
Big Data - Hadoop/MapReduce

Assignment Project Exam Help

Sambit Sahu
<https://powcoder.com>



Add WeChat powcoder

Logistics

- Quiz 1 next week. Sample released.
- Assignment 2 (tentative) draft uploaded.
- In-depth exploration topics
 - Kubernetes and DevOps: you will explore containers, Kubernetes and its application to cloud application devOps pipeline.
 - Kafka and Events/Logs: you will explore how kafka and kafka stream used for handling events/logs including internals, use cases and scale.
 - Elastic Search and data pipeline: understand in detail Elastic Search how it is used for handling unstructured data and build data pipeline to process and query such data efficiently.
 - Spark and cluster computing: learn how to effectively use Spark at scale and build use cases for demonstrating compute in a cluster including scale, administration, performance tuning.
- Paper reviews

Assignment Project Exam Help

<https://powcoder.com>

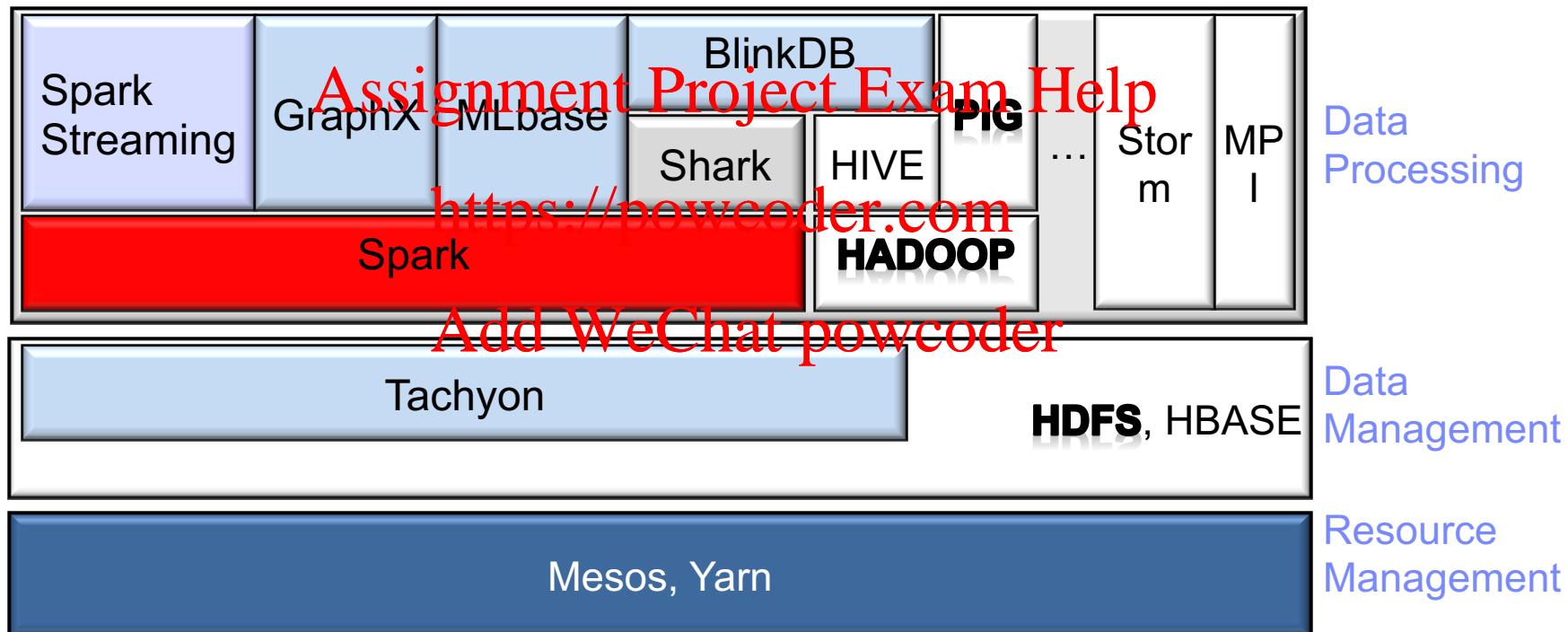
Add WeChat powcoder

Cluster Computing

- Designing a cluster
 - Automation of tasks in a cluster
 - Assignment Project Exam Help
 - Programming in a cluster
 - Hadoop/MapReduce
 - Spark
- <https://powcoder.com>
- Add WeChat powcoder

Big Data System Architecture

- Current system modules for each software layer



Agenda

- Why Big Data?

Assignment Project Exam Help

- Apache Hadoop <https://powcoder.com>

- Introduction [Add WeChat powcoder](#)
 - Architecture
 - Programming

Hypothetical Job

- You just got an awesome job at data-mining start-up ..
Congratulations !!
 - Free Snacks, Soda and Coffee YAY
 - Assignment Project Exam Help

<https://powcoder.com>

- Your first day of work you are given a task
 - Add WeChat powcoder
 - The company has a new algorithm they want you to test.
 - Your boss gives you
 - The algorithm library
 - A test machine and
 - 1GB input data file

Java Document Scorer Program

```
public static long scoreDocument(String fileName, Scorer scorer) throws IOException {  
    BufferedReader reader = new BufferedReader(new FileReader(fileName));  
  
    long totalScore = 0;  
    String line = null;  
    while((line = reader.readLine()) != null) {  
        totalScore += scorer.getScore(line);  
    }  
    reader.close();  
  
    return totalScore;  
}
```

Throughput 1GB per hour.

What if we wanted to process 10GB data set? 10hours!!
How can we improve the performance?

Assignment Project Exam Help Read Input

<https://powcoder.com>

Process Data

Add WeChat powcoder

Some Options

1. Faster CPU
2. More Memory
3. Increase the number of cores
4. Increase the number of threads
5. Increase the number of threads and cores

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Java Document Scorer Program – Multi Threaded

```
public static long scoreDocument(String fileName, Scorer scorer, int threads)
    throws Exception {

    BufferedReader reader =
        new BufferedReader(new FileReader(fileName));

    //thread safe structures
    BlockingQueue<String> queue =
        new LinkedBlockingQueue<String>(threads + 100);

    //initialize and start threads
    AtomicInteger completionCounter = new AtomicInteger();
    ExecutorService executors = Executors.newFixedThreadPool(threads);
    List<CounterThread> counters = new ArrayList<CounterThread>();
    for(int i=0;i<threads; i++) {
        executors.execute(new CounterThread(queue, scorer, completionCounter));
    }
    String line = null;
    while((line = reader.readLine()) != null) {
        queue.put(line);
    }

    //terminating condition for threads
    for(int i=0;i<threads; i++)
        queue.put("EXIT");
    while(completionCounter.intValue() < threads) {
        Thread.sleep(100);
    }

    executors.shutdown();
    reader.close();

    //summarize results
    long total = 0;
    for(CounterThread counter : counters)
        total+=counter.getTotal();

    return total;
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

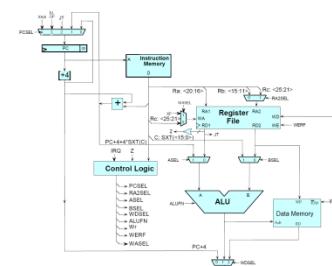
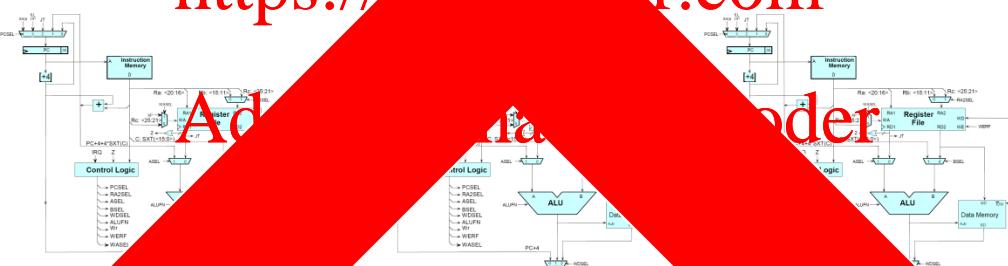
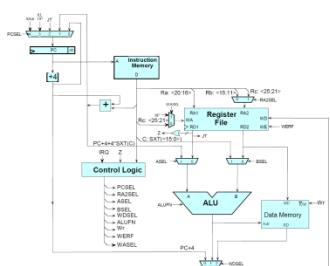
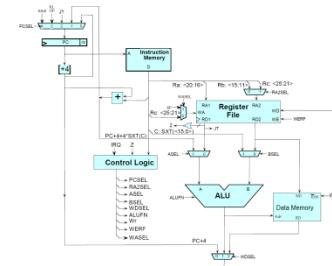
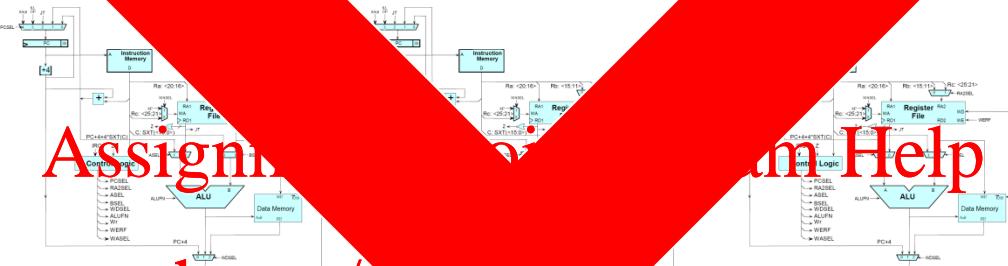
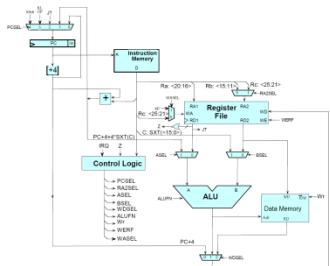
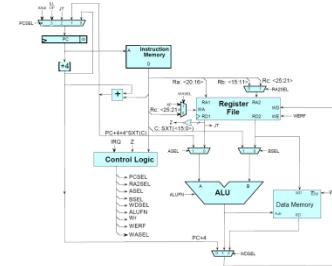
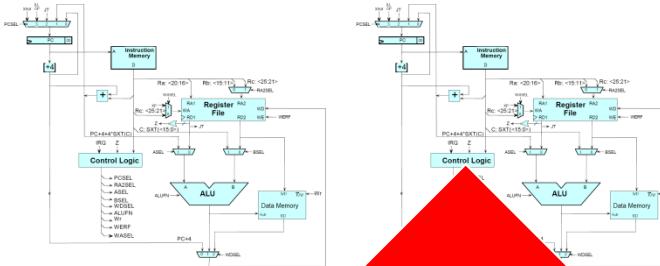
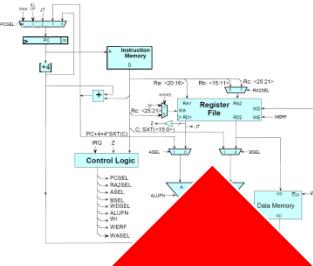
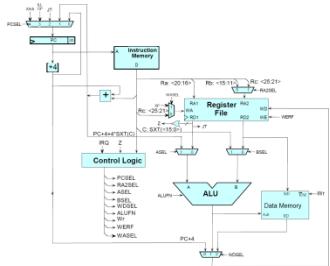
```
static class CounterThread implements Runnable {
    BlockingQueue<String> queue = null;
    Scorer scorer = null; AtomicInteger ai = null; long total=0;
    CounterThread(BlockingQueue<String> queue, Scorer scorer,
                  AtomicInteger ai ) {
        this.queue=queue; this.scorer=scorer; this.ai=ai;
    }
    @Override
    public void run() {
        while(true) {
            try {
                String line = queue.take();
                //terminating condition
                if(line.equals("EXIT")) break;
                total+=scorer.getScore(line);
            }catch(Exception e) {
                break;
            }
        }
        ai.incrementAndGet();
    }

    public long getTotal() {
        return total;
    }
}
```

Throughput 4GB per hour.

How long for 100GB?
What else can we do?

Get An Even Faster Machine with more Cores?



The diagram illustrates the memory assignment for the ALU and Data Memory components in the PC+HEMATIC system. The ALU is assigned to memory location 159-160, and the Data Memory is assigned to memory location 158-159.

<https://www.sher.com>

Adressen und Bilder

Current Tools

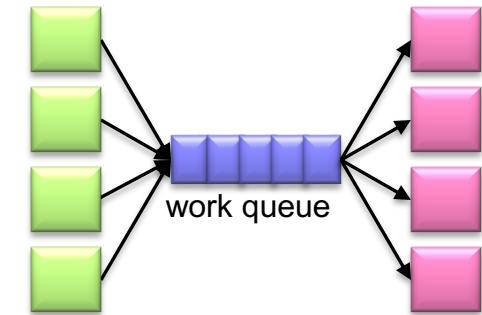
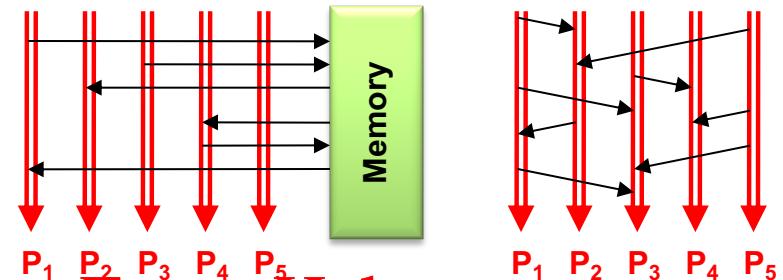
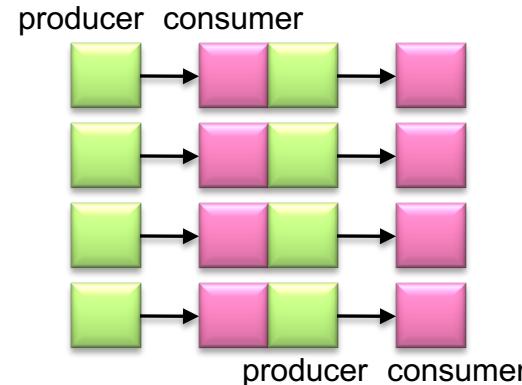
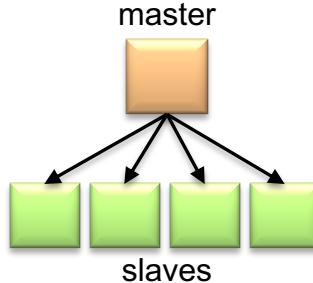
- Programming models
 - Shared memory (pthreads)
 - Message passing (MPI)

- Design Patterns

Assignment Project Exam Help

- Master-slaves
- Producer-consumer flows
- Shared work queues

Add WeChat powcoder



Where the rubber meets the road

- Concurrency is difficult to reason about
- Concurrency is even more difficult to reason about
 - At the scale of datacenters (even across datacenters)
 - In the presence of failures
 - In terms of multiple interacting services
- Not to mention debugging...
- The reality: <https://powcoder.com>
 - Lots of one-off solutions, custom code
 - Write your own dedicated library, then program with it
 - Burden on the programmer to explicitly manage everything

What's the common theme?

- To improve performance, you have to re-write the code
- The code has to adapt to the expected performance.
 - This doesn't work since you may not know the amount of data beforehand.
- The actual Intellectual Property (IP) of the company is the analytic algorithm
 - However a lot of effort is spent on scaling the analytic

Big Data - Motivation

- Google processes 20 PB a day (2008)
- Wayback Machine has 3 PB + 100 TB/month (3/2009)
- Facebook has 2.5 PB of user data + 15 TB/day (4/2009)
- eBay has 6.5 PB of user data + 50 TB/day (5/2009)
- CERN's LHC will generate 15 PB a year

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

640K ought
to be enough
for anybody.



Enter .. Apache Hadoop

- Hadoop is a high-level Open Source project
 - Under Apache Software Foundation
 - Inspired by Google's MapReduce and GFS papers
- It contains several individual projects
 - HDFS
 - MapReduce
 - Yarn
- It also has a slew of related projects
 - PIG
 - HIVE
 - Hbase
- Has been implemented for the most part in Java.

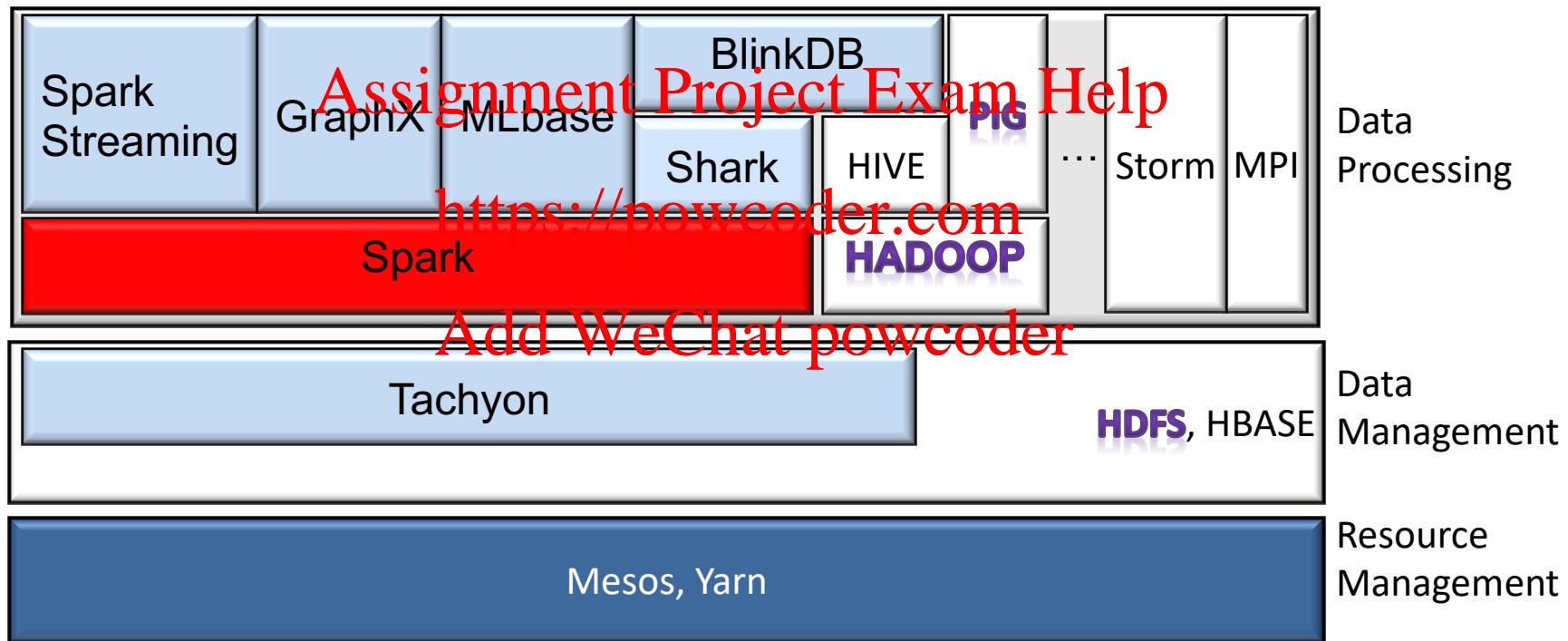
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Big Data System Architecture

- Current system modules for each software layer



A closer look

```
public static long scoreDocument(String fileName, Scorer scorer, int threads)
    throws Exception {

    BufferedReader reader =
        new BufferedReader(new FileReader(fileName));

    //thread safe structures
    BlockingQueue<String> queue =
        new LinkedBlockingQueue<String>(threads + 100);

    //initialize and start threads
    AtomicInteger completionCounter = new AtomicInteger();
    ExecutorService executors = Executors.newFixedThreadPool(threads);
    List<CounterThread> counters = new ArrayList<CounterThread>();
    for(int i=0;i<threads; i++) {
        executors.execute(new CounterThread(queue, scorer, completionCounter));
    }
    String line = null;
    while((line = reader.readLine()) != null) {
        queue.put(line);
    }

    //terminating condition for threads
    for(int i=0;i<threads; i++)
        queue.put("EXIT");
    while(completionCounter.intValue() < threads) {
        Thread.sleep(100);
    }

    executors.shutdown();
    reader.close();

    //summarize results
    long total = 0;
    for(CounterThread counter : counters)
        total+=counter.getTotal();

    return total;
}
```

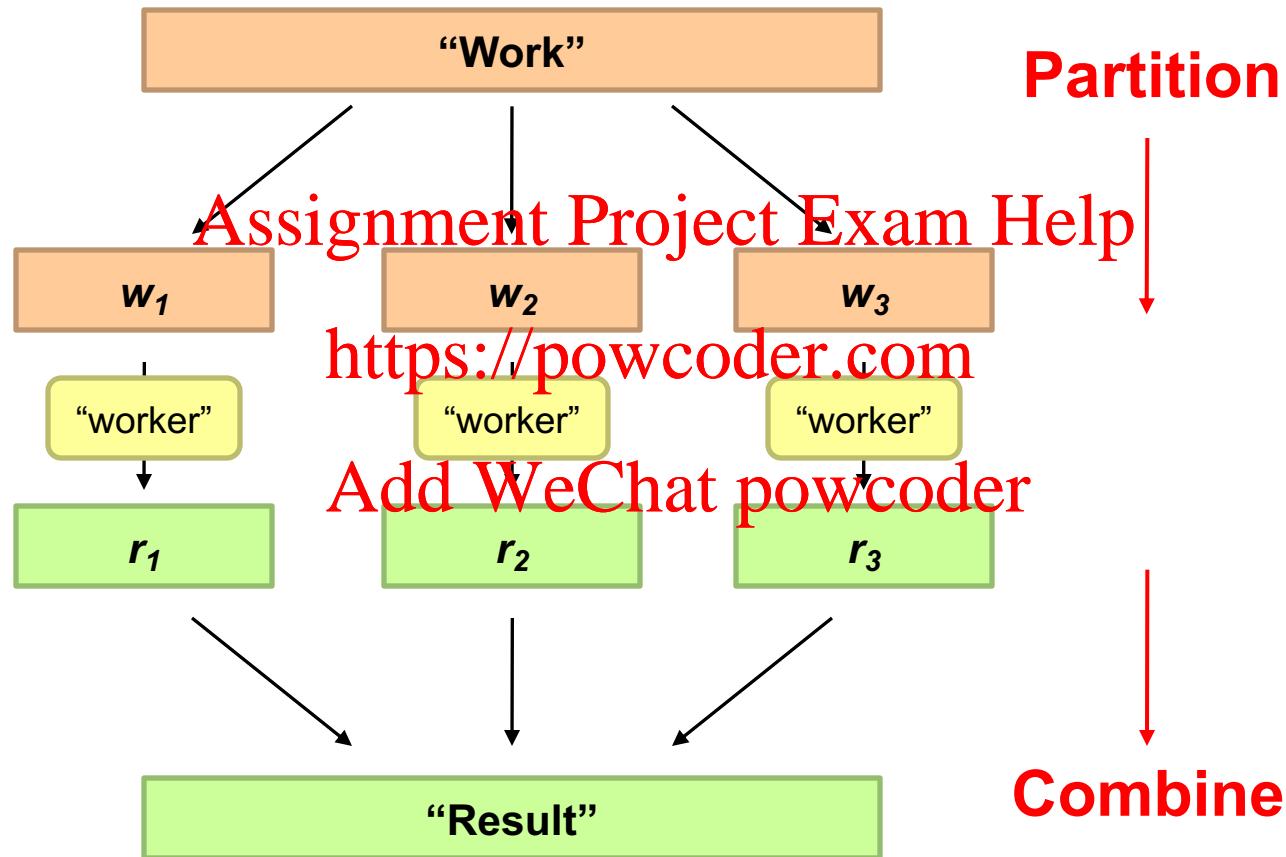
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder Partition Work

Combine Results

Divide and Conquer



Parallelization Challenges

- How do we assign work units to workers?
- What if we have more work units than workers?
- What if workers need to share partial results?
Assignment Project Exam Help
- How do we aggregate partial results?
- How do we know all the workers have finished?
<https://powcoder.com>
- What if workers die?
Add WeChat powcoder

What is the common theme of all of these problems?

What's the point?

- It's all about the right level of abstraction
 - The von Neumann architecture has served us well, but is no longer appropriate for the multi-core/cluster environment
- Hide system-level details from the developers
 - No more race conditions, lock contention, etc.
- Separating the *what* from *how*
 - Developer specifies the computation that needs to be performed
 - Execution framework ("runtime") handles actual execution

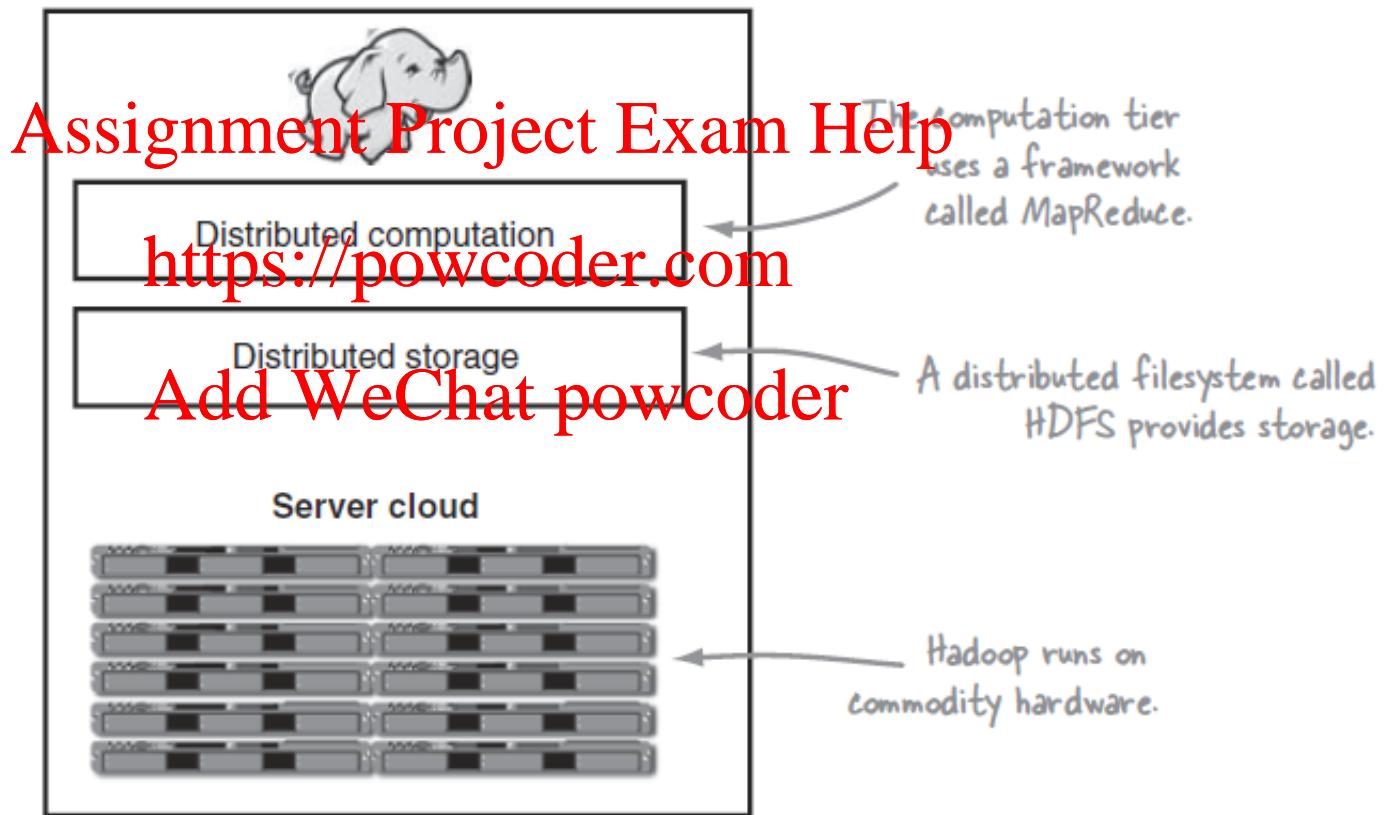
The datacenter *is* the computer!

“Big Ideas”

- Scale “out”, not “up”
 - Limits of SMP and large shared-memory machines
- Move processing to the data
 - Clusters have limited bandwidth
- Process data sequentially, avoid random access
 - Seek operations are expensive, disk throughput is reasonable
- Seamless scalability
 - From the mythical man-month to the tradable machine-hour

Hadoop

- Platform for distributed **storage** and **computation**
 - HDFS
 - MapReduce
 - Ecosystem



Source: **Hadoop in Practice**, Alex Holmes, Manning Publications Co., 2012

What are we missing here?

```
public static long scoreDocument(String fileName, Scorer scorer, int threads)
    throws Exception {
    BufferedReader reader =
        new BufferedReader(new FileReader(fileName));
    //thread safe structures
    BlockingQueue<String> queue =
        new LinkedBlockingQueue<String>(threads + 100);
    //initialize and start threads
    AtomicInteger completionCounter = new AtomicInteger();
    ExecutorService executors = Executors.newFixedThreadPool(threads);
    List<CounterThread> counters = new ArrayList<CounterThread>();
    for(int i=0;i<threads; i++) {
        executors.execute(new CounterThread(queue, scorer, completionCounter));
    }
    String line = null;
    while((line = reader.readLine()) != null) {
        queue.put(line);
    }
    //terminating condition for threads
    for(int i=0;i<threads; i++)
        queue.put("EXIT");
    while(completionCounter.intValue() < threads) {
        Thread.sleep(100);
    }
    executors.shutdown();
    reader.close();
    //summarize results
    long total = 0;
    for(CounterThread counter : counters)
        total+=counter.getTotal();
    return total;
}
```

Sequential File Read

Assignment Project Exam Help

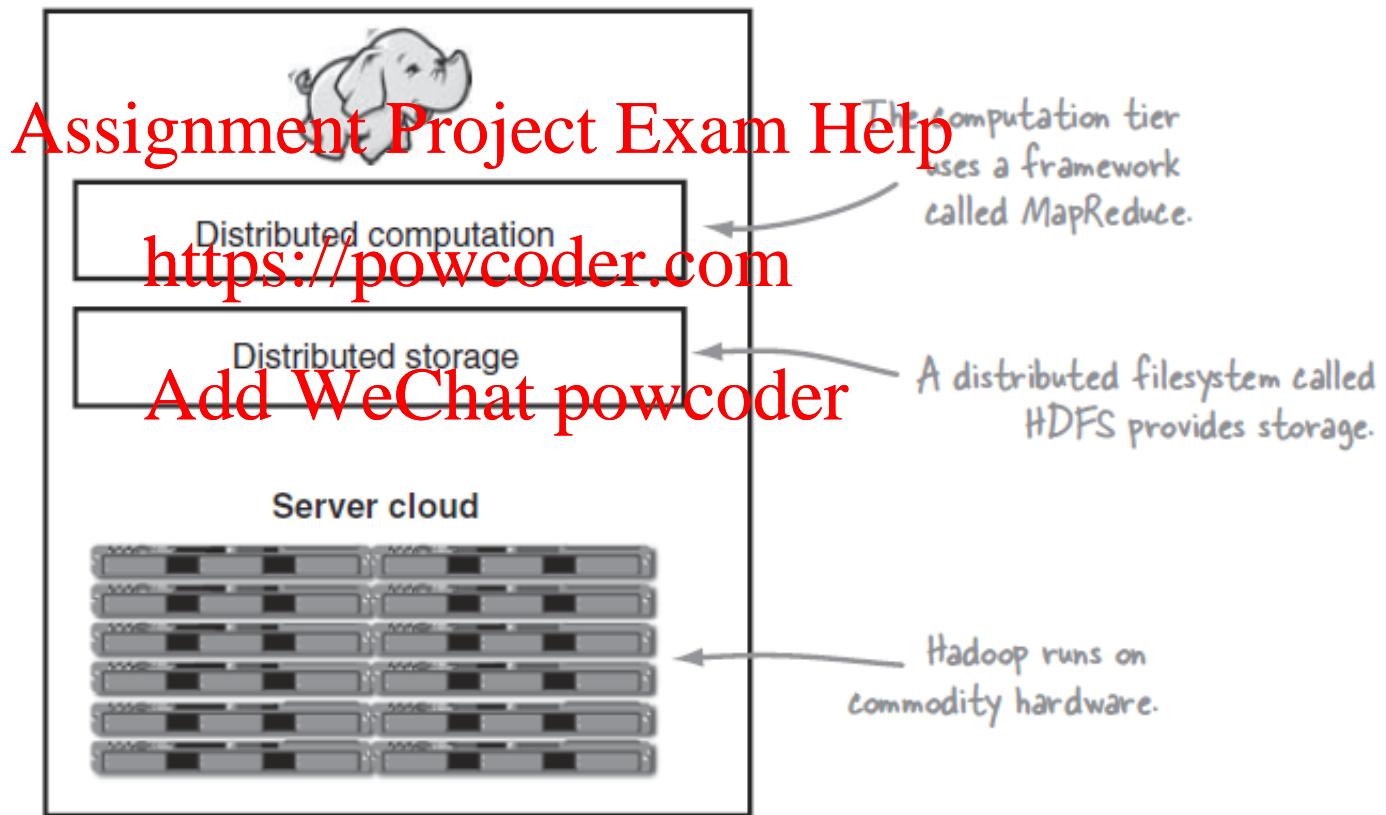
<https://powcoder.com>

Add WeChat powcoder Partition Work

Combine Results

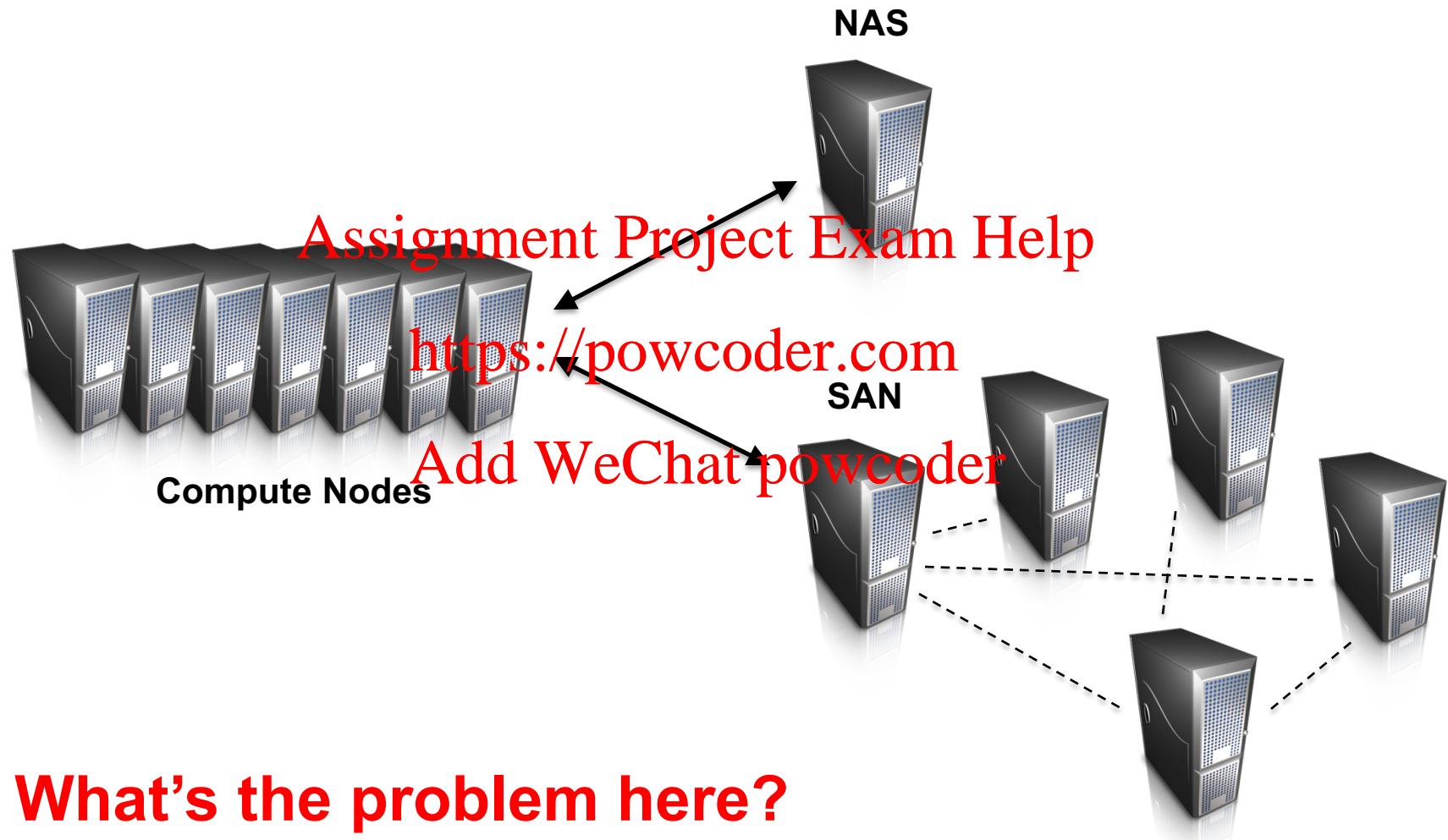
Hadoop

- Platform for distributed **storage** and **computation**
 - **HDFS**
 - **MapReduce**
 - **Ecosystem**



Source: **Hadoop in Practice**, Alex Holmes, Manning Publications Co., 2012

How do we get data to the workers?



HDFS: Assumptions

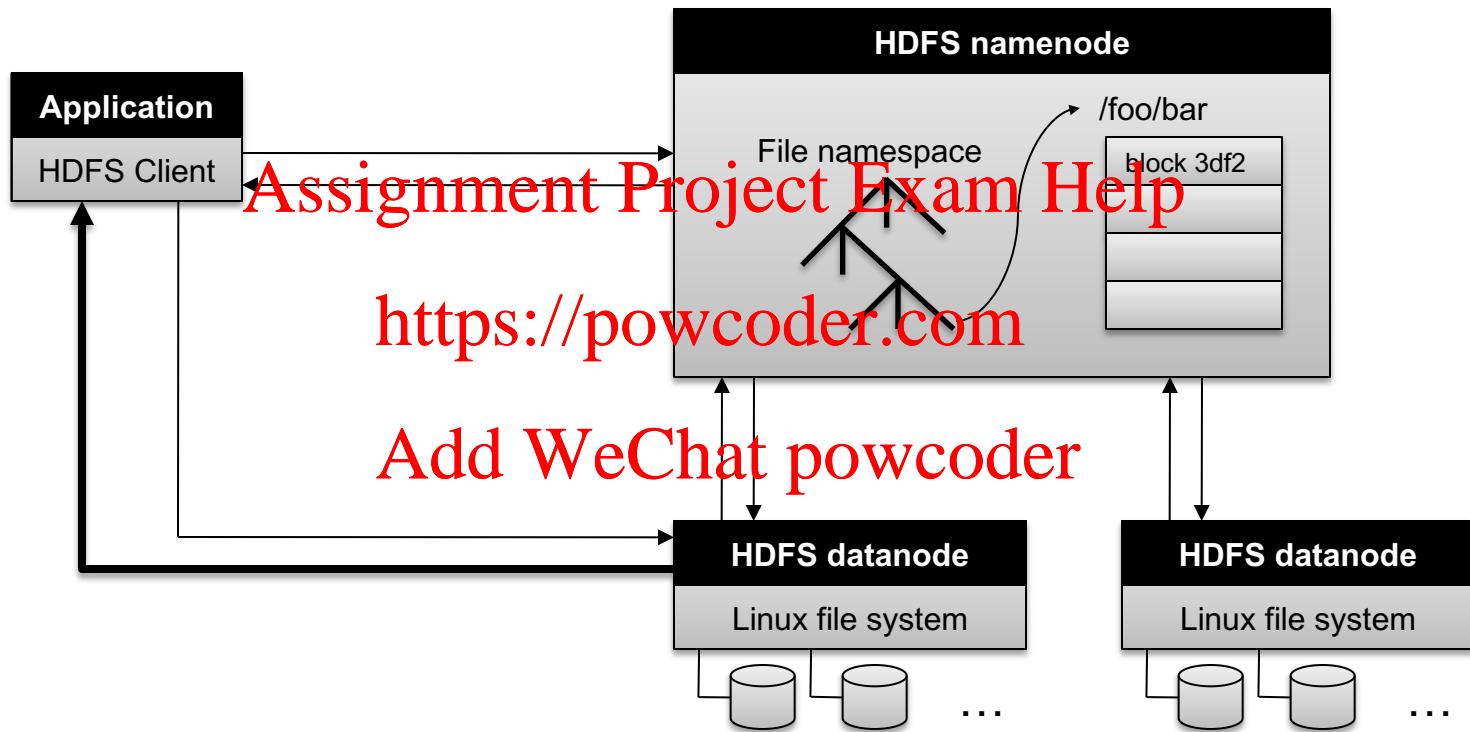
- Commodity hardware over “exotic” hardware
 - Scale “out”, not “up”
- High component failure rates
 - Inexpensive commodity components fail all the time
- “Modest” number of huge files
 - Multi-gigabyte files are common, if not encouraged
- Files are write-once, read many
 - Perhaps concurrently
- Large streaming reads over random access
 - High sustained throughput over low latency

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

HDFS Architecture



How HDFS works

- When an input file is added to HDFS
 - File is split into smaller blocks of fixed size
 - Each block is replicated
 - Each replicated block is stored on a different host
- Block size is configurable. Default is 128/256MB.
- Replication level is configurable. Default is 3
 - Replication is necessary for
 - Scaling
 - High Availability
- In case a host crashes or is removed
 - All blocks on that host are automatically replicated to other hosts
- In case a host is added
 - Blocks will be rebalanced so that some blocks from other hosts will be placed on the new host

HDFS Component Responsibilities

- Name Node

- Managing the file system namespace:
 - Holds file/directory structure, metadata, file-to-block mapping, access permissions, etc.
 - Coordinating file operations:
 - Directs clients to datanodes for reads and writes
 - No data is moved through the Namenode
 - Maintaining overall health:
 - Periodic communication with the datanodes
 - Block re-replication and rebalancing
 - Garbage collection

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

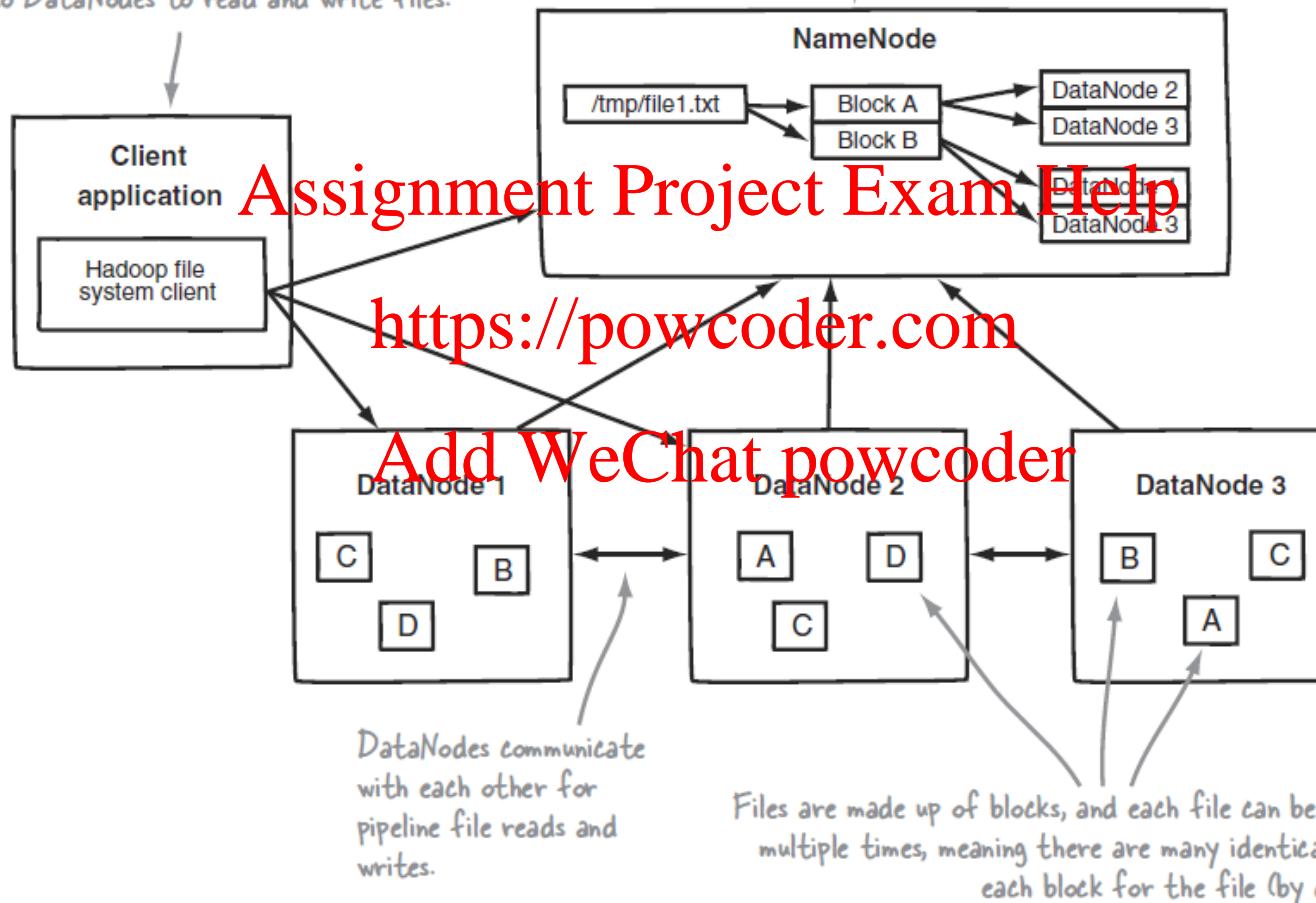
- Data Node

- Actual storage and management of data block on a single host
 - Provides clients with access to data

HDFS

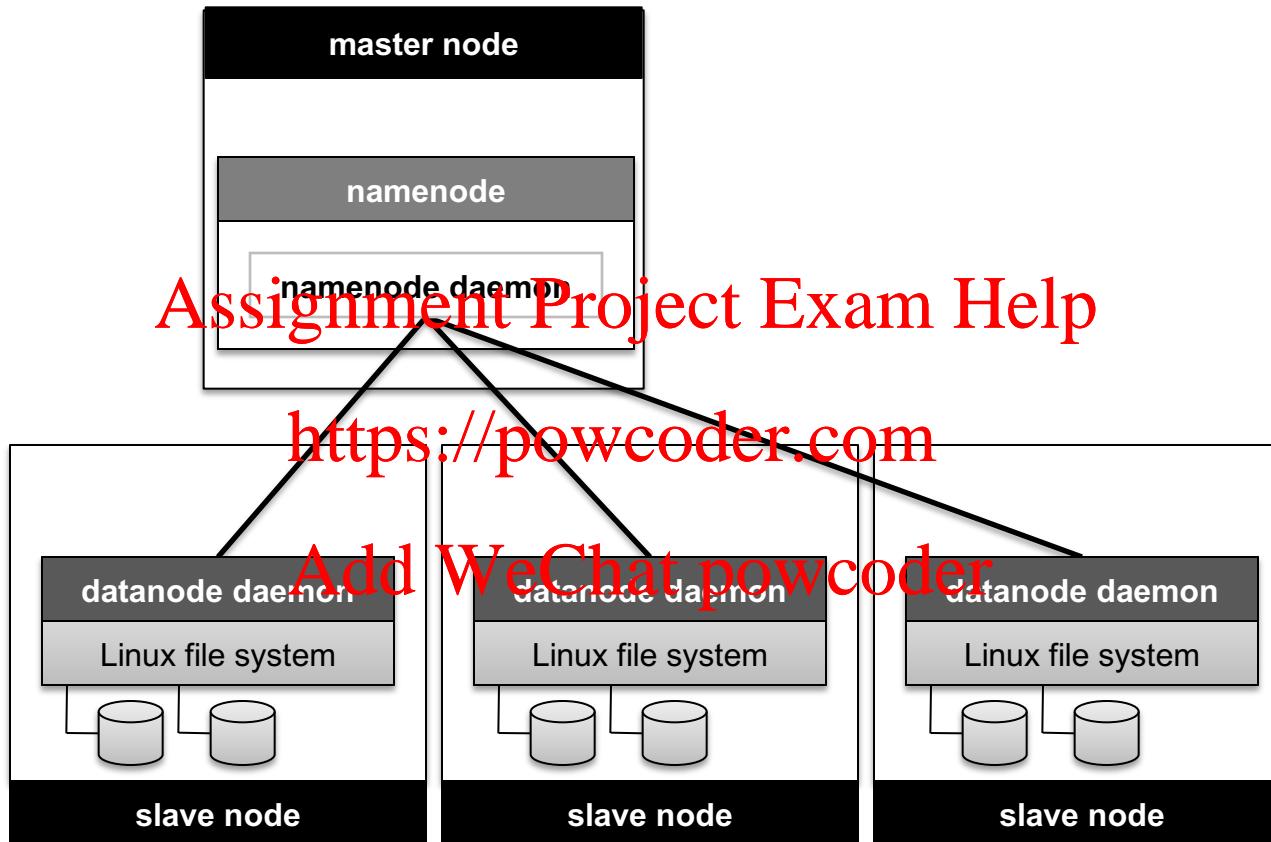
HDFS clients talk to the NameNode for metadata-related activities, and to DataNodes to read and write files.

The HDFS NameNode keeps in memory the metadata about the filesystem, such as which DataNodes manage the blocks for each file.



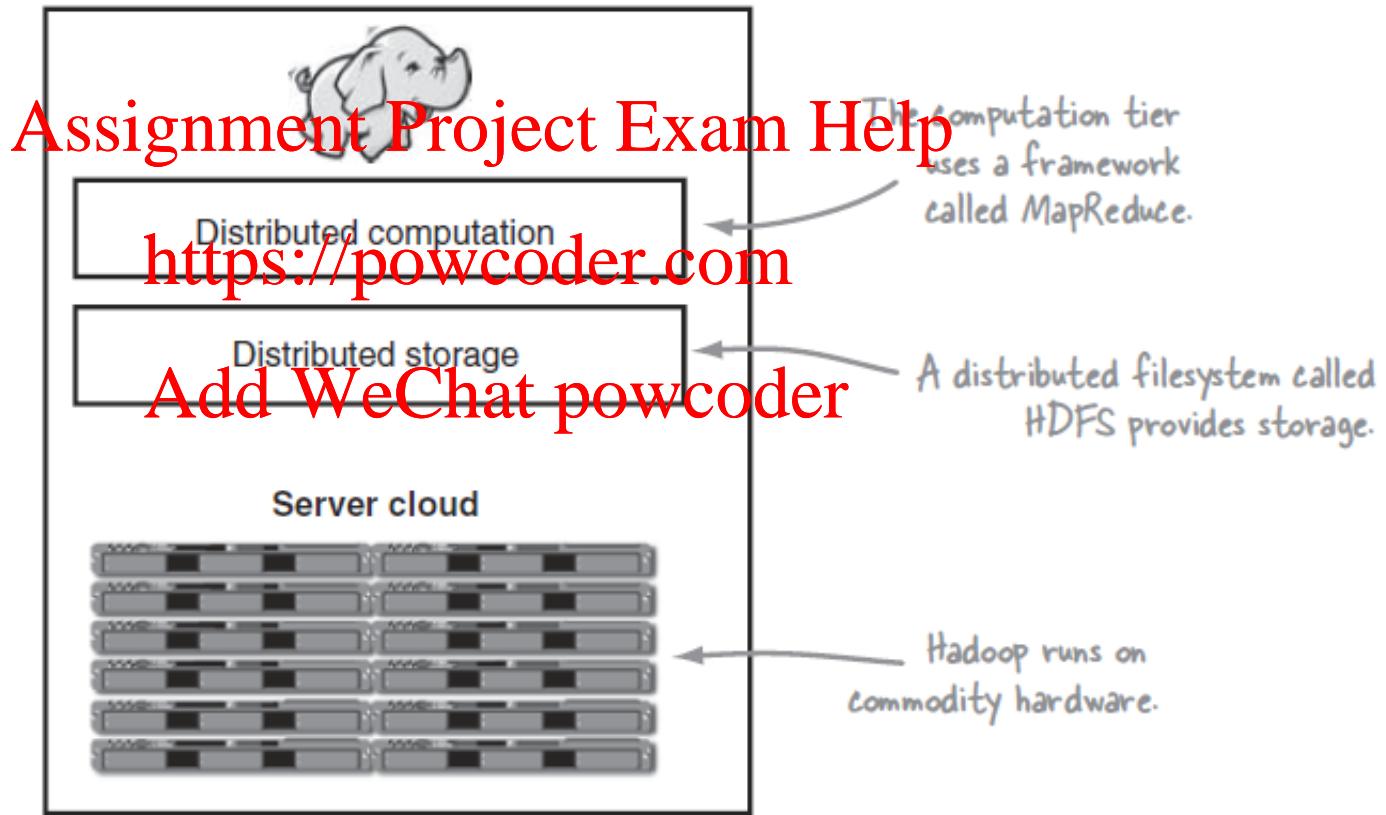
HDFS architecture shows an HDFS client communicating with the master NameNode

HDFS Components in Cluster



Hadoop

- Platform for distributed **storage** and **computation**
 - HDFS
 - MapReduce
 - Ecosystem



Source: **Hadoop in Practice**, Alex Holmes, Manning Publications Co., 2012

MapReduce (MR) can refer to...

- The execution framework (aka “runtime”)
- The programming model
- The specific implementation

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Usage is usually clear from context!

MR Framework Components

- Job Tracker
 - Central component responsible for managing job lifecycles
 - One Job Tracker per MR framework instance
 - Accepts job submissions, queries etc. from clients
 - Enqueues jobs and schedules individual tasks.
 - Communicates with Task Trackers to deploy and run tasks
 - Attempts to assign tasks to support Data Locality.

Add WeChat powcoder

- Task Tracker
 - One Task Tracker per host
 - Runs and manages individual tasks
 - Communicates progress of tasks back to Job Tracker.

MR Programming Model

- Programmers specify two functions:
 $\text{map } (k, v) \rightarrow \langle k', v' \rangle^*$
 $\text{reduce } (k', v') \rightarrow \langle k', v' \rangle^*$
 - All values with the same key are sent to the same reducer
- The MR Execution framework handles everything else...

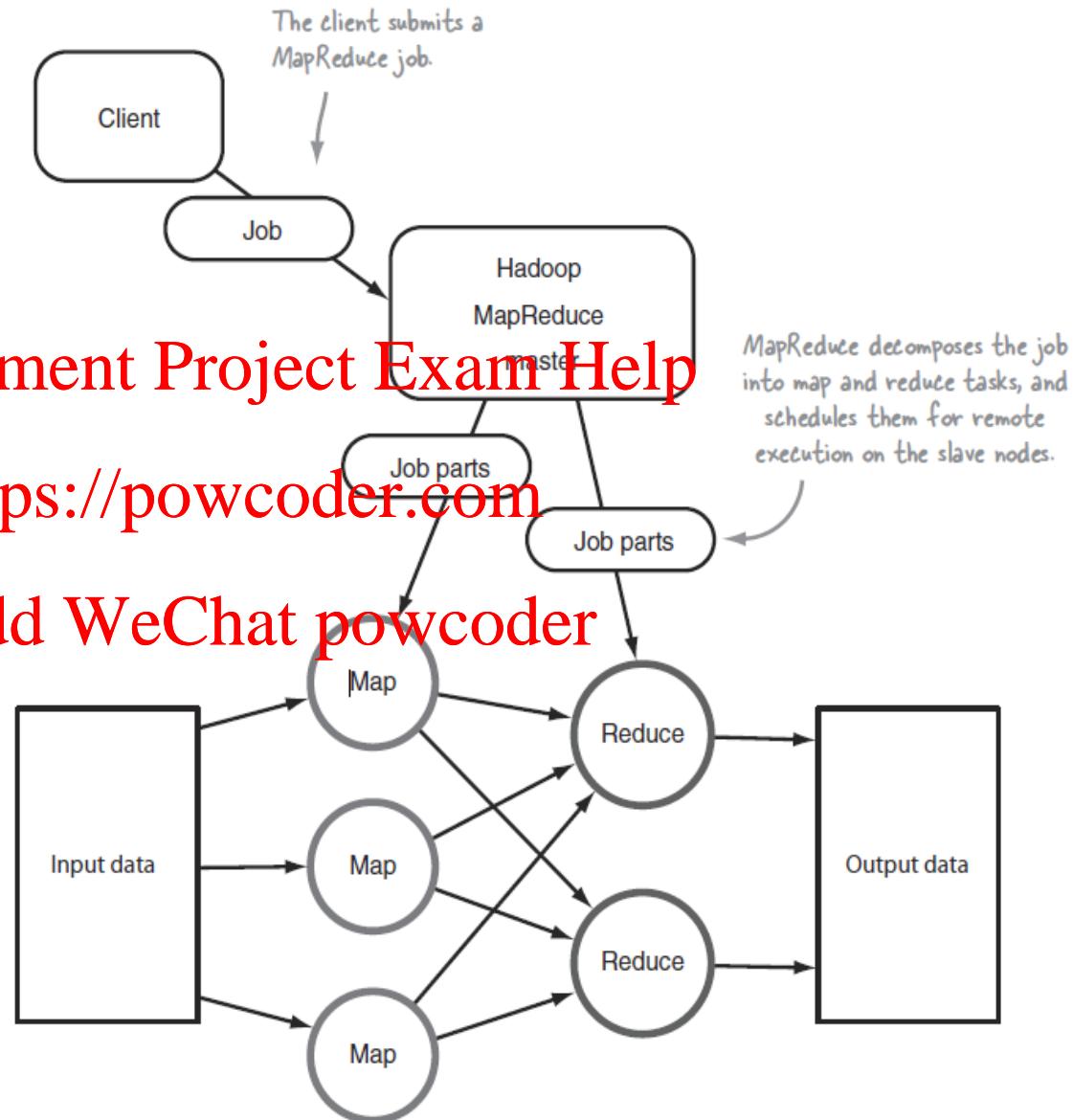
<http://powcoder.com> “What’s everything else”?

Add WeChat powcoder

MapReduce

■ Everything Else

- Handles scheduling
 - Assigns workers to map and reduce tasks
- Handles “data distribution”
 - Moves processes to data
- Handles synchronization
 - Gathers, sorts, and shuffles intermediate data
- Handles errors and faults
 - Detects worker failures and restarts
- Everything happens on top of a distributed FS (HDFS)



Our Scoring Algorithm as a Map Reduce Program

```
//MAPPER
public static class Map
    extends MapReduceBase
    implements Mapper<LongWritable, Text, Text, LongWritable> {

    private Scorer scorer = new MyScorer();

    public void map(
        LongWritable key, Text value, //map input
        OutputCollector<Text, LongWritable> output, //map output <key,value>
        Reporter reporter) throws IOException {
        String line = value.toString();
        output.collect(mykey, new LongWritable(scorer.getScore(line)));
    }
}

//REDUCER
public static class Reduce
    extends MapReduceBase
    implements Reducer<Text, LongWritable, Text, LongWritable> {

    public void reduce(
        Text key, Iterator<LongWritable> values, //reducer input
        OutputCollector<Text, LongWritable> output, //reducer output
        Reporter reporter) throws IOException {
        long sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
            output.collect(mykey, new LongWritable(sum));
        }
    }
}
```

Assignment Project Exam Help

<https://powcoder.com>

Our Analytic

Add WeChat powcoder

Basic Hadoop API*

- Mapper

- void map(K1 key, V1 value, OutputCollector<K2, V2> output, Reporter reporter)
- void configure(JobConf job)
- void close() throws IOException

- Reducer/Combiner

<https://powcoder.com>

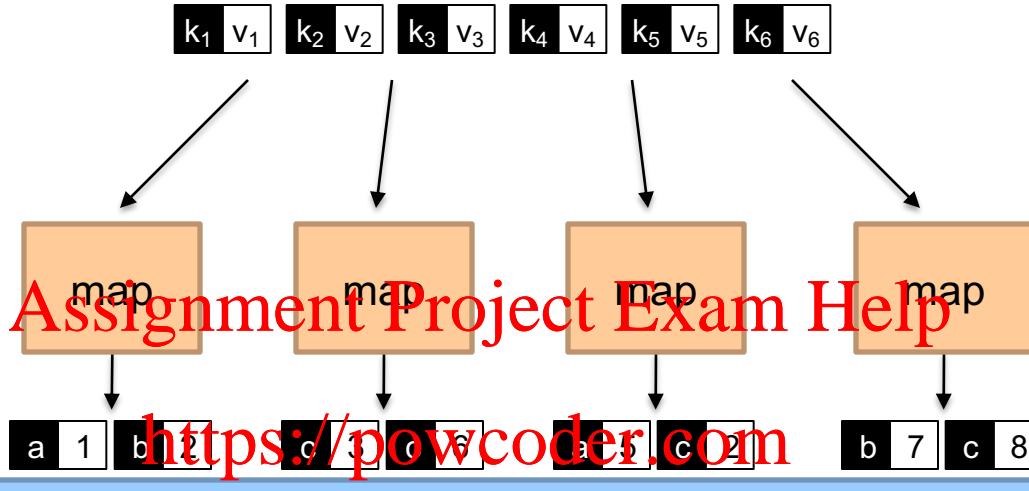
- void reduce(K2 key, Iterator<V2> values, OutputCollector<K3,V3> output, Reporter reporter)
- void configure(JobConf job)
- void close() throws IOException

- Partitioner

- void getPartition(K2 key, V2 value, int numPartitions)

Assignment Project Exam Help

Add WeChat powcoder



Assignment Project Exam Help

<https://powcoder.com>

Shuffle and Sort: aggregate values by keys

Add WeChat [powcoder](https://powcoder.com)

reduce

reduce

reduce

$r_1 s_1$

$r_2 s_2$

$r_3 s_3$

Lets Talk Numbers

- How many mappers?
 - Depends on the size of input data
 - Typically 1 mapper per data block
 - So 1 GB input data will have around 8 Mappers
 - Assuming 128MB block size
- How many reducers?
 - Depends on cluster reducer capacity
 - Can be set depending on the expected number of keys
 - For large data sets, set it to cluster reducer capacity

MapReduce

- Programmers specify two functions:
 $\text{map } (k, v) \rightarrow \langle k', v' \rangle^*$
 $\text{reduce } (k', v') \rightarrow \langle k', v' \rangle^*$
 - All values with the same key are reduced together
- The execution framework handles everything else...
- Not quite... usually programmers also specify:
 $\text{combine } (k', v') \rightarrow \langle k', v' \rangle^*$
 - Mini-reducers that run in memory after the map phase
 - Used as an optimization to reduce network traffic $\text{partition } (k', \text{number of partitions}) \rightarrow \text{partition for } k'$
 - Often a simple hash of the key, e.g., $\text{hash}(k') \bmod n$
 - Divides up key space for parallel reduce operations

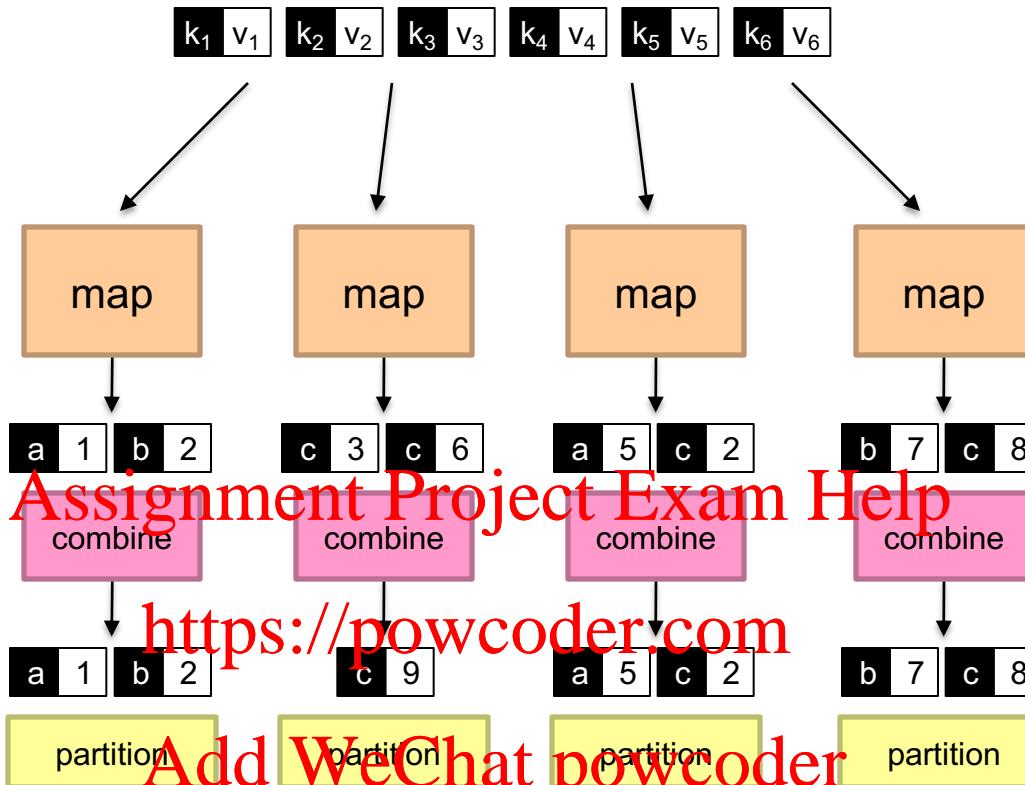
Two more details...

- Barrier between map and reduce phases
 - But we can begin copying intermediate data earlier
- Keys arrive at each reducer in sorted order
 - No enforced ordering across reducers

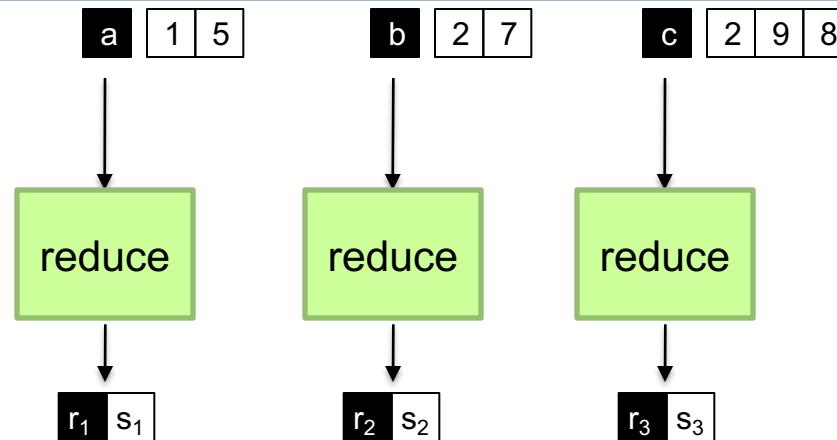
Assignment Project Exam Help

<https://powcoder.com>

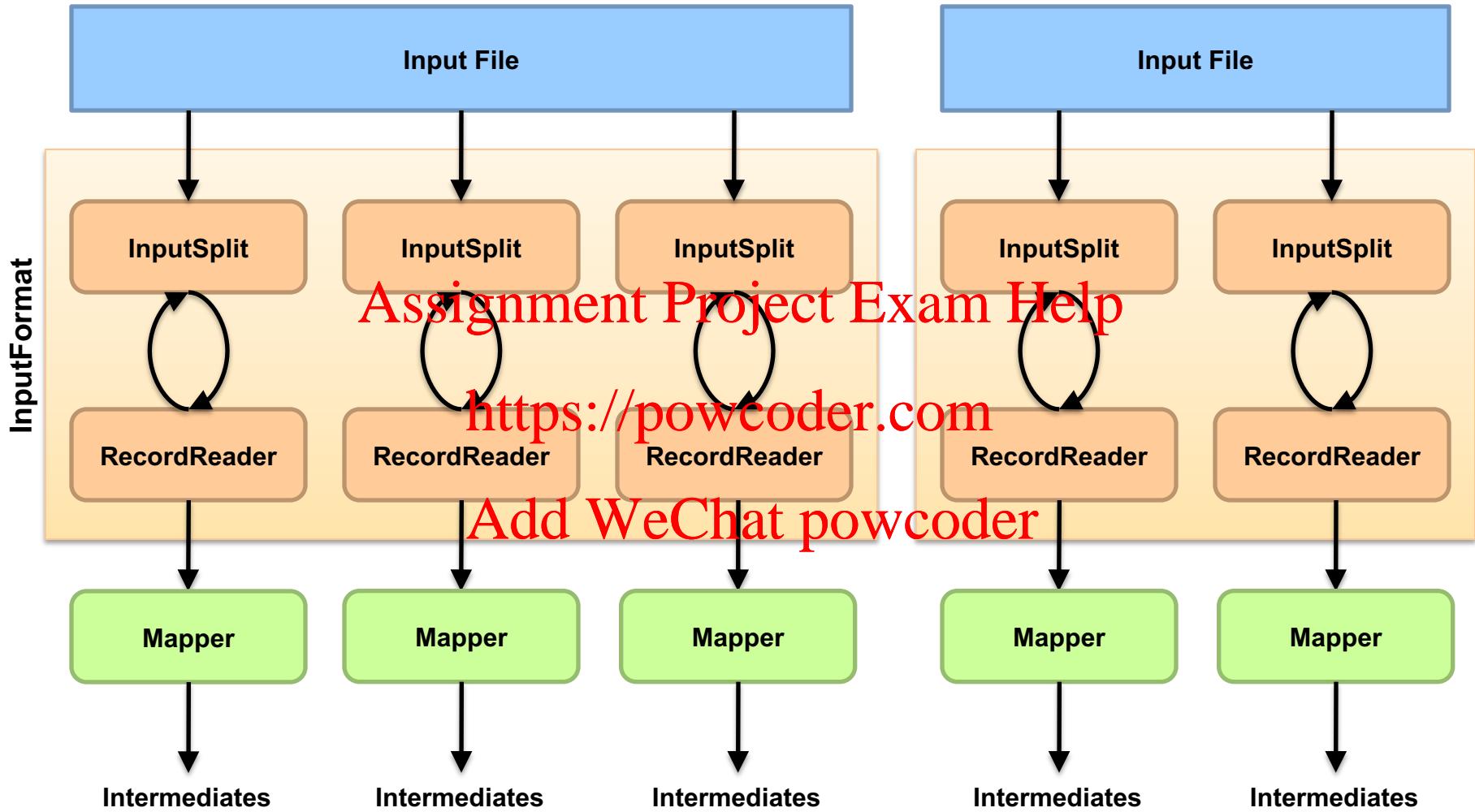
Add WeChat powcoder



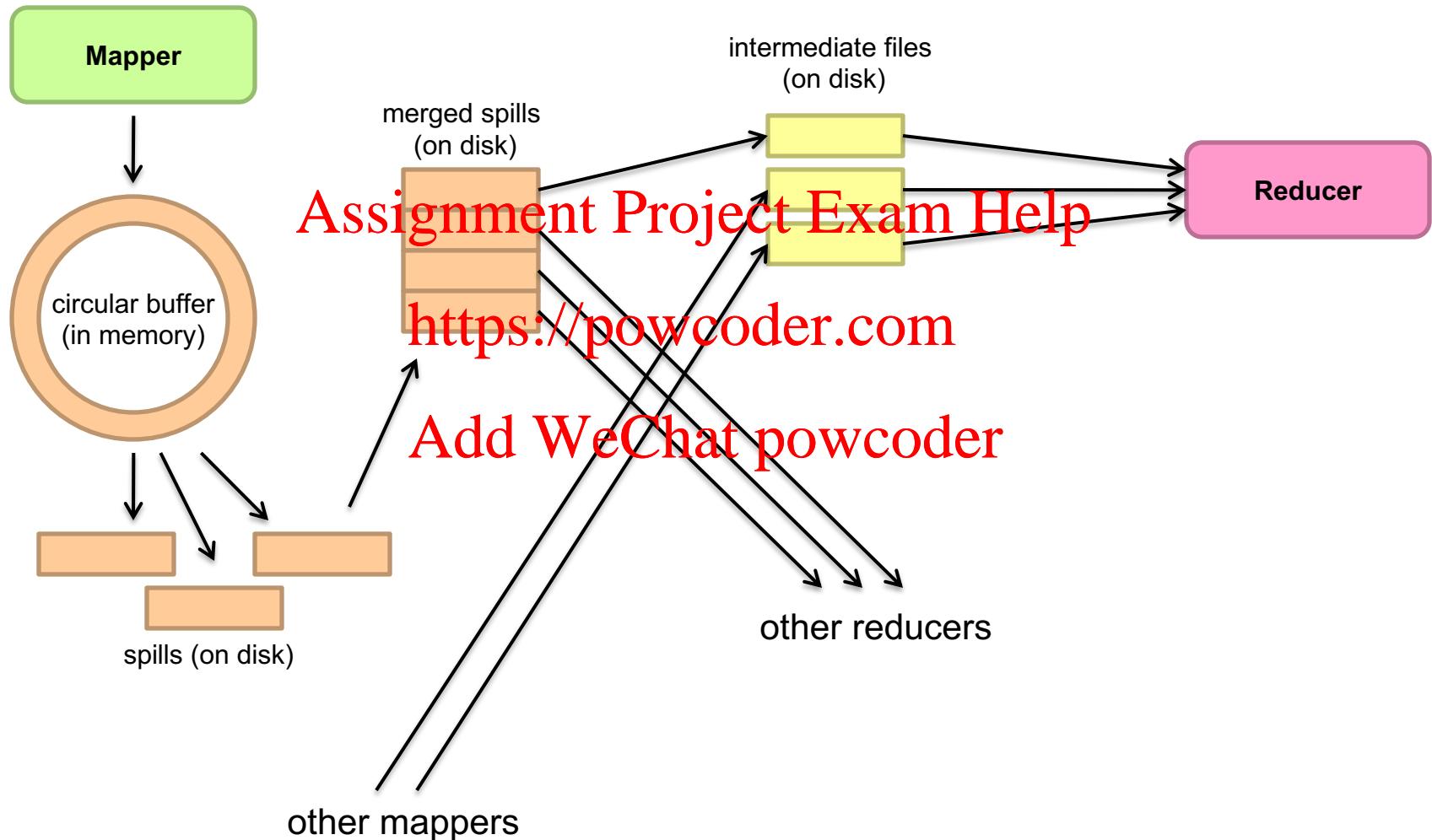
Shuffle and Sort: aggregate values by keys



Input To Mappers

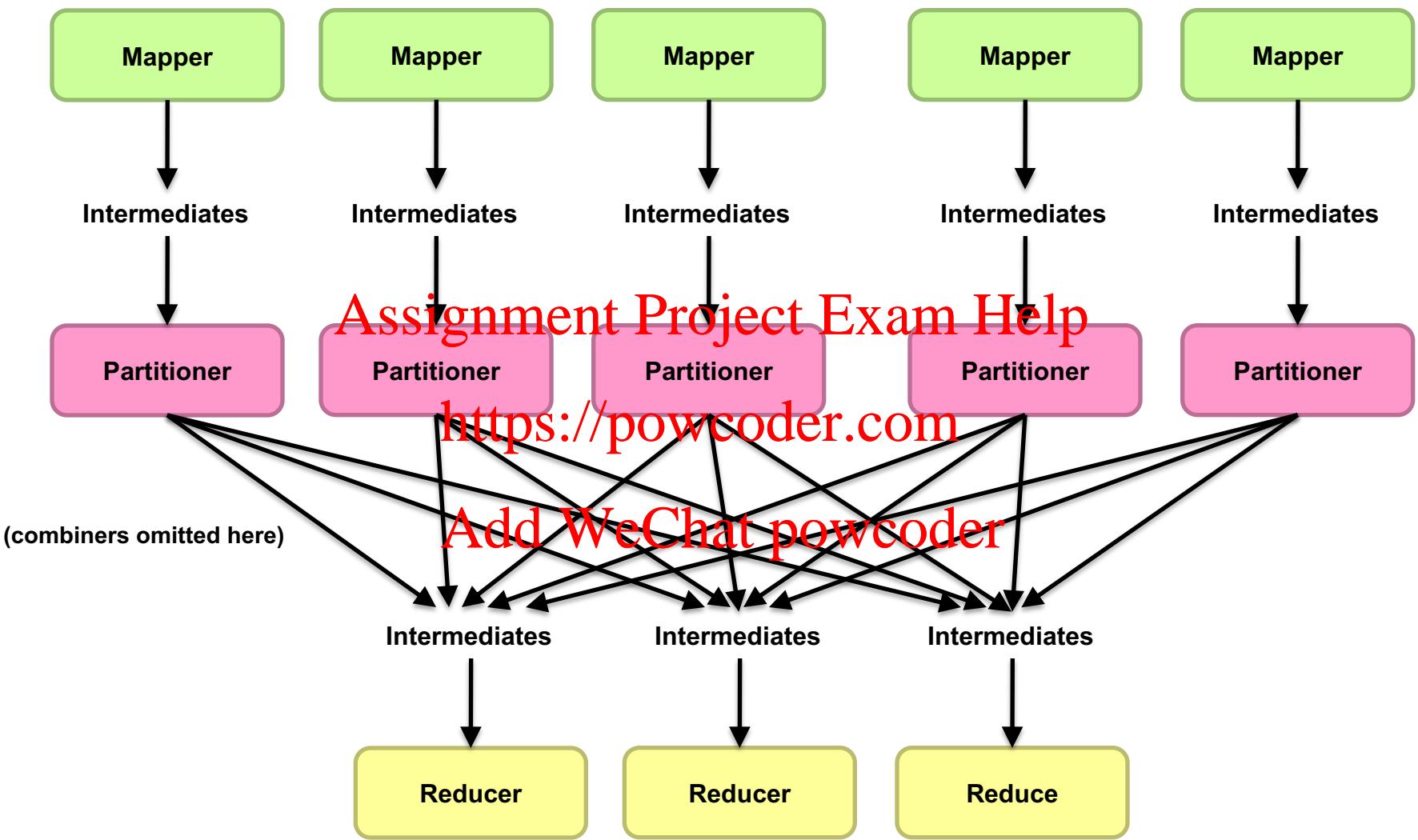


Shuffle and Sort

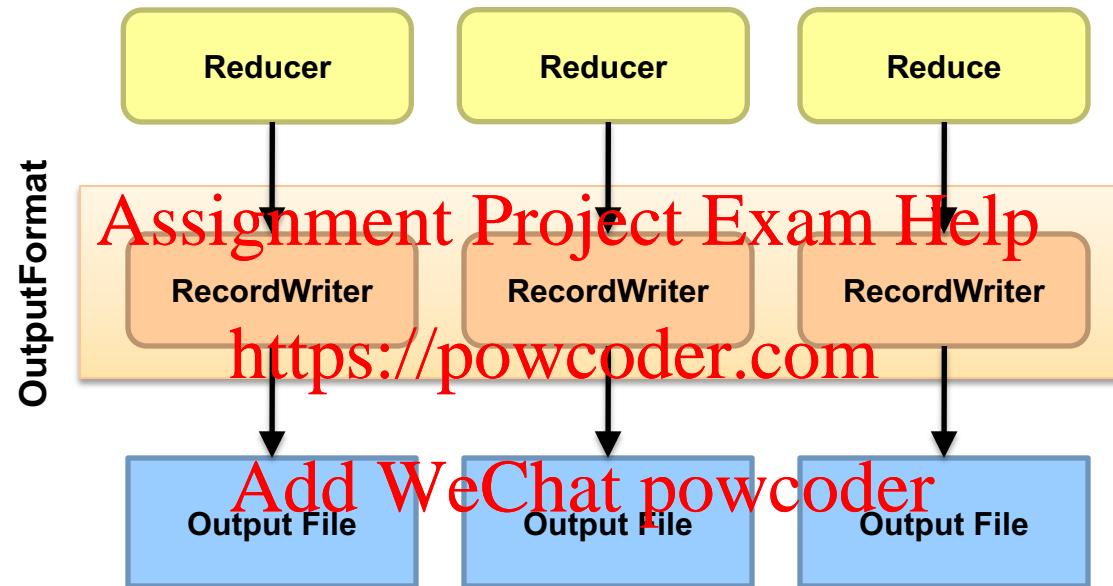


Shuffle and Sort in Hadoop

- Probably the most complex aspect of MapReduce!
- Map side
 - Map outputs are buffered in memory in a circular buffer
 - When buffer reaches threshold, contents are “spilled” to disk
 - Spills merged in a single, partitioned file (sorted within each partition): combiner runs here
<https://powcoder.com>
- Reduce side Add WeChat powcoder
 - First, map outputs are copied over to reducer machine
 - “Sort” is a multi-pass merge of map outputs (happens in memory and on disk): combiner runs here
 - Final merge pass goes directly into reducer



Reducer to Output



Input and Output

- InputFormat:

- TextInputFormat
- KeyValueTextInputFormat
- SequenceFileInputFormat
- ...

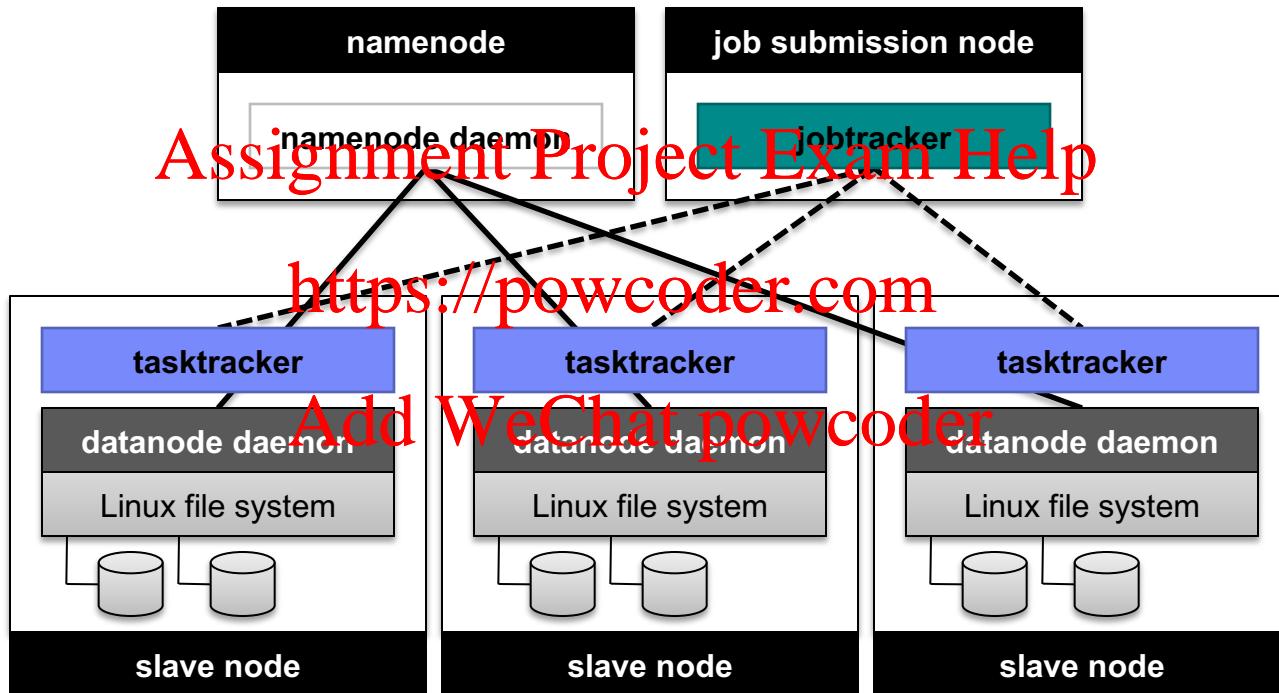
Assignment Project Exam Help

- OutputFormat: <https://powcoder.com>

- TextOutputFormat
- SequenceFileOutputFormat
- ...

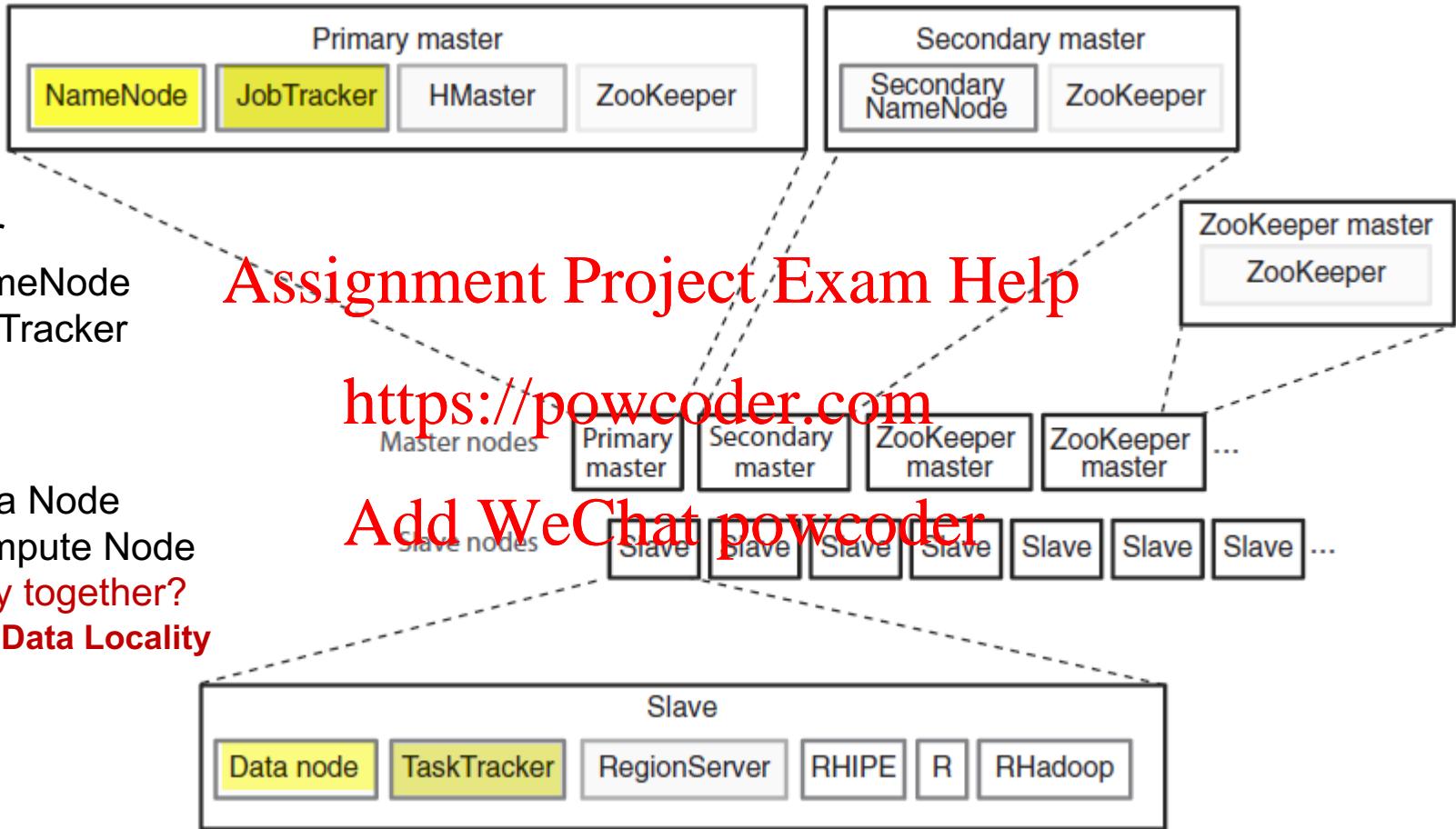
Add WeChat powcoder

Putting everything together...



HADOOP Architecture

- Master
 - NameNode
 - JobTracker
- Slaves
 - Data Node
 - Compute Node
 - Why together?
 - Data Locality



One More Thing

- Distributed Cache
 - Usually used for files of small size
 - Provides a convenient way to propagate applications and configuration files
 - HDFS is not used handle such files due to their small size
 - Shared across all nodes in the MapReduce cluster

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Dizzy Yet?

- OK, we went through a lot of details
- Whatever happened to the simplicity of programming??
- Do I really have to write a MapReduce program every time I want to run a new analytic?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

We went from..

Multi-Threaded

```
public static long scoreDocument(String fileName, Scorer scorer, int threads)
    throws Exception {
    BufferedReader reader =
        new BufferedReader(new FileReader(fileName));

    //thread safe structures
    BlockingQueue<String> queue =
        new LinkedBlockingQueue<String>(threads + 100);

    //initialize and start threads
    AtomicInteger completionCounter = new AtomicInteger();
    ExecutorService executors = Executors.newFixedThreadPool(threads);
    List<CounterThread> counters = new ArrayList<CounterThread>();
    for(int i=0;i<threads; i++) {
        executors.execute(new CounterThread(queue, scorer, completionCounter));
    }
    String line = null;
    while((line = reader.readLine()) != null) {
        queue.put(line);
    }

    //terminating condition for threads
    for(int i=0;i<threads; i++)
        queue.put("EXIT");
    while(completionCounter.intValue() < threads) {
        Thread.sleep(100);
    }

    executors.shutdown();
    reader.close();

    //summarize results
    long total = 0;
    for(CounterThread counter : counters)
        total+=counter.getTotal();

    return total;
}
```

Map-Reduce

```
//MAPPER
public static class Map
    extends MapReduceBase
    implements Mapper<LongWritable, Text, Text, LongWritable> {
    private Scorer scorer = new MyScorer();

    public void map(
        LongWritable key, Text value, //map input
        OutputCollector<Text, LongWritable> output, //map output <key, value>
        Reporter reporter) throws IOException {
        String line = value.toString();
        output.collect(mykey, new LongWritable(scorer.getScore(line)));
    }
}

//REDUCER
public static class Reduce
    extends MapReduceBase
    implements Reducer<Text, LongWritable, Text, LongWritable> {
    public void reduce(
        Text key, Iterator<LongWritable> values, //reducer input
        OutputCollector<Text, LongWritable> output, //reducer output
        Reporter reporter) throws IOException {
        long sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(mykey, new LongWritable(sum));
    }
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Enter PIG ... Oink!

- High Level Languages for Map-Reduce

- PIG
 - Developed by Yahoo
 - HIVE
 - Developed by Facebook
 - JAQL
 - Developed by IBM

Assignment Project Exam Help

- All of these languages provide similar functionality

https://powcoder.com
Add WeChat powcoder

- All of them allow users to plug in their own user defined functions (UDFs)

Lets get Practical – From Setup to Results

Setting up a Hadoop Cluster

- Minimum recommended configuration (4 Hosts)
 - 1 Host Dedicated for Management Services (Job Tracker, Name Node etc)
 - 3 Hosts as Slave nodes (Data Node , Task Trackers)

Assignment Project Exam Help

- Data nodes should have high capacity local disks attached.
 - This is where all your data is going to be <https://powcoder.com>
- How much total disk space?
 - Depends on input data to be processed
 - Effective Storage Space Recommended: Typically 3 times your input data size
 - Actual Storage Space: Effective Storage Space * 3 (replication level)
- Single node installation is fine for development/testing on very small data
 - Perhaps not the best for testing performance
- Installation instructions vary from provider to provider

Add WeChat powcoder

Some cluster configuration parameters

- HDFS configuration parameters
 - Stored in hdfs-site.xml
 - Block size
 - Default replication count

Assignment Project Exam Help

<https://powcoder.com>

- MapReduce configuration parameters
 - Stored In “mapred-site.xml”
 - Java heap size for mappers/reducers
 - Number of mappers/reducers per host
 - See <http://wiki.apache.org/hadoop/HowManyMapsAndReduces>

Add WeChat powcoder

- ***IMPORTANT***
 - Job Tracker URL: `http://<masterhost>:50030`
 - Name Node URL: `http://<masterhost>:50070`

Job Tracker Web Page (port 50030)

tdma1 Hadoop Map/Reduce Administration

Quick Links

State: RUNNING

Started: Tue Feb 25 15:21:13 EST 2014

Version: 1.1.1, r70b5aad8822a30795c1acdb966c97316387e1fc0

Compiled: Thu May 30 17:51:51 PDT 2013 by jenkins

Identifier: 201402251521

SafeMode: OFF

Assignment Project Exam Help

Cluster Summary (Heap Size is 298.91 MB/2.09 GB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes	Excluded Nodes
0	0	1021	4	0	0	0	0	128	64	48.00	0	0	0

Add WeChat [powcoder](https://powcoder.com)

Working with data

- Lets say you have 1 GB of data in your local filesystem (mydata.txt)
- Load into HDFS
 - hadoop fs –mkdir /path/mydirectory
 - hadoop fs –put mydata.txt /path/mydirectory
 - where /path/mydirectory is in HDFS
- List the file you just uploaded
 - hadoop fs –ls /path/mydirectory
- “hadoop fs” works similar to linux filesystem commands
 - However HDFS is not POSIX compliant.
 - It cannot be mounted as a regular filesystem

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Writing your program .. see the simplicity!!

- JAQL program for running our scorer

```
// import the module  
import scorerModule(*);  
  
//get the total score  
mytotal =  
    read(lines("path/to/your/myfile.txt"))  
    -> transform score($)  
    -> sum();
```

Assignment Project Exam Help
<https://powcoder.com>

- PIG program for running our scorer

register 'myudfs.jar'; Add WeChat powcoder

```
//read data  
mydata = load '/path/mydirectory/mydata.txt'  
    using TextLoader() as (line : chararray);  
  
//score each line  
mylines = foreach mydata generate myudfs.score(line) as scorecount;  
  
//find total score  
alllines = group mylines all;  
mytotal = foreach alllines generate SUM(mylines.scorecount);
```

All languages provide similar functionality

- LOAD (various data formats)
- JOIN
- FOR-EACH
- GROUP
- SORT
- FILTER
- Pluggable UDFs

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Hadoop Programming Tips

- Thinking at scale
 - Filter unwanted data earlier in the flow
 - Store intermediate data
 - Use “sequence” format for storing data.

Assignment Project Exam Help

- These are not iterative languages
 - i.e. No *for* or *while* loops
- Watch out for obvious bottlenecks
 - Single key for all mapper output will send data to one reducer
 - Too much data sent to a UDF will result in OOM errors

<https://powcoder.com>

Add WeChat powcoder

Submitting a Job

- Create and save your PIG script (myscript.pig)
- To deploy (pig command will be in your installation)
 - pig –f myscript.pig
 - Command will complete once your job completes

Assignment Project Exam Help

- To check the status of your job
 - Use the Job Tracker URL (easiest) OR
 - hadoop job –list (will print all job ids)
 - hadoop job –status <jobid> (will print the job status)
- To get the results
 - hadoop fs –get /path/results.txt .

<https://powcoder.com>

Add WeChat powcoder

Anatomy of a Job

- MapReduce program in Hadoop = Hadoop job
 - Jobs are divided into map and reduce tasks
 - An instance of running a task is called a task attempt
 - Multiple jobs can be composed into a workflow
- Job submission process
 - Client (i.e., driver program) creates a job, configures it, and submits it to job tracker
 - JobClient computes input splits (on client end)
 - Job data (jar, configuration XML) are sent to JobTracker
 - JobTracker puts job data in shared location, enqueues tasks
 - TaskTrackers poll for tasks
 - Off to the races...

A simple illustration of MR process

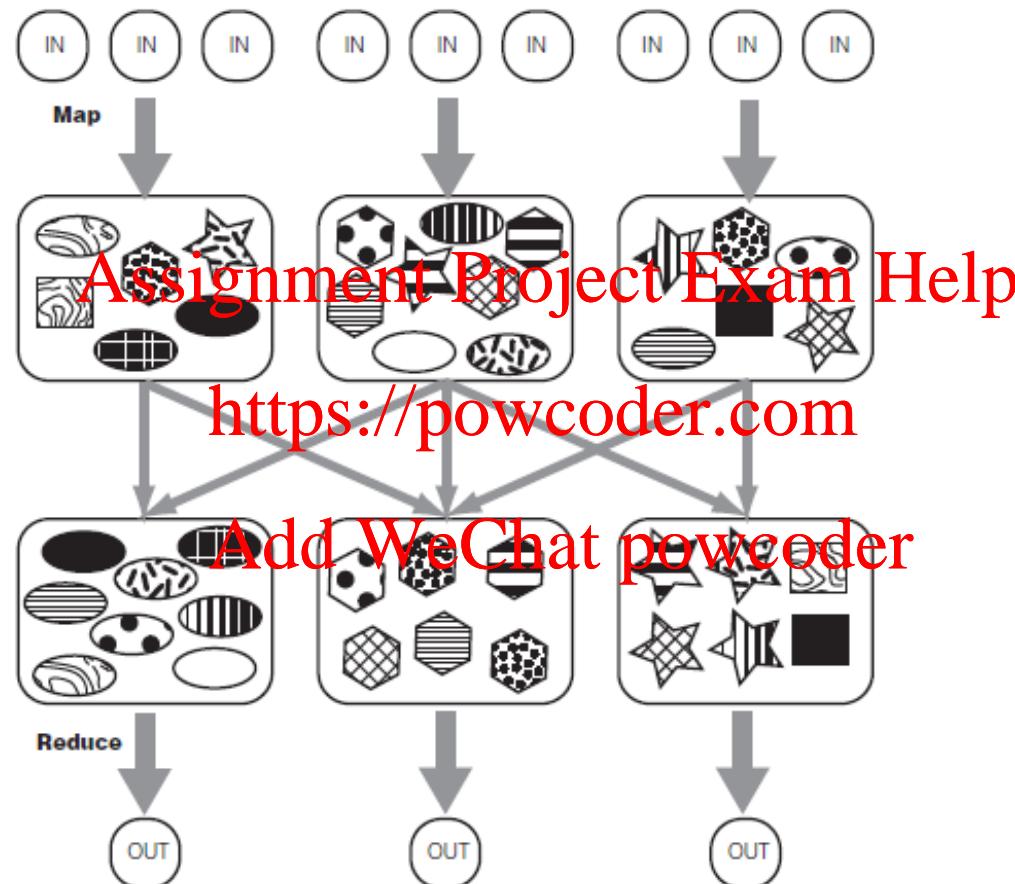
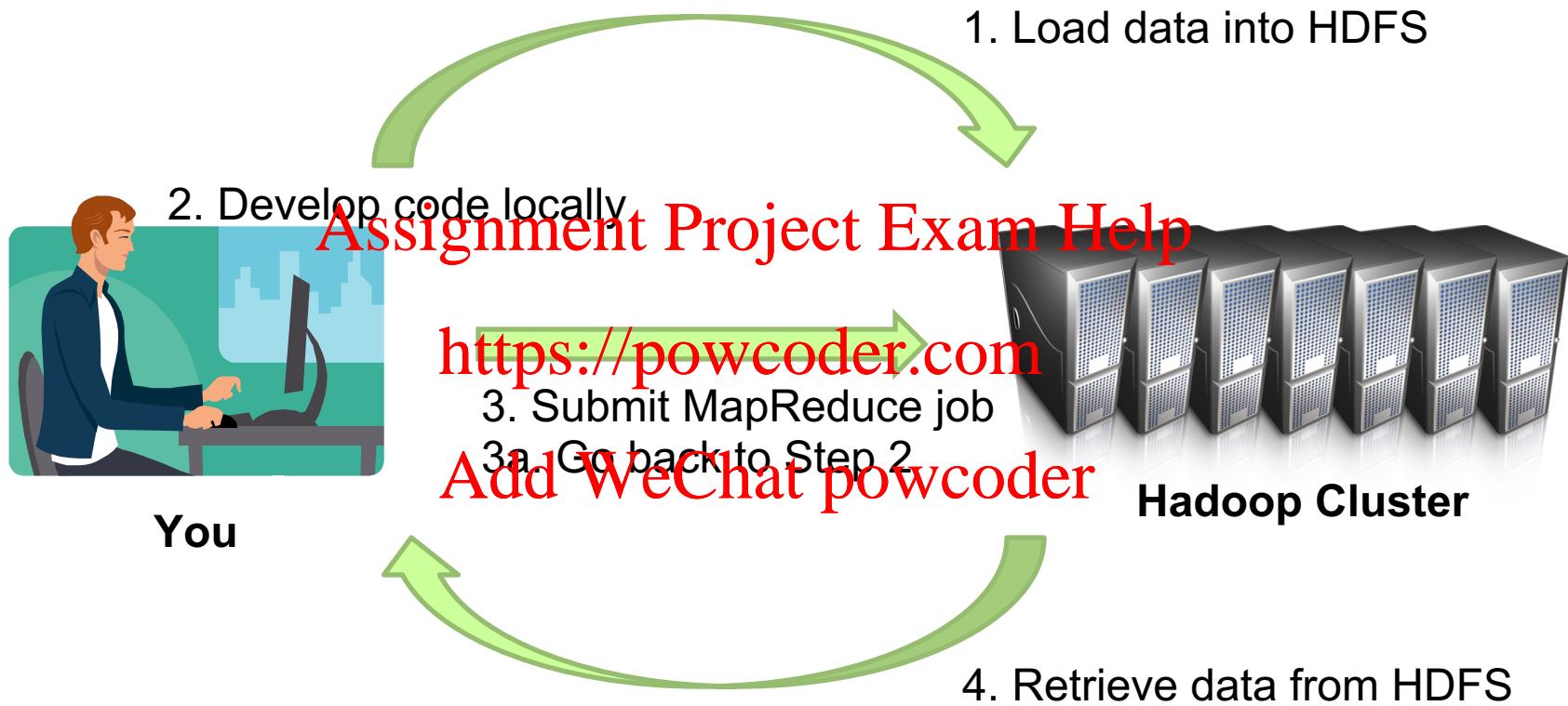
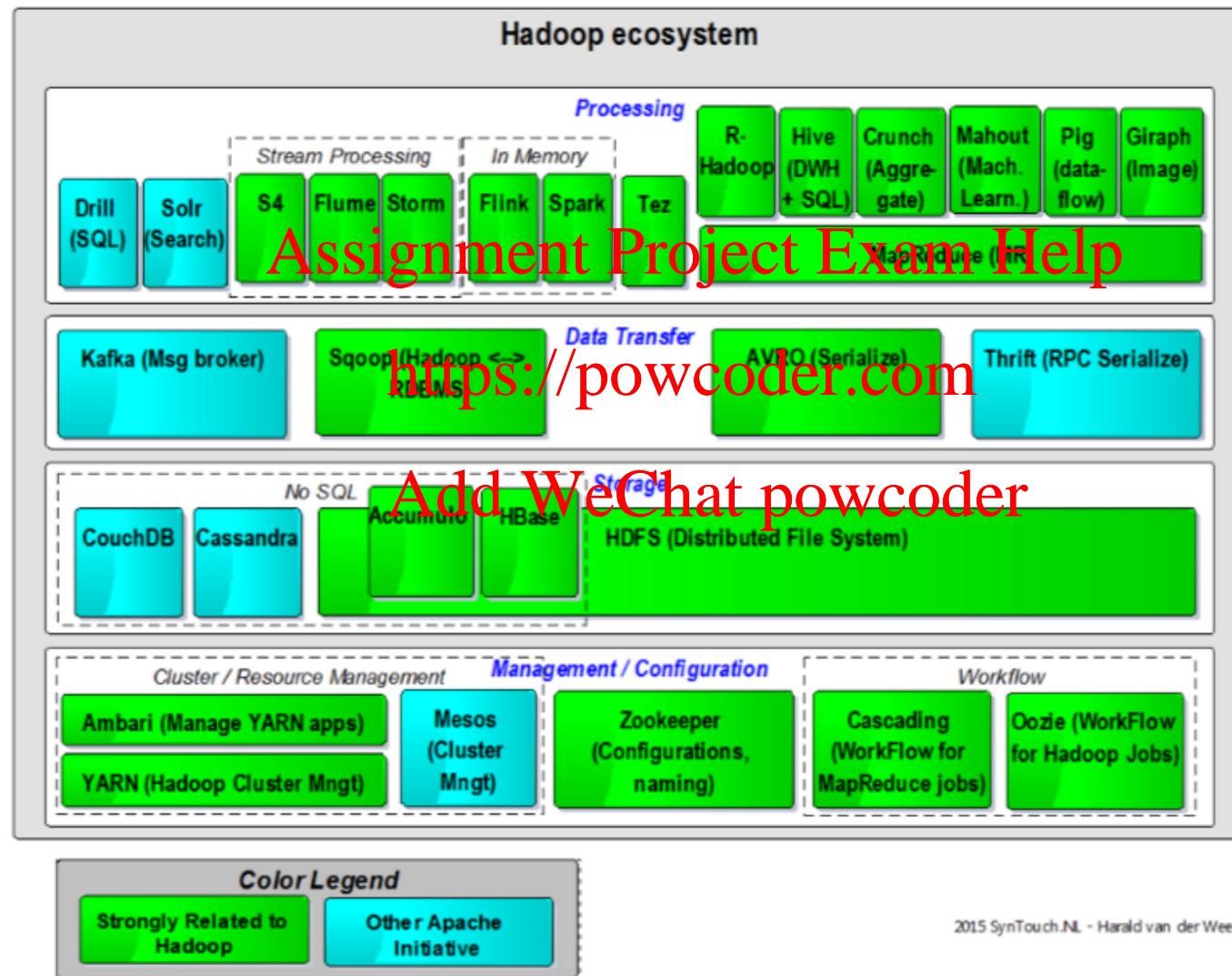


Figure 3.2 The MapReduce data flow, with an emphasis on partitioning and shuffling. Each icon is a key/value pair. The shapes represent keys, whereas the inner patterns represent values. After shuffling, all icons of the same shape (key) are in the same reducer. Different keys can go to the same reducer, as seen in the rightmost reducer. The partitioner decides which key goes where. Note that the leftmost reducer has more load due to more data under the “ellipse” key.

Hadoop Workflow



Larger View...



MR Patterns Examples

- Jimmy Lin's book
- Jeffrey Ulman's book
- An excellent blog: <https://highlyscalable.wordpress.com/2012/02/01/mapreduce-patterns/>

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Uh Oh.. My Job Failed...Now what?

- First, take a deep breath
- Start small, start locally
- Strategies
 - Learn to use the webapp
 - Where does `println` go?
<https://powcoder.com>
 - Don't use `println`, use logging
 - Throw `Runtime`~~Exception~~
~~Add WeChat powcoder~~
- Logs are most easily accessible via the Job Tracker URL

Time for a Raise

- Finally you have mastered Hadoop Big Data
- Your applications are scaling.
 - You deserve a raise!!
- Boss **Assignment Project Exam Help**
 - Can we query the data for specific entities?
<https://powcoder.com>
 - How long will that take?
- Problem **Add WeChat powcoder**
 - Remember this is still sequential access
 - To find a specific entity, you still need to read the entire data set.
- What now?
 - How is this solved in traditional systems?

Databases

Other projects based on Hadoop

- HBase
- Hive
- PIG
- Spark
- Mahout

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Hive – a SQL-like data warehouse on Hadoop

<https://cwiki.apache.org/confluence/display/Hive/Tutorial>

- Supports a SQL-like data warehouse on top of Hadoop
 - began at Facebook
- Provides SQL users the capability of big data without requiring lower level programming for a wide range of tasks
- Fewer lines of code!
- /bin/hive –help

Assignment Project Exam Help

```
hive> CREATE TABLE cite (citing INT, cited INT)
      > ROW FORMAT DELIMITED
      > FIELDS TERMINATED BY ','
      > STORED AS TEXTFILE;
OKAdd WeChat powcoder
Time taken: 0.246 seconds
```

```
hive> LOAD DATA LOCAL INPATH 'cite75_99.txt'
      > OVERWRITE INTO TABLE cite;
Copying data from file:/root/cite75_99.txt
Loading data to table cite
OK
Time taken: 9.51 seconds
```

Wordcount Example

```
public class WordCount {  
  
    public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {  
        private final static IntWritable one = new IntWritable(1);  
        private Text word = new Text();  
  
        public void map(LongWritable key, Text value, Context context)  
            throws IOException, InterruptedException {  
            String line = value.toString();  
            StringTokenizer tokenizer = new StringTokenizer(line);  
            while (tokenizer.hasMoreTokens()) {  
                word.set(tokenizer.nextToken());  
                context.write(word, one);  
            }  
        }  
    }  
  
    public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {  
  
        public void reduce(Text key, Iterable<IntWritable> values, Context context)  
            throws IOException, InterruptedException {  
            int sum = 0;  
            for (IntWritable val : values) {  
                sum += val.get();  
            }  
            context.write(key, new IntWritable(sum));  
        }  
    }  
}  
  
}  
  
}  
  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

CREATE TABLE docs (line STRING);

LOAD DATA INPATH 'docs' OVERWRITE INTO TABLE docs;

CREATE TABLE word_counts AS
SELECT word, count(1) AS count FROM
 (SELECT explode(split(line, '\s')) AS word FROM docs) W
GROUP BY word
ORDER BY word;

Hive – SQL-like data warehouse on top of Hadoop

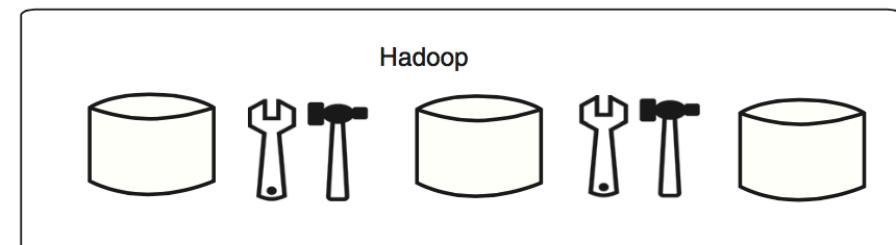
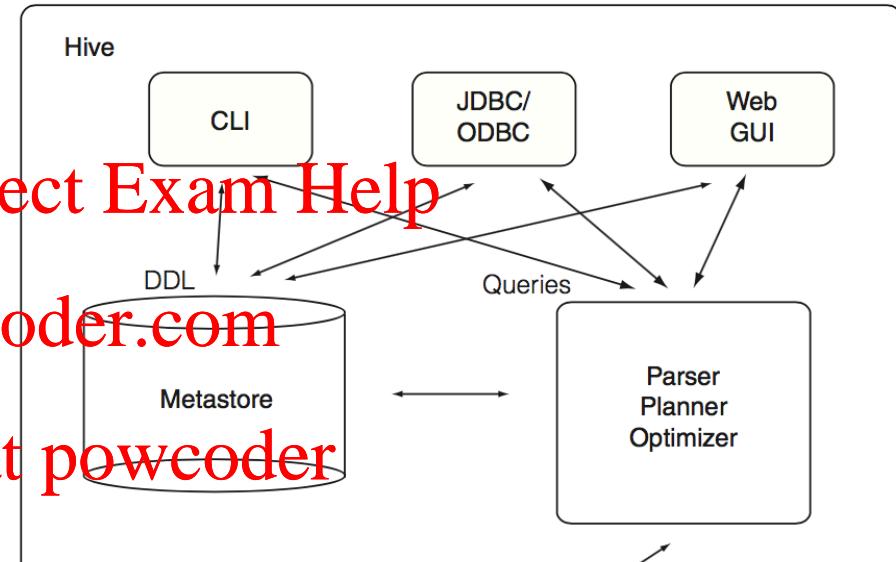
- A data warehouse with SQL like (HiveQL) interface on Hadoop for processing and managing structured data
- Hive hides the complexity of MR programming details from user with interest for processing structured data

```
INSERT OVERWRITE TABLE user_active  
SELECT user.*  
FROM user  
WHERE user.active =1;
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



- Key components
 - Metastore
 - Parser/planner/optimizer
 - Interface

Hive Installation and config

- Just download recent Hive tarball and extract it.
- Need to set up few directories
 - bin/hadoop fs -mkdir /tmp
 - bin/hadoop fs -mkdir /user/hive/warehouse
- Hive manages all the data under this directory
- Hive stores metadata in a standard relational database

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Hive examples

- 11.1.2

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Data model – column partition

- Tables as fundamental data model – stored under /usr/hive/warehouse
- Hive uses concept of partition columns – partitions data by column index, i.e., state column, date column partition..

/user/hive/warehouse/users/date=20090901/state=CA

/user/hive/warehouse/users/date=20090901/state=NY

/user/hive/warehouse/users/date=20090901/state=TX ...

Assignment Project Exam Help

/user/hive/warehouse/users/date=20090902/state=CA

/user/hive/warehouse/users/date=20090902/state=NY

/user/hive/warehouse/users/date=20090902/state=TX ...

<https://powcoder.com>

/user/hive/warehouse/users/date=20090903/state=CA

/user/hive/warehouse/users/date=20090903/state=NY

/user/hive/warehouse/users/date=20090903/state=TX ...

Add WeChat powcoder

Hive Data Model – partition and cluster

- Tables stored under user/hive/warehouse in HDFS
- Partition columns

```
/user/hive/warehouse/users/date=20090901/state=CA  
/user/hive/warehouse/users/date=20090901/state=NY  
/user/hive/warehouse/users/date=20090901/state=TX  
...  
/user/hive/warehouse/users/date=20090902/state=CA  
/user/hive/warehouse/users/date=20090902/state=NY  
/user/hive/warehouse/users/date=20090902/state=TX  
...  
/user/hive/warehouse/users/date=20090903/state=CA  
/user/hive/warehouse/users/date=20090903/state=NY  
/user/hive/warehouse/users/date=20090903/state=TX  
...
```

- Buckets – allows to create smaller range partitions

Add WeChat powcoder

```
CREATE TABLE page_view(viewTime INT, userid BIGINT,  
                      page_url STRING, referrer_url STRING,  
                      ip STRING COMMENT 'IP Address of the User')  
COMMENT 'This is the page view table'  
PARTITIONED BY (dt STRING, country STRING)  
CLUSTERED BY (userid) INTO 32 BUCKETS  
ROW FORMAT DELIMITED  
                      FIELDS TERMINATED BY '\t'  
                      LINES TERMINATED BY '\n'  
STORED AS SEQUENCEFILE;
```

Data model - buckets

- concept of *buckets*, which provide efficiency to queries that can work well on a random sample of data
- Bucketing divides data into a specified number of files based on the hash of the bucket column

/user/hive/warehouse/users/date=20090901/state=CA/part-00000.

/user/hive/warehouse/users/date=20090901/state=CA/part-00031

/user/hive/warehouse/users/date=20090901/state=NY/part-00000 ...

/user/hive/warehouse/users/date=20090901/state=NY/part-00031

/user/hive/warehouse/users/date=20090901/state=TX/part-00000

- Example - pageview

```
CREATE TABLE page_view(viewTime INT, userid BIGINT,
                      page_url STRING, referrer_url STRING,
                      ip STRING COMMENT 'IP Address of the User')
COMMENT 'This is the page view table'
PARTITIONED BY (dt STRING, country STRING)
CLUSTERED BY (userid) INTO 32 BUCKETS
ROW FORMAT DELIMITED
  FIELDS TERMINATED BY '\t'
  LINES TERMINATED BY '\n'
STORED AS SEQUENCEFILE;
```

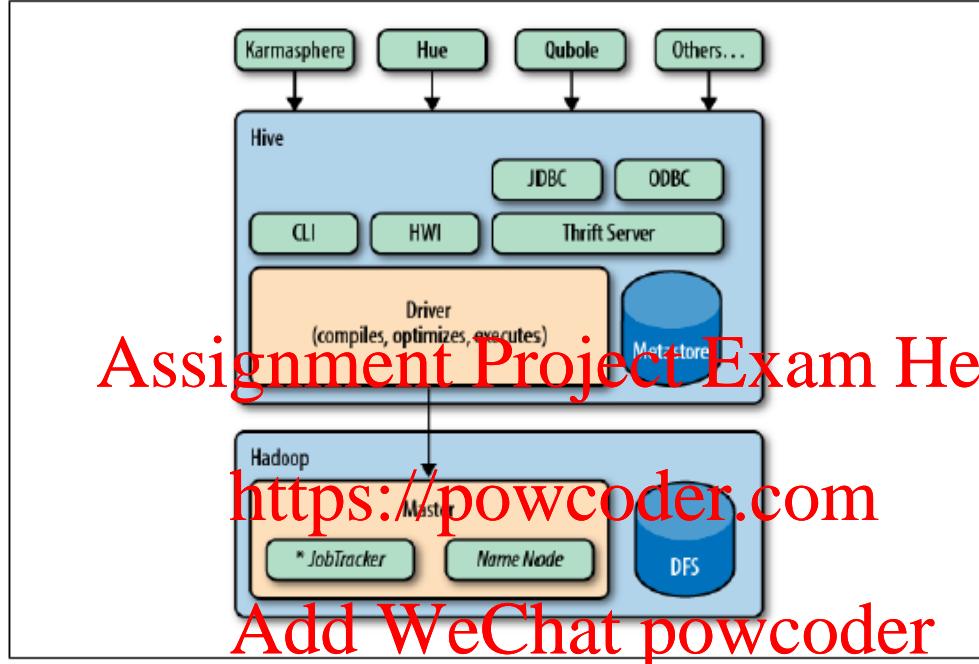
Resources

- Papers
 - Google File System, 2003
 - Google MapReduce, 2004
 - Google Bigtable, 2006
- URLs
 - Apache Hadoop: <https://powcoder.com>
- Available Hadoop Distributions
 - Apache, IBM, Cloudera, Hortonworks

Assignment Project Exam Help

Add WeChat powcoder

Hive Architecture



- Main components
 - SQL interface
 - Parser/Planner
 - Metastore
 - Driver