MapReduce application support on MTA Cloud

Enikő Nagy
József Kovács
Róbert Lovas

MTA SZTAKI Computer and Automation Research Institute
Hungarian Academy of Sciences,
Laboratory of Parallel and Distributed Systems

Topics

- Hadoop advantages
- Main goals
- Occopus
- Occopus descriptors
- Usage
- Experiences on MTA Cloud

MapReduce usage

Many scientific applications, such as
- weather forecasting
- DNA sequencing
- and molecular dynamics

have now been parallelized using Hadoop.

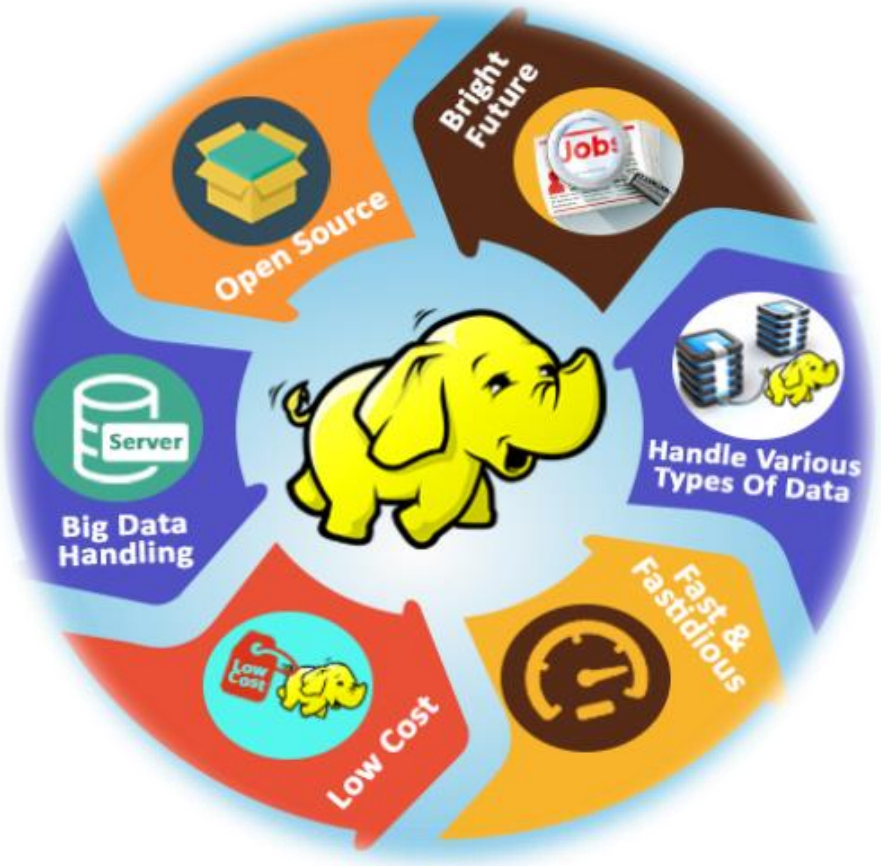


To run MapReduce application in an efficient way it needs Hadoop cluster.

However, the deployment of a fully functional Hadoop cluster is not a trivial task, it is currently not in line with the capabilities of the data scientists, and therefore there is still a significant barrier for this technology to spread among data scientists.

Hadoop - 5 major advantages

- Fast
- Flexible
- Resilient to failure
- Cost effective
- Scalable



Main goals

Motivation:

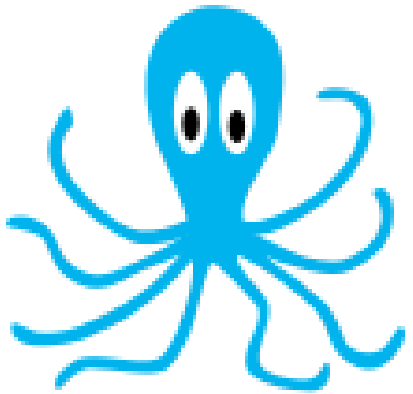
- MTA Cloud provides easy to create Linux and Windows machine images, however complex infrastructures are not supported yet
- Hadoop cluster as a complex infrastructure should be supported by MTA Cloud (since Hadoop cluster is highly needed by Big Data application)

Goals:

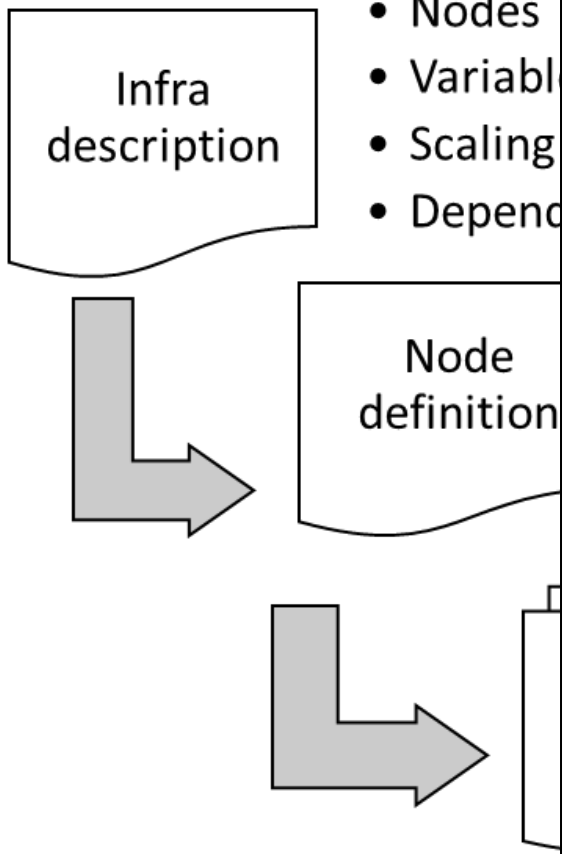
1. Usability and flexibility
2. Easy to use
3. Scalable
4. Does not require any prepared image

Occopus

- Hybrid, cloud orchestrator tool
- Developed by MTA SZTAKI
- Multi-cloud solution (can be used in private and in public cloud too)
- Contextualization with cloud-init
- Enable scaling manually
- No vendor lock-in (portable)



Occopus descriptors



- Nodes
- Variabl
- Scaling
- Depend

```
'node_def: hadoop_master_node ':
    -
        resource:
            type: nova
            endpoint: https://sztaki.
            image_id: ...
            network_id: ...
            flavor_name: ...
            security_groups:...
        contextualisation:
            type: cloudinit
            context_template: !yaml_i
                url: file://cloud_ini
        health_check:
            ports:
                - 50070
'node_def: hadoop_slave_node ':
    -
        resource:
            type: nova
            endpoint: https://sztaki.cloud.mta.hu...
            project id: ...
```

```
nodes:
    - &M
        name: hadoop_master
        type: hadoop_master_node
    - &S
        name: hadoop_slave
        type: hadoop_slave_node
        scaling:
            min: 2
            max: 10
dependencies:
    -   connection: [ *S, *M ]
```

- Download binaries (Consul, Hadoop, Java, SSH)
- Deploy configuration files
- Create Hadoop user
- Configure SSH, Hadoop
- Start Hadoop deamons
- Start Consul template and service

Levels of usability

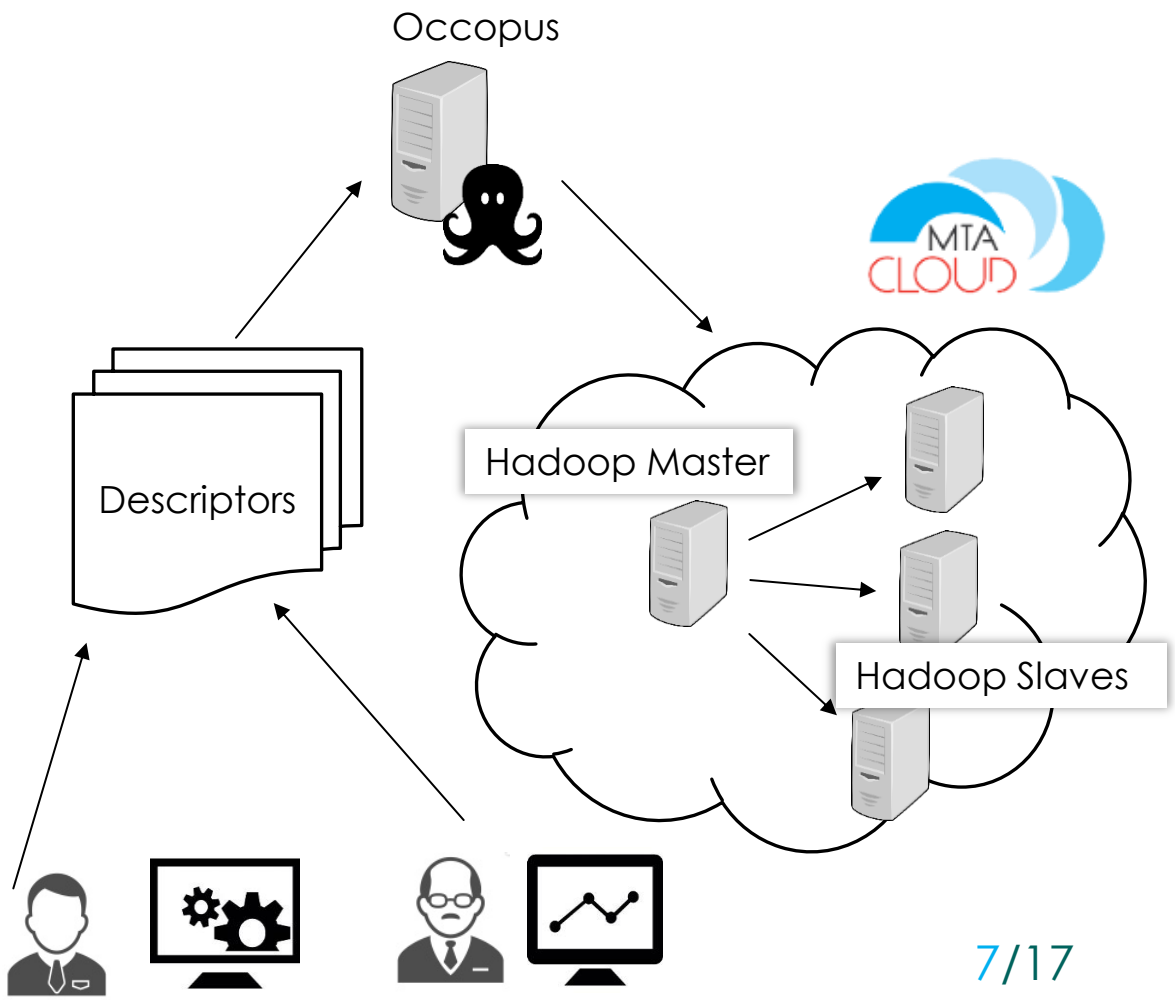
Level 1: Creation of Occopus
(done by SZTAKI)

Level 2: Creation of Occopus descriptors for Hadoop
(done by SZTAKI)

Level 3: User personalisation of Occopus descriptors
(institutional IT experts based on Hadoop Tutorial on Occopus webpage)

Level 4: Build Hadoop cluster
(end-user scientists using personalized descriptors and Occopus)

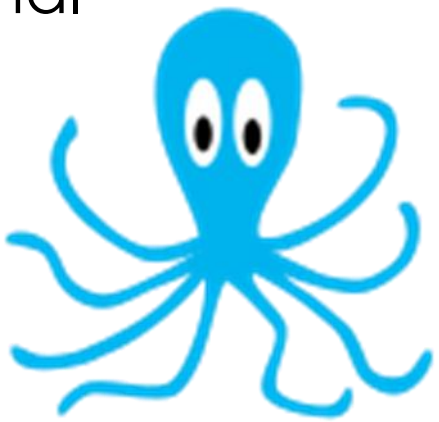
Level 5: Execution of MapReduce application in the Hadoop cluster
(end-user scientists)

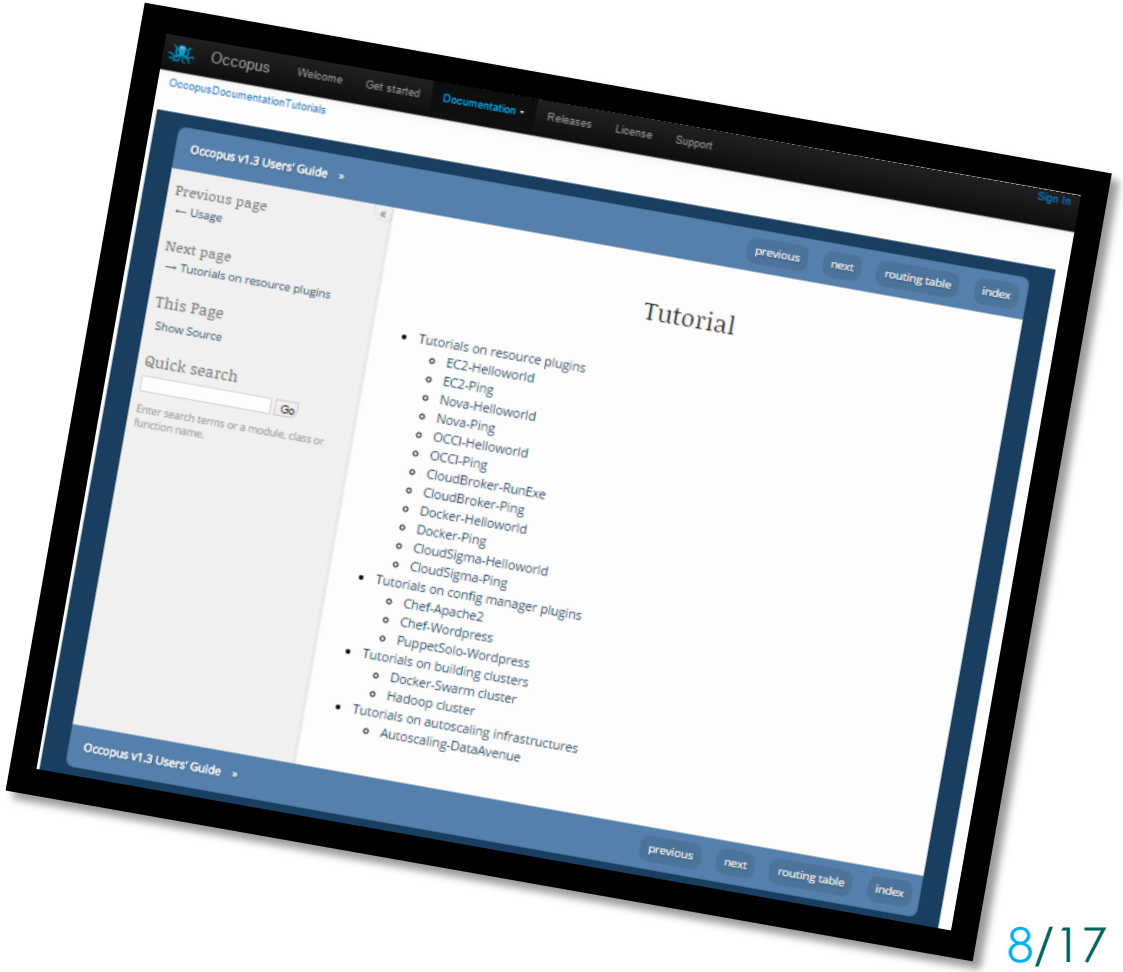


Hadoop Tutorial
(Showing result of Level 2 action)

Visit: http://occopus.lpds.sztaki.hu

- **Users' guide - Tutorial**
  - Tutorials on building clusters
  - Hadoop cluster
- Download descriptors
- Step-by-step tutorial





Personalizing descriptors (Level 3 action)

1. **Authenticati**
   - Users ha… authentic… …tly in their …e to use.
2. **Node …**
   - you…
   - you…

For more …

- Visit htt…
  - you…
  - you…
- Alterna…

```
'node_def: hadoop_master_node ':
    -
        resource:
            type: nova
            endpoint: https://sztaki.cloud.mta.hu...
            image_id: ...
            network_id: ...
            flavor_name: ...
            security_groups:...
        contextualisation:
            type: cloudinit
            context_template: !yaml_import
                url: file://cloud_init_hadoop_master.yaml
        health_check:
            ports:
                - 50070
'node_def: hadoop_slave_node ':
    -
        resource:
            type: nova
            endpoint: https://sztaki.cloud.mta.hu...
            project_id: ...
```

How to build a Hadoop cluster with Occopus? (Level 4 action)

Step 0: **Create a VM in MTA Cloud** (recommended)

Step 1: **Install Occopus**
http://occopus.lpds.sztaki.hu
Follow the steps below: Get started → Install Manual

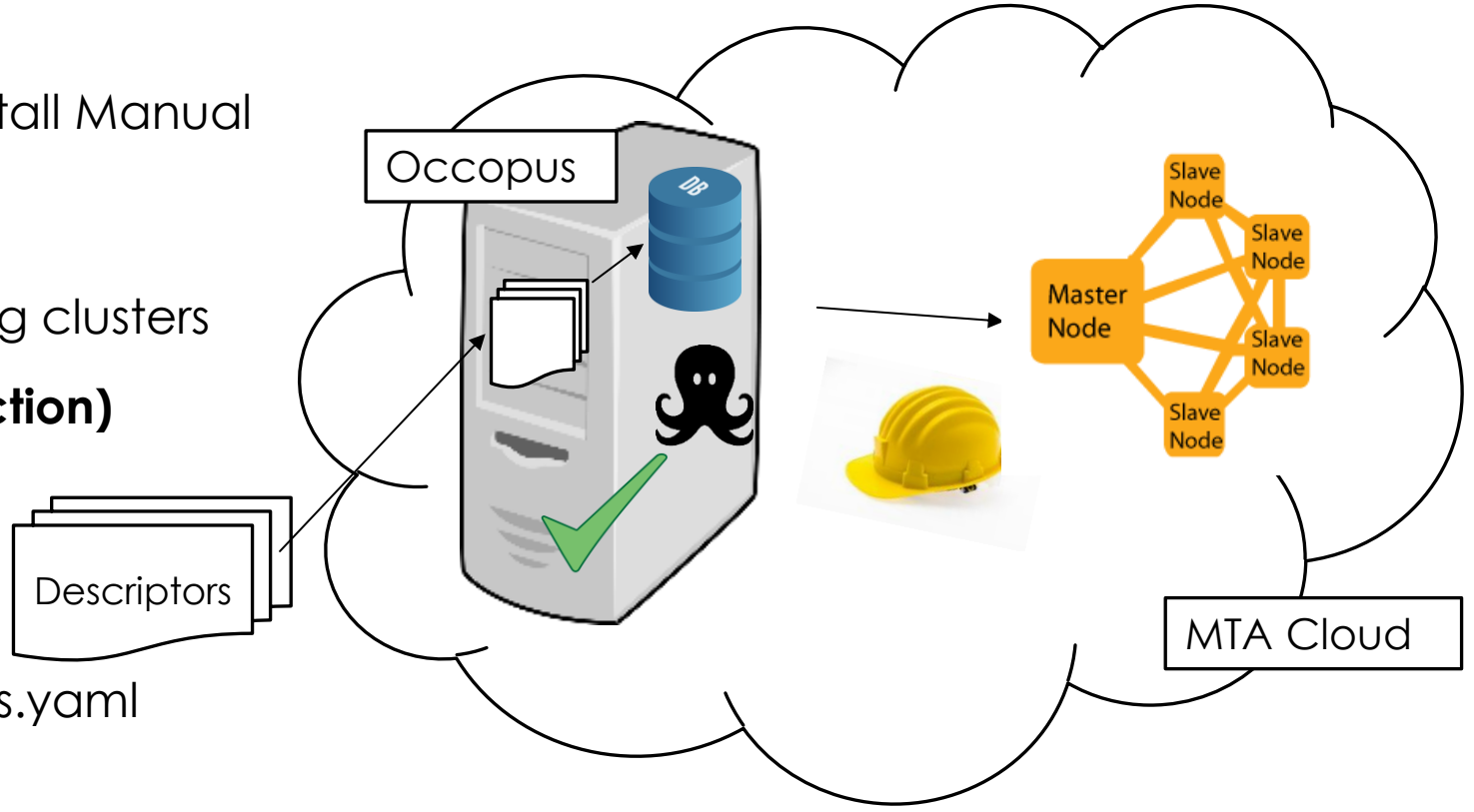
Step 2: **Download descriptors**
Visit: http://occopus.lpds.sztaki.hu
Users' guide - Tutorial - Tutorials on building clusters

Step 3: **Personalize descriptors (Level 3 action)**

Step 4: **Make sure Occopus is activated:**
$ source ~/occopus/bin/activate

Step 5: **Import node definitions:**
$ occopus-import nodes/node_definitions.yaml

Step 6: **Start building process:**
$ occopus-build --parallelize infra-hadoop-cluster.yaml



Scale-up or down

Scaling is a two-phase operation: first we register the scaling request, and after that we scale up/down the selected infrastructure by building new nodes /destroying old ones

1. **$ occopus-scale**
   Registers scaling requests
   Usage: occopus-scale -n hadoop_slave -c COUNT -i INFRA_ID
   Count: positive/negative number expressing the direction and magnitude of scaling
2. **$ occopus-maintain**
   Requests are handled and realized by this command
   Usage: occopus-maintain –i INFRA_ID

For more information visit: http://occopus.lpds.sztaki.hu

How to run a Hadoop MapReduce job?
1. Inputs – on Hadoop Master node

After building-up a virtual Hadoop infrastructure we can run MapReduce job on it, follow these steps:

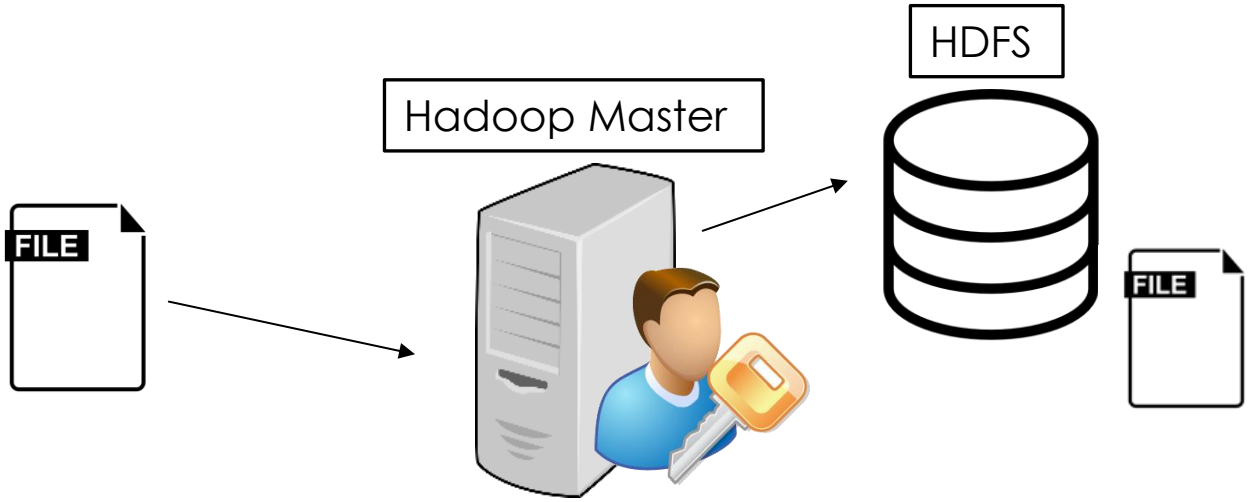
Step 1: Copy input files to Hadoop Master node

Step 2: Log in to HadoopMaster node (SSH)

Step 3: Import inputs to HDFS (use commands as hduser):

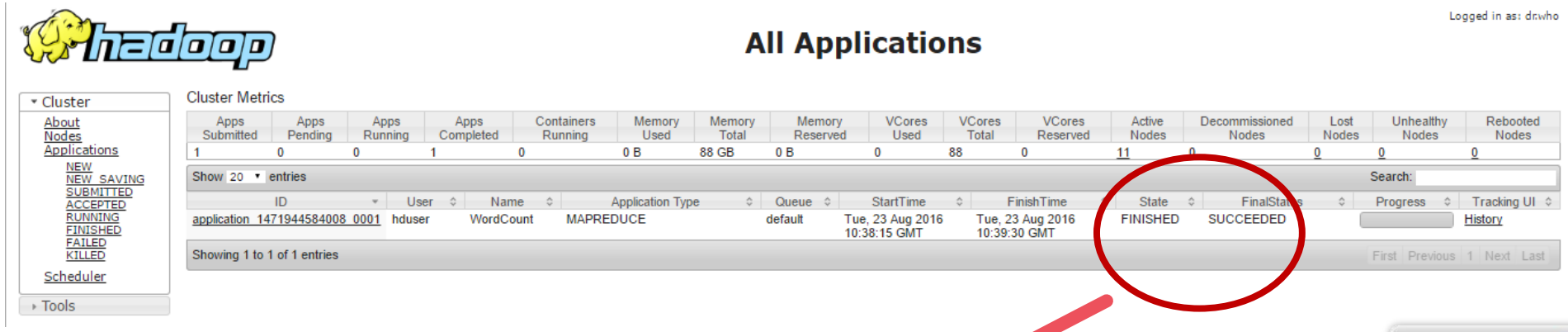
$HADOOP_HOME/bin/hadoop fs -mkdir /input

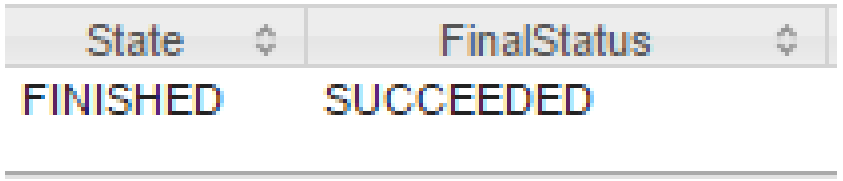
$HADOOP_HOME/bin/hadoop fs -put /home/hduser/input/file01.txt /input



How to run a Hadoop MapReduce job?
2. Run a Hadoop job – on Hadoop Master node

- Use this command as **hduser on Hadoop Master node**:
- **$HADOOP_HOME/bin/hadoop jar /home/hduser/input/application.jar org.myorg.Application /input /output**
- To check, read console or visit: http://HadoopMasterIP:8088



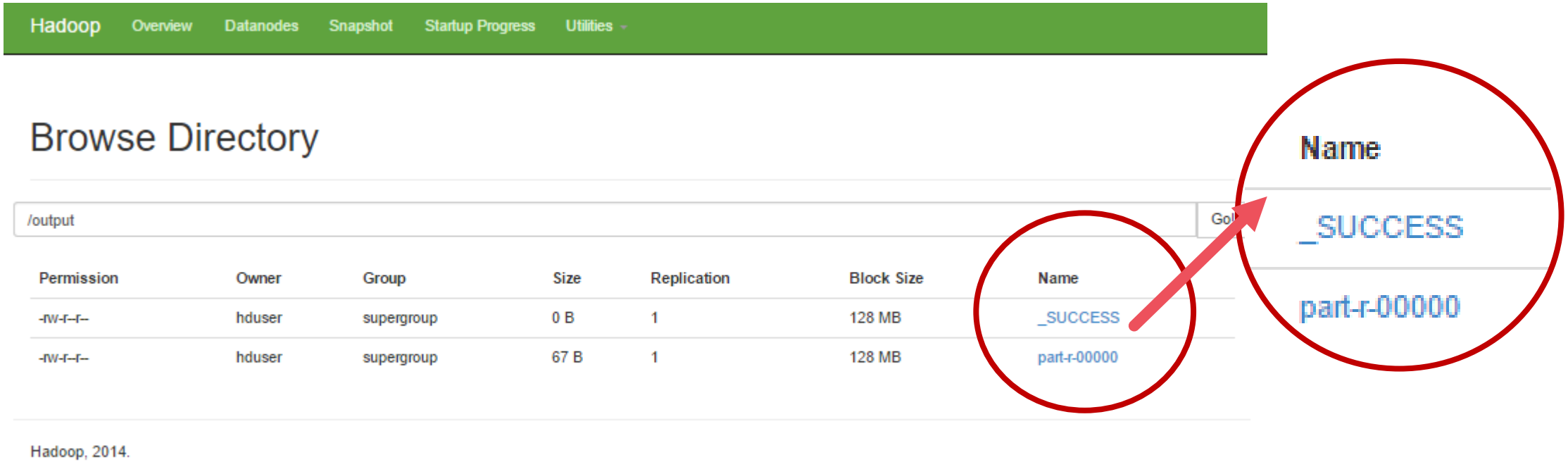


http://HadoopMasterIP:8088

Hadoop Web UI

How to run a Hadoop MapReduce job?
2. Run a Hadoop job – on Hadoop Master node

To check the **output** of the MapReduce job visit web UI of the NameNode:
http://HadoopMasterIP:50070
Choose: Utilities -> Browse the file system -> select /output -> **download** part-r-00000



Conclusion

Advantages of the solution:

- If you have a MapReduce application you would like to run on MTA Cloud you can easily build the required Hadoop cluster on MTA Cloud (see the steps we showed before)
- This Hadoop Cluster will be
  - Portable,
  - Scalable
- Building the Hadoop cluster does not require any specially prepared image, a simple Ubuntu image is enough

New features coming soon:

- Tutorial on automatic scaling of Hadoop cluster with Prometheus

Experiences on MTA Cloud

- Although I experienced small operational problems during my work the administrators responded rapidly fixed the problem or gave me assistance, for example:
  - Moving to another project (Oktatas →Occopus)
  - Quota lifting (more floating IP)
  - Help with NOVA plugin during Occopus development (code review)
  - VM slow start → they immediately began looking for the error and repaired it

Thank you for your attention!

Enikő Nagy
E-mail: eniko.nagy@sztaki.mta.hu