JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

ANALISIS THROUGHPUT PADA HADOOP MENGGUNAKAN ALGORITMA DELAY SCHEDULING UNTUK PENGIRIMAN 2 JOB YANG BERBEDA

WAHYUDDIN SAPUTRA[1], M. HASRUL H[2]
Jurusan Teknik Informatika[1,2],
Fakultas Sains dan Teknologi, Universitas Islam Negeri Alauddin Makassar
Email : wahyuddin.saputra@uin-alauddin.ac.id[1] , muhammad.hasrul@uin-alauddin.ac.id[2]

ABSTRAK

Pertumbuhan data yang semakin besar tidak lepas dari semakin berkembangnya teknologi saat ini. Hal tersebut menciptakan tantangan tersendiri dalam hal pengelolaan, pengoleksian, analisa dengan mengandalkan *system* database biasa yang umumnya digunakan untuk menyimpan dan mengelola data dengan jumlah yang tidak terlalu besar, sehingga dibutuhkan solusi untuk itu salah satunya dengan *Parallel computing* dengan menggunakan menggunakan Hadoop yang merupakan *platform* untuk mengolah data yang berukuran besar *(big data*) secara terdistribusi dan dapat berjalan diatas cluster. Penelitian ini menggunakan algoritma FIFO dan *Delay Scheduling* sebagai *job scheduler* dan menggunakan parameter pengujian *job throughput* sebagai acuan perhitungan performansi sistem. Dari hasil analisis menunjukkan bahwa penambahan jumlah job yang diberikan akan menyebabkan nilai *througput* menjadi kecil karena jumlah job sangat berpengaruh terhadap nilai *throughput*. Meski demikian untuk masing-masing skenario yang telah dilakukan penggunaan algoritma *delay scheduling* memiliki nilai *throughput* yang lebih besar bila dibandingkan dengan algoritma FIFO.
Kata Kunci : *Throughput, Hadoop,* FIFO, *Delay Scheduling*.

I.PENDAHULUAN

Pertumbuhan data yang semakin besar tidak lepas dari semakin berkembangnya teknologi saat ini. Hal tersebut menciptakan tantangan tersendiri dalam hal pengelolaan, pengoleksian, Analisa dengan mengandalkan *system database* biasa yang umumnya digunakan untuk menyimpan dan mengelola data dengan jumlah yang tidak terlalu besar.

Data yang terus bertambah baik dari segi kapasitas, kompleksitas serta volumenya tentunya membutuhkan tempat penyimpanan yang lebih kompleks dan

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

pengelolaan yang tepat agar mudah dalam mengolahnya. Untuk dapat menyimpan dan mengolah data yang berukuran besar (*big data*) secara baik dan cepat dibutuhkan teknologi komputer yang khusus yang disebut *high performance computer* atau super *computer*, tetapi untuk membangun suatu sistem super *computer* tersebut membutuhkan biaya yang tidak murah. Untuk mengatasi masalah ini, maka *platforms* yang digunakan untuk menyimpan dan mengolah big data menggunakan sebuah sistem yang disebut *parallel computing*. *Parallel computing* adalah penggunaan beberapa komputer yang saling terhubung untuk mengolah data dalam ukuran yang besar. Salah satu *platform* yang masih sering digunakan sampai saat ini untuk mengolah data yang berukuran besar *(big data*) secara terdistribusi dan dapat berjalan di atas cluster adalah *Hadoop*.

Hadoop merupakan *framework software* berbasis java dan open-source yang berfungsi untuk mengolah data yang besar secara terdistribusi dan berjalan di atas cluster yang terdiri atas beberapa komputer yang saling terhubung. *Hadoop* mempunyai kelebihan dari segi ekonomi karena tidak berbayar dan dapat diimplementasikan pada perangkat keras dengan spesifikasi yang tidak terlalu tinggi. Arsitektur *Hadoop* terdiri dari dua layer yaitu layer *MapReduce* dan *layer Hadoop Distributed File System* (HDFS). *MapReduce* merupakan *framework* dari aplikasi yang terdistribusi sedangkan *Hadoop Distributed File System* (HDFS) merupakan data yang terdistribusi.

II. METODE PENELITIAN

Hadoop merupakan *framework software* berbasis Java dan *opensource* yang berfungsi untuk mengolah data yang memiliki ukuran yang besar secara terdistribusi dan berjalan diatas cluster yang terdiri dari beberapa komputer yang saling terhubung (*parallel computing*). Berdasarkan *Hadoop* dapat mengolah data dalam jumlah yang sangat besar hingga petabyte (1 petabyte = 10245 bytes) dan dijalankan di atas ratusan bahkan ribuan komputer. *Hadoop* dibuat oleh Doug Cutting yang pada asalnya *Hadoop* ini adalah sub project dari Nutch yang digunakan untuk *search engine*. *Hadoop* bersifat open source dan berada di bawah bendera Apache *Software Foundation*.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

1. *Arsitektur Hadoop*

Hadoop terdiri dari *common Hadoop* yang berguna dalam menyediakan akses ke dalam file *system* yang didukung oleh *Hadoop*. *Common Hadoop* ini berisi paket yang diperlukan oleh JAR file, skrip yang dibutuhkan untuk memulai *Hadoop* dan dokumentasi pekerjaan yang telah dilakukan oleh *Hadoop*.

Bedasarkan inti dari *Hadoop* adalah terdiri dari:

1. *Hadoop* Distributed File *System* (HDFS) →Untuk data yang terdistribusi.
2. *MapReduce* → *Framework* untuk aplikasi dan programing

Sebuah *cluster* kecil pada *Hadoop* dapat terdiri dari satu *master* node dan beberapa *slave* node. *Master* node ini terdiri dari *NameNode* dan *JobTracker*, sedangkan *slave* node terdiri dari *DataNode* dan *TaskTracker*. *Hadoop* membutuhkan JRE 1.6 atau JRE dengan versi yang lebih tinggi. Dalam menjalankan dan menghentikan sistem pada *Hadoop* dibutuhkan ssh yang harus dibentuk antar *node* pada sebuah *cluster*.

2. HDFS

Hadoop Distributed File System (HDFS) merupakan file *system* berbasis Java yang terdistribusi pada *Hadoop*. Sebagai file *system* terdistribusi, HDFS berguna untuk menangani data dalam jumlah besar yang disimpan dan tersebar didalam banyak komputer yang berhubungan yang biasa disebut dengan *cluster*. File *system* terdistribusi pada *Hadoop* dapat diartikan sebagai file *system* yang menyimpan data tidak dalam satu Hard Disk Drive (HDD) atau media penyimpanan lainnya, tetapi data dipecah-pecah (file dipecah dalam bentuk block dengan ukuran 64 MB – bisa dikonfigurasi besarnya) dan disimpan tersebar dalam suatu cluster yang terdiri dari beberapa computer.HDFS menyimpan suatu data dengan cara membelahnya menjadi potongan potongan data yang berukuran 64 MB(default) dan potongan-potongan data tersebut kemudian disimpan tersebar dalam setiap node yang membentuk clusternya. Potongan-potongan data tersebut didalam HDFS disebut block.Ukuranblock pada setiap file tidak terpaku harus 64 MB, dimana ukuran block tersebut dapat disesuaikan dengan keinginan user. Meskipun data yang ada disimpan secara tersebar ke beberapa

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

node, namun dari kacamata user, data tersebut tetap terlihat seperti halnya kita mengakses file pada satu komputer. File yang secara fisik tersebar dalam banyak komputer dapat diperlakukan layaknya memperlakukan file dalam satu computer.

3. *MapReduce*

MapReduce adalah sebuah *framework* untuk aplikasi dan programming yang diperkenalkan oleh Google dan digunakan untuk melakukan suatu pekerjaan dari komputasi terdistribusi yang dijalankan pada sebuah cluster. *MapReduce* ini terdiri dari konsep fungsi map dan reduce yang biasa digunakan pada *functional programming*. Salah satu program yang menggunakan konsep *MapReduce* yang telah disediakan oleh *Hadoop* adalah *Wordcount*. *Wordcount* merupakan program yang bertujuan untuk menghitung kata pada file plaintext. Proses *MapReduce* pada *Wordcount* ini dibagi menjadi 2 tahap yaitu proses mapping dan reducing.

4. *FIFO Scheduling*

Algoritma FIFO merupakan *default* yang digunakan oleh *Hadoop* pada proses penjadwalan. Algoritma ini mengatasi permasalahan antrian pada *job* dengan cara menjalakan sebuah *job* yang datang untuk pertama kali. Algoritma FIFO tidak menangani adanya skala prioritas dan perhitungan *long jobs* atau *short jobs*. Sehingga mengakibatkan penggunaan algoritma FIFO kurang efektif. Pada implementasi algoritma FIFO dalam server berskala besar, algoritma FIFO dapat menurunkan performansi dari sisi server terutama pada sistem yang mempunyai layanan *sharing data* pada *multiple-user*.

5. *Delay Scheduling*

Algoritma *Delay Scheduling* mempunyai struktur yang berbeda dibandingkan dengan penjadwalan pada konfigurasi *default* dari sebuah *Hadoop* sistem. Pada konfigurasi *default* dari sebuah *Hadoop* adalah memakai algoritma *FIFO* sebagai teknik utama penjadwalan.

Delay Scheduling tidak menggunakan mekanisme *FIFO* yang memindahankan data pada *virtual hard disk* yang ada pada *Hadoop* dan memerlukan sinkronisasi

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

terhadap data yang dipakai. Sehingga beberapa perpindahan *Resource* ini membuat *job* mengalami *fail* dan dilakukan pengulangan *job* tidak digunakan karena pada awal *job* dibuat, *Delay Scheduling* sudah membagi *Resource* data terhadap *pool* yang sesuai dengan *Resource* data yang ada pada *virtual hard disk Hadoop* karena konfigurasi *pool* merupakan karakterisktik dari *Fair Scheduler* yang dimodifikasi dengan algoritma *Delay Scheduling*. Selain itu *Delay Scheduler* akan menggunakan metode menunda jalannya *jobs* selanjutnya untuk memperbaiki data lokalitas sebelumnya. Sehingga dapat meminimalkan *Response Time* dan memaksimalkan *Job throughput*.

6. Skenario Pengujian

Tabel 1. Skenario Pengujian

Jenis *Job*	FIFO dan Delay scheduling					
Wordcount & Grep	Skenario 1	5 *jobs*	10 *jobs*	20 *jobs*	50 *jobs*	50 *jobs*
Wordcount & RandomTextwriter	Skenario 2	5 *jobs*	10 *jobs*	20 *jobs*	50 *jobs*	50 *jobs*
Grep & RandomTextWriter	Skenario 3	5 *jobs*	10 *jobs*	20 *jobs*	50 *jobs*	50 *jobs*

Tabel di atas merupakan skenario pengujian yang akan dilakukan dimana pada proses pengiriman *job* menggunakan tiga jenis *job* yaitu *job wordcoun*, *job grep*, *job randomtextwriter*, dengan menggunakan kombinasi 2 jenis *job* yaitu antara *job wordcount* dengan *job grep*, *job wordcount* dengan *job randomtextwriter*,dan *job grep* dengan *job randomtextwriter*.

Semua skenario terdiri dari lima *virtual* user yang jumlah *job*-nya diberikan tidak harus sama setiap user karena jumlah user tidak mempengaruhi performansi scheduling melainkan jumlah *job*. Adapun *resource data* yang digunakan sebagai berikut:

Tabel 2. R*esource* data

Data	Sumber	Size
1	facebook-names-withcount.txt	2,3 GB
2	facebook-names-unique.txt	1,5 GB
3	wikipedia-wordlist-sraveau-20090325.txt	700 MB
4	facebook-f.last.txt	155 MB
5	facebook-firstnames-withcount.txt	70 B

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

7.Teknik Analisis Data

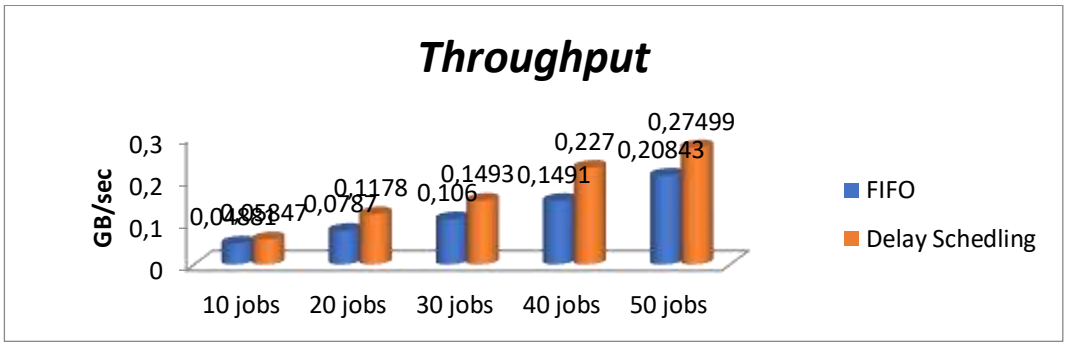
Semua skenario terdiri dari lima *virtual* user yang jumlah *job*-nya tidak harus sama setiap user karena jumlah user tidak mempengaruhi performansi *scheduling* melainkan jumlah *job*. Hal ini dilakukan untuk dapat meneliti antara antrian yang memiliki *job* berjenis sama dan antrian yang memiliki jenis *job* yang berbeda yang penempatan jenis *job* dilakukan secara acak.

Pada parameter ini akan diukur jumlah *job* yang berhasil diselesaikan dalam selang waktu tertentu. Nilai yang dihasilkan akan diperoleh dari awal suatu *job* diselesaikan pada suatu antrian. Satuan pada parameter ini adalah *job*/jam.

III.HASIL DAN PEMBAHASAN

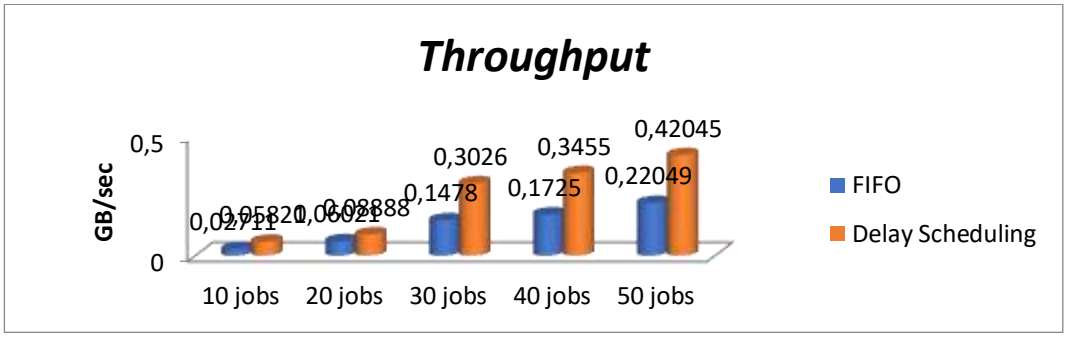
A.Analisis Yang Telah Dilakukan

a. Pengiriman *Job Wordcount & Grep*



Grafik 1. *Throughput* Skenario

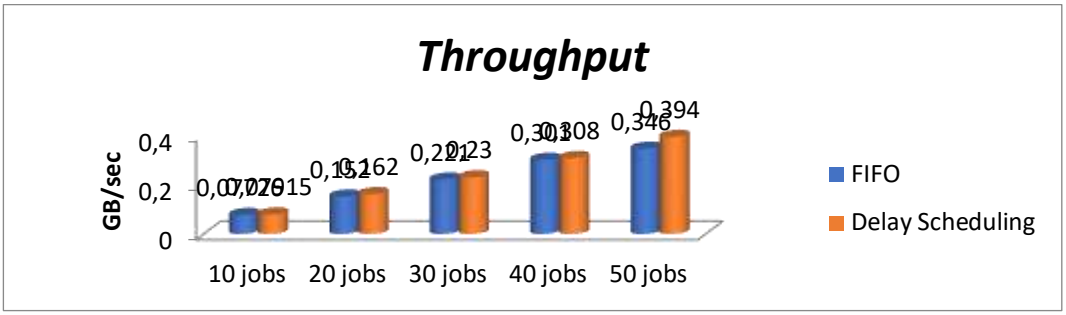
b. Pengiriman *Job Wordcount & RandomTextwriter*



Grafik 2. *Throughput* Skenario 2

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

c. Pengiriman *Job Grep & RandomTextWriter*



Grafik 3. *Throughput* Skenario 3

B.Pembahasan

Pada grafik 1 parameter *Job throughput* pada algoritma FIFO jika dibandingkan dengan algoritma *Delay Scheduling* memiliki ciri mengalami penurunan seiring bertambahnya jumlah job seperti yang terlihat pada semua jumlah *job* terjadi peningkatan nilai *Throughput* dari FIFO dengan ukuran file 11630MB, 22325MB, 28850MB, 42880MB, dan 53455MB pada kedua algoritma . Hal ini dapat terjadi karena *resourse* data sangat berpengaruh sehingga selisih jumlah maps yang jauh berbeda dimana pada *Delay Scheduling* lebih besar. Jumlah maps yang berbeda dikarenakan sistem random pada permintaan job dari *client*. Sehingga nilai *Job throughput* yang memiliki jumlah *maps* besar akan menambah nilai *Job throughput*. Hal ini diperkuat pada tabel 3 di bawah.

Tabel 3. Jumlah Maps data 10

Jumlah *Maps* FIFO	Jumlah *Maps Delay Scheduling*
11	3
24	1
3	24
1	36
36	24
36	1
11	11
3	11
36	24
24	24

Berdasarkan tabel 3 total maps pada FIFO yaitu 185 maps sedangkan pada Delay Scheduling 159 maps. Selisih 26 maps sangat berpengaruh sehingga meningkatkan nilai Job throughput.

Pada grafik 2 parameter *Job throughput* pada algoritma FIFO jika dibandingkan dengan algoritma *Delay Scheduling* memiliki ciri dimana meningkatnya nilai *Job*

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

throughput berpengaruh pada jumlah *job*. Semakin bertambahnya jumlah *job* menyebabkan nilai *throughput* yang semakin kecil pada kedua algoritma. Hal ini dapat terjadi karena waktu yang digunakan untuk setiap jenis *job* yang berbeda ditambah lagi waktu yang ditambahkan saat menangani *job*s yang *fail* dan kemudian diulang kembali sehingga *job*s yang awalnya sedang diproses mengalami penundaan waktu mengalami delay sampai *job*s yang dikirim kembali selesai baru kemudian akan dilanjutkan dengan *job* yang telah mengalami delay sebelumnya sehingga pada skenario ini algoritma Delay Scheduling lebih unggul dibanding dengan FIFO seperti yang terlihat pada semua jumlah *job*s terlihat peningkatan nilai througput bila dibandingkan dengan FIFO dengan ukuran file pada FIFO 10855MB, 18205MB, 51225MB, 58455MB, dan 68750MB dan 10855MB, 18340MB, 51360MB, 58590MB, dan 68885MB pada delay scheduling. Hal ini terlihat pada semua jumlah *job* dari Delay Scheduling yang menghasilkan nlai lebih besar dari FIFO.

Pada grafik 3 parameter *Job throughput* pada algoritma FIFO jika dibandingkan dengan algoritma Delay Scheduling memiliki ciri dimana nilai *Job throughput* pada Delay Scheduling memiliki nilai throughput yang lebih besar dari FIFO seperti terlihat pada semua jumlah *job* selisih antara keduanya dengan ukuran file pada FIFO 13600MB, 23055MB, 32050MB, 43205MB, dan 53680MB dan 13500MB, 20695MB, 31550MB, 41525MB, dan 53380 pada delay scheduling. Hal ini dapat terjadi karena resourse data sangat berpengaruh sehingga selisih jumlah maps yang jauh berbeda yang disebabkan sistem random pada permintaan *job* dari *client*. Sehingga nilai *Job throughput* yang memiliki jumlah maps keci akan menambah nilai *Job throughput*. Hal ini diperkuat pada tabel perbedaan jumlah maps pada jumlah *job* 10 *job*s.

Tabel 4. Jumlah Maps 10

Jumlah Maps FIFo	Jumlah Maps Delay Scheduling
50	50
36	36
50	50
24	24
50	11
50	50

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 7, Nomor. 2, Oktober 2022
P –ISSN : 2541-1179, E-ISSN : 2581-1711
OJS :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

36	24
50	50
24	50
50	3

Berdasarkan tabel 5 total maps pada FIFO yaitu 420 maps sedangkan pada *Delay Scheduling* 348 maps. Selisih 72 maps sangat berpengaruh sehingga meningkatkan nilai *Job throughput*.

IV.KESIMPULAN

Pada pengukuran *respons time* untuk kedua algoritma menunjukkan penambahan jumlah *job* akan mengakibatkna nilai *respons time* bertambah pada masing-masing algoritma meski demikian algoritma *delay scheduling* menghasilkan *respons time* yang lebih sedikit daripada algoritma FIFO sehingga dapat meningkatkan nila *throughput* pada algoritma *Delay Scheduling*.

DAFTAR PUSTAKA

Arslan Engin, Shekhar mrigank, and kosar Tevfik, 2014.Locality and Network-Aware Reduce Task Scheduling forData-Intensive Applications.DataCloud 2014.

Apache TM *Hadoop* @ homepage. http://*Hadoop*.apache.org/. Diakses 17 Oktober 2013.

Chuck Lam. (2011). *Hadoop* In Action. Stamford: Mainning Publications Co.

Colin White. (2012, January). *MapReduce* and the Data Scientist. BI Research.

Dima May. (2012). *Hadoop* Distributed File *System* (HDFS) Overview. coreservlets.com.

Magang Industri. (2013). Definisi Cloud Computing. Meruvian.org Cloud Computing.

Priagung Khusumanegara. 2014. *Analisis Performa Kecepatan MapReduce pada Hadoop Menggunakan TCP Packet Flow Analysis.* Universitas Indonesia

Rasooli Aysan, Douglas G. Down (2011). Guidelines for Selecting *Hadoop* Schedulers based on *System* Heterogeneity, Hamilton.

Thirumala Rao. B, Reddy. Dr. L.S.S.2011.*Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environment.*International Jurnal of Cumputer Application (0975-8887) volume 34- No.9, November 2011

Wang Yiantian, Rao Ruonan, and Wang Yingling. 2014. *A Round Robin with Multiple Feedback Job Sceduler in Hadoop*.978-1-4799-3 /114. 2014 IEEE.