

Tugas 7 IF4031- Pemrograman Riak

Anggota Kelompok:

- 13512070 – Willy
- 13521076 - Ahmad Zaky

Hasil eksekusi berbagai command Riak:

```
$ sudo riak stop
```

```
ok
```

```
$ sudo riak start
```

```
$ sudo riak-admin member-status
```

```
===== Membership =====
Status      Ring      Pending   Node
-----
valid       32.8%     --        'riak@167.205.33.134'
valid       32.8%     --        'riak@167.205.33.191'
valid       34.4%     --        'riak@167.205.33.99'
-----
Valid:3 / Leaving:0 / Exiting:0 / Joining:0 / Down:0
```

```
# Command-command di bawah dijalankan bukan dari server Riak
```

```
$ curl http://167.205.33.191:8098/ping
```

```
OK
```

```
$ curl -v -X PUT http://167.205.33.191:8098/riak/bucketWillyZaky/mykey \
-H "Content-type: text/html" \
-d "<html><body><h1>My Data for key: mykey</h1></body></html>"
* timeout on name lookup is not supported
* About to connect() to 167.205.33.191 port 8098 (#0)
*   Trying 167.205.33.191...
* connected
* Connected to 167.205.33.191 (167.205.33.191) port 8098 (#0)
> PUT /riak/bucketWillyZaky/mykey HTTP/1.1
> User-Agent: curl/7.26.0
> Host: 167.205.33.191:8098
> Accept: */*
> Content-type: text/html
> Content-Length: 57
>
* upload completely sent off: 57 out of 57 bytes
< HTTP/1.1 204 No Content
< Vary: Accept-Encoding
< Server: MochiWeb/1.1 WebMachine/1.10.8 (that head fake, tho)
< Date: Mon, 02 Nov 2015 11:29:34 GMT
< Content-Type: text/html
< Content-Length: 0
<
* Connection #0 to host 167.205.33.191 left intact
```

```

* Closing connection #0

# Perintah asli menggunakan -X d menghasilkan response 501

$ curl http://167.205.33.191:8098/riak/bucketWillyZaky/mykey
<html><body><h1>My Data for key: mykey</h1></body></html>

$ curl -X GET http://167.205.33.191:8098/riak?buckets=true
{"buckets":["habibiefaried","Aldy","bucketSetyo","bucketWillyZaky","willy65","test","anda","alifradityar","bucketA","renusa","13512018","ramandika","kevinzhong","Orders","ichwan","kucingbooks","Wishnu","__riak_client_test__","sudib","bucketriefky","bucketFaiz","OrderSummaries","wibuteladan","jais","animals","food","emhaje","AdhityaRamadhanus","willyfitrahendria","test_riak","pandu","RumahPanda","andreas","fahziar","json","bucketritra","tolep","13512081","13509085","humbaba","fauzanhilmi","sonny","genta","Customers","andrian","bucketRanggaY","Michael","korean","hmif","13512042","yanuararistya","dalva","gentaiscool","mahasiswa","ontaimba","booksssss","hahahaha","gentas","willy","books_riak","riady","bucketRex","test_riak_2","reximir","guest","yomanovian","salvianreynaldi","asepbucket","quotes","AndrianOcto","animal","christian","MuhammadHusain","Ihsan","books","icha","calvin","edmund","bucketFeli","escepta","anpandu"]}

$ curl -v -X DELETE http://167.205.33.191:8098/riak/bucketWillyZaky/mykey
* timeout on name lookup is not supported
* About to connect() to 167.205.33.191 port 8098 (#0)
*   Trying 167.205.33.191...
* connected
* Connected to 167.205.33.191 (167.205.33.191) port 8098 (#0)
> DELETE /riak/bucketWillyZaky/mykey HTTP/1.1
> User-Agent: curl/7.26.0
> Host: 167.205.33.191:8098
> Accept: */*
>
< HTTP/1.1 204 No Content
< Vary: Accept-Encoding
< Server: MochiWeb/1.1 WebMachine/1.10.8 (that head fake, tho)
< Date: Mon, 02 Nov 2015 11:32:59 GMT
< Content-Type: text/html
< Content-Length: 0
<
* Connection #0 to host 167.205.33.191 left intact
* Closing connection #0

$ curl http://167.205.33.191:8098/riak/bucketWillyZaky?keys=true
{"props":{"name":"bucketWillyZaky","allow_mult":false,"basic_quorum":false,"big_vclock":50,"chash_keyfun":{"mod":"riak_core_util","fun":"chash_std_keyfun"},"dvv_enabled":false,"dw":"quorum","last_write_wins":false,"linkfun":{"mod":"riak_kv_wm_link_walker","fun":"mapreduce_linkfun"},"n_val":3,"notfound_ok":true,"old_vclock":86400,"postcommit":[],"pr":0,"precommit":[],"pw":0,"r":"quorum","rw":"quorum","small_vclock":50,"w":"quorum","write_once":false,"young_vclock":20},"keys":[]}

```

Jawaban Pertanyaan nomor 9

9. Pelajari mekanisme replikasi pada Riak

(<http://docs.basho.com/riak/latest/dev/advanced/replication-properties/>), dan jawablah pertanyaan berikut:

- a. Bagaimanakah sebuah data disimpan pada node? Jelaskan pada kasus mesin di atas, dimana terdapat sebuah cluster yang terdiri dari 4 node.

Vnode adalah abstraksi node pada Riak. Sebuah node mengandung banyak vnode. Hal ini memberikan kesempatan pada saat menambahkan sebuah node, vnode dipindahkan ke node yang baru. Data disimpan pada vnode. Data tersebut juga akan direplikasikan ke sebanyak n vnode lainnya. Jumlah vnode pada setiap node akan terdistribusi merata.

- b. Apakah yang dimaksud dengan ring size?

Ring size adalah banyaknya vnode yang ada pada satu ring. Jadi jika terdapat 4 node di satu cluster dan terdapat 8 vnode pada tiap node, maka ring size adalah 32.

- c. Apakah yang dimaksud dengan vnode?

Vnode (virtual node) adalah proses pada suatu node yang akan me-manage partisi pada sebuah Ring. Tiap partisi data memiliki vnode tertentu untuk menyimpan partisi itu.

- d. bagaimana algoritma quorum diterapkan pada Riak? Jelaskan menggunakan parameter N , R dan W

Kuorum adalah jumlah vnode minimal yang harus ditulis/dibaca pada tiap data. R disebut kuorum read dan W disebut kuorum write. Artinya, tiap penulisan harus dilakukan pada paling tidak R buah vnode dan pembacaan harus dilakukan pada paling tidak W buah vnode sebelum dianggap berhasil. Secara default, $N = 3$ dan $W = R = N / 2 + 1$, di mana N adalah banyak replikasi. Untuk mencapai konsistensi, syarat yang harus dipenuhi adalah $R + W > N$. Dengan begitu, maka tiap kali read dilakukan, paling tidak ada satu vnode yang telah ditulis dengan data terbaru.

- e. buatlah sebuah program pada Java yang menyimpan data pada Riak dengan menggunakan parameter W bernilai 0. Apakah data yang ditulis pada program ini akan tersimpan dengan baik pada database?

Dengan parameter $W = 0$, artinya penulisan akan dianggap sukses jika data telah berhasil ditulis pada 0 vnode. Dengan kata lain, sebelum menulis apapun Riak akan mengembalikan notifikasi bahwa penulisan sukses. Jadi, tidak ada jaminan bahwa data akan berhasil ditulis pada salah satu vnode.

Program Java:

```
public class Quorum0Test {
```

```

// This will create a client object that we can use to interact with Riak
private static RiakCluster setUpCluster() throws UnknownHostException {
    // This example will use only one node listening on localhost:10017
    RiakNode node = new RiakNode.Builder()
        .withRemoteAddress("167.205.33.99")
        .withRemotePort(8098)
        .build();

    // This cluster object takes our one node as an argument
    RiakCluster cluster = new RiakCluster.Builder(node)
        .build();

    // The cluster must be started to work, otherwise you will see errors
    cluster.start();

    return cluster;
}

public static void main(String[] args) {
    try {
        RiakObject quoteObject = new RiakObject()
            .setContentType("text/plain")
            .setValue(BinaryValue.create("String Quorum 0"));

        Namespace quotesBucket = new Namespace("buckets");
        Location quoteObjectLocation = new Location(quotesBucket, "keys");

        StoreValue storeOp = new StoreValue.Builder(quoteObject)
            .withLocation(quoteObjectLocation)
            .withOption(StoreValue.Option.W, new Quorum(0))
            .build();
        System.out.println("StoreValue operation created");

        RiakCluster cluster = setUpCluster();
        RiakClient client = new RiakClient(cluster);
        StoreValue.Response storeOpResp = client.execute(storeOp);

        cluster.shutdown();
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}

```

f. Berapakah setting N, R dan W yang baik untuk program yang banyak melakukan:

1) Pembacaan

R = 1, W = N. Artinya tiap pembacaan hanya akan dilakukan pada satu vnode, sedangkan penulisan dilakukan pada semua vnode. Dengan konfigurasi berikut, client hanya perlu

2) Penulisan

$R = N$, $W = 1$. Artinya tiap pembacaan akan dilakukan dari seluruh vnode, dan penulisan akan dianggap sukses saat satu vnode telah berhasil ditulis. Dengan konfigurasi berikut, client yang melakukan write hanya perlu menunggu 1 vnode untuk mengembalikan respon.

- g. mungkin jika kita melakukan setting $N = W$, dan jika terjadi kegagalan node, data dapat hilang? Mengapa?

Karena $N = W$, maka tiap write akan ditulis pada semua replikasi. Artinya, seluruh replica akan menyimpan copy dari data tersebut. Oleh karena itu, data tidak mungkin hilang selama masih ada satu vnode yang menyala yang berisi salah satu replika. Data dapat hilang jika banyak node yang gagal cukup banyak sehingga semua vnode yang berisi replika tersebut ikut gagal.