | | | | | | | 0.61667 | | | | | |
| | | | 0.6088 | | | | | | | | | 0.6088 | | | | | |
| | | | AVG 0.60849 | | | | | | | | | AVG 0.60849 | | | | | |
| CPU | | | Memory | | | Throughput | | | CPU | | | Memory | | | Throughput | | |
| N | D | R | N | D | R | N | D | R | N | D | R | N | D | R | N | D | R |
| 0.3 | 0.3 | 0.3 | 5 | 5 | 5 | 650 | 715 | 450 | 0.3 | 0.3 | 0.3 | 4 | 5 | 4 | 650 | 715 | 450 |

maka setiap objek assessment akan dituangkan ke dalam tabel berikut ini

Tabel di atas menggunakan beberapa istilah dan singkatan yang berlaku untuk tabel sejenis sebagai berikut.

1. MTBF adalah *Mean Time Between Failure* yang ditulis dalam satuan jam. Setiap *harddisk* memiliki MTBF dua juta jam
2. MTTR adalah *Mean Time To Repair* yang ditulis dalam satuan Jam. Dibatasi sepuluh angka dibelakang koma
3. CPU adalah *Central Processing Unit* yang ditulis dalam persentasi dari penggunaan total dalam kondisi tertentu
4. Memory adalah persentasi penggunaan RAM.
5. *Throughput* adalah kecepatan transfer data yang dibutuhkan antara *dev/zero* ke *mount point* atau *mount point* ke */dev/null*.
6. N adalah kondisi sistem dalam keadaan normal.
7. D adalah kondisi sistem dalam keadaan kekurangan (*degraded*) *resource harddisk* atau OSD.
8. R adalah kondisi sistem dalam keadaan *rebuilding* atau *recovery*. Sistem akan dilihat alokasi *resource*-nya dalam keadaan tersebut.

Tabel di atas menunjukkan bahwa nilai *availability* antara sebelum dan sesudah upaya peningkatan tidak mendapatkan hasil karena tidak ada penambahan jumlah replika. Namun secara *robust* sistem mengalami peningkatan dikarenakan adanya pengefektifan penggunaan *battery write back cache*. Penggunaan benda tersebut akan mengurangi resiko terhadap terjadinya kerusakan data, kehilangan data pada saat sistem secara mendadak kehilangan listrik.

d. Resume hasil pengujian

Tabel 4. resume hasil pengujian

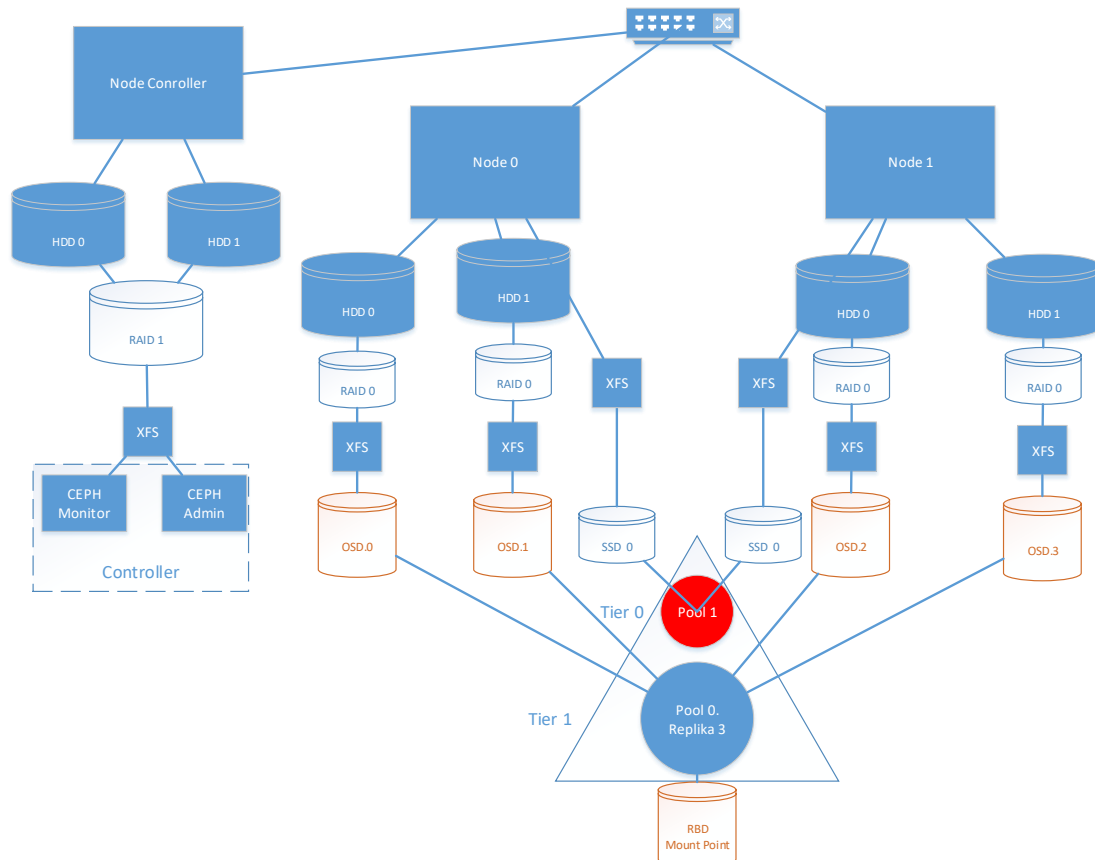
| No | Objek Percobaan | Nilai Availability Sebelum | Nilai Availability Sesudah | Selisish | Redudancy |
|----|---|----------------------------|----------------------------|----------------------|-----------|
| 1 | Block Level menggunakan RAID 1 data Penuh (replication) | 0.999998486000 | 0.999998898000 | 41×10^{-10} | 1 to 2 |
| 2 | Block Level menggunakan RAID 1 data kosong (replication) | 0.999998398000 | 0.999998479000 | 81×10^{-10} | 1 to 2 |
| 3 | Block Level menggunakan RAID 5 dan 6 (erasure coding) data Penuh | 0.999998486000 | 0.999998898000 | 41×10^{-9} | 1 to 2 |
| 4 | Block Level menggunakan RAID 5 dan 6 (erasure coding) data kosong | 0.999998333334 | 0.999998888889 | 55×10^{-9} | 1 to 2 |
| 5 | File Level menggunakan (replication) data penuh | 0.999998414225 | 0.999998505046 | 90×10^{-10} | 1 to 2 |

| No | Objek Percobaan | Nilai Availability Sebelum | Nilai Availability Sesudah | Selisish | Redudancy |
|----|--|----------------------------|----------------------------|----------------------|-----------|
| 6 | File Level menggunakan (replication) Data Kosong | 0.999999986111 | 0.9999998505046 | 15×10^{-10} | 1 to 2 |
| 7 | Object level CEPH cluster (replication) Data Penuh | 0.9999999236110 | 0.9999999143520 | 92×10^{-10} | 1 to 2 |

Tabel resume hasil pengujian di atas menunjukkan bahwa adanya peningkatan availability dengan skema dan metode yang diusulkan secara kuantitatif. Secara kualitatif pun Ketersediaan *Storage* akan bertambah kekuatannya (redundancy) baik pada scenario replica dua, tiga maupun erasure satu dan dua. Selain itu ada beberapa hal lain yang dapat disimpulkan dari percobaan yang dilakukan sebagai berikut :

1. Block adalah level objek yang memakan waktu recovery paling lama karena data yang ada ataupun tidakpun semuanya di recovery
2. Block storage yang menggunakan RAID Controller tidak menggunakan resource Memory dan CPU Server melainkan resource yang tersedia pada controller masing masing
3. Penggunaan battery write back cache pada RAID akan membantu
4. Replikasi relative lebih cepat dan lebih efisien resource dibandingkan dengan erasure coding jika dilakukan di level object atau file karena tidak melakukan proses encoding dan decoding
5. Erasure coding pada level object atau file akan menggunakan resource CPU besar besaran ketika rebuilding karena melakukan proses encoding dan decoding
6. rebuilding menggunakan ceph cluster terkadang akan sangat cepat karena bisa saja replika diambil dari OSD yang berbeda namun dalam node yang sama (localhot)
7. Object dan file melakukan recovery atau rebuilding sejumlah data yang dibutuhkan saja sehingga prosesnya efektif

Berdasarkan perhitungan dan analisis dari percobaan yang dilakukan , maka kami coba mengusulkan Topologi / deployment skema storage / model cluster storage yang mengkombinasikan penggunaan erasure coding dan replikasi sebagai rekomendasi di dalam penelitian ini sesuai gambar berikut :



Gambar 2. Usulan Model konfigurasi Cluster yang Diusulkan

Berdasarkan percobaan maka dihasilkan usulan model konfigurasi *cluster* dengan penjelasan sebagai berikut :

1. Berdasarkan hasil percobaan diketahui bahwa Object Level Storage adalah yang paling baik sehingga diusulkan penggunaan object level storage
2. Software *storage* menggunakan CEPH Cluster, terdiri dari node controller, dan node data
3. Interface Jaringan antar node disesuaikan dengan kebutuhan, penulis menyarankan menggunakan interface network 10GBE minimal supaya kecepatan Harddisk dan SSD antar node tidak terbuang sia sia (bottleneck)
4. Menggunakan Replika 3 untuk penyimpanan Data dan menggunakan replika dua untuk SSD
5. Setiap satu harddisk adalah satu buah OSD
6. OSD berada di atas linux file system XFS dan Menggunakan RAID 0 supaya dapat mengaktifkan fitur battery

KESIMPULAN

Dari penelitian yang dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

1. Vulnerability yang sangat penting untuk dimitigasi adalah kelemahan pada konfigurasi storage cluster, dalam hal ini vulnerability disebabkan oleh jumlah redundancy yang terlalu sedikit dan jika dibiarkan vulnerability ini akan menyebabkan adanya kehilangan atau kerusakan data.
2. Meningkatkan availability adalah dengan cara mengefektifkan penggunaan fault tolerance dengan memperbanyak jumlah replica menjadi tiga pada setiap sistem cluster storage.
3. Mengefektifkan fault tolerance adalah memperkecil waktu rebuilding atau mengurangi nilai MTTR waktu recovery atau repair dan mengaktifkan fungsi batre pada RAID.

4. Tipe Storage yang memiliki waktu rebuilding yang paling tinggi adalah object level storage karena rebuilding yang dilakukan hanya pada data yang tersedia saja.
5. Konfigurasi yang diusulkan adalah menggunakan RAID nol pada physical block level di masing masing harddisk yang diiringi oleh penggunaan RAID battery lalu setiap harddisk dijadikan satu buah OSD serta menyediakan tiga buah OSD sebagai replika.

DAFTAR PUSTAKA

- [1] William Stallings, 2011, *Network Security Essentials: Applications and Standards (Fourth edition)*, Pearson Education
- [2] Lawless, J.F., 1982, *Statistical Models and Methods for Lifetime Data*, John Wiley and Sons, Inc.,
- [3] Hoda Rohani, Azad Kamali Roosta, 2014, Amsterdam: *Calculating Total System Availability*, KLM-Air France
- [4] H. Paul Barringer, P.E., 1997, *Availability, Reliability, Maintainability, and Capability*, Barringer & Associates, Inc.
- [5] Reliability, Availability, Maintainability, and Cost Rationale Report Manual, Department of Defense of USA, 2009
- [6] Cluster Computing, Kumar Kaushik, Cochin University Of Science And Technology, 2008
- [7] Arif Sari, Murat Akkaya, 2015, *Fault Tolerance Mechanisms in Distributed Systems*, Girne American University
- [8] Deepali Mittal, Neha Agarwal, 2015, *review paper on Fault Tolerance in Cloud Computing*, IEEE.
- [9] Zeeshan Amin, Nisha Sethi, Harshpreet Singh, 2015, *Review on Fault Tolerance Techniques in Cloud Computing*, International Journal of Computer Applications (0975 – 8887)
- [10] Wendy Torell, Victor Avelar, 2004, *Mean Time Between Failure: Explanation and Standards*, APC.

Biodata Penulis

Muhamad Ikmal Wiawan, M.Kom, C.E.H memperoleh gelar Magister Komputer (M.Kom) dari Program Studi Teknik Informatika Universitas Langlangbuana. Lulus Tahun 2017 Saat ini sebagai Direktur Utama PT. Rumantak Seki Indotama

Agung Baitul Hikmah, S.Kom, M.Kom., memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Sistem Informasi STMIK Nusa Mandiri Jakarta, lulus tahun 2009. Tahun 2013 memperoleh gelar Magister Komputer (M.Kom) dari Program Studi Ilmu Komputer Program Pascasarjana STMIK Nusa Mandiri Jakarta. Saat ini sebagai Staf Pengajar program studi D3 Manajemen Informatika AMIK BSI Tasikmalaya