Big data for internet applications

Structure of Spark programs

- The Driver program
  - Contains the main method
  - “Defines” the workflow of the application
  - Accesses Spark through the **SparkContext** object
    - The SparkContext object represents a connection to the cluster
  - Defines Resilient Distributed Datasets (RDDs) or Datasets that are “allocated” on the nodes of the cluster
  - Invokes parallel operations on RDDs or Datasets

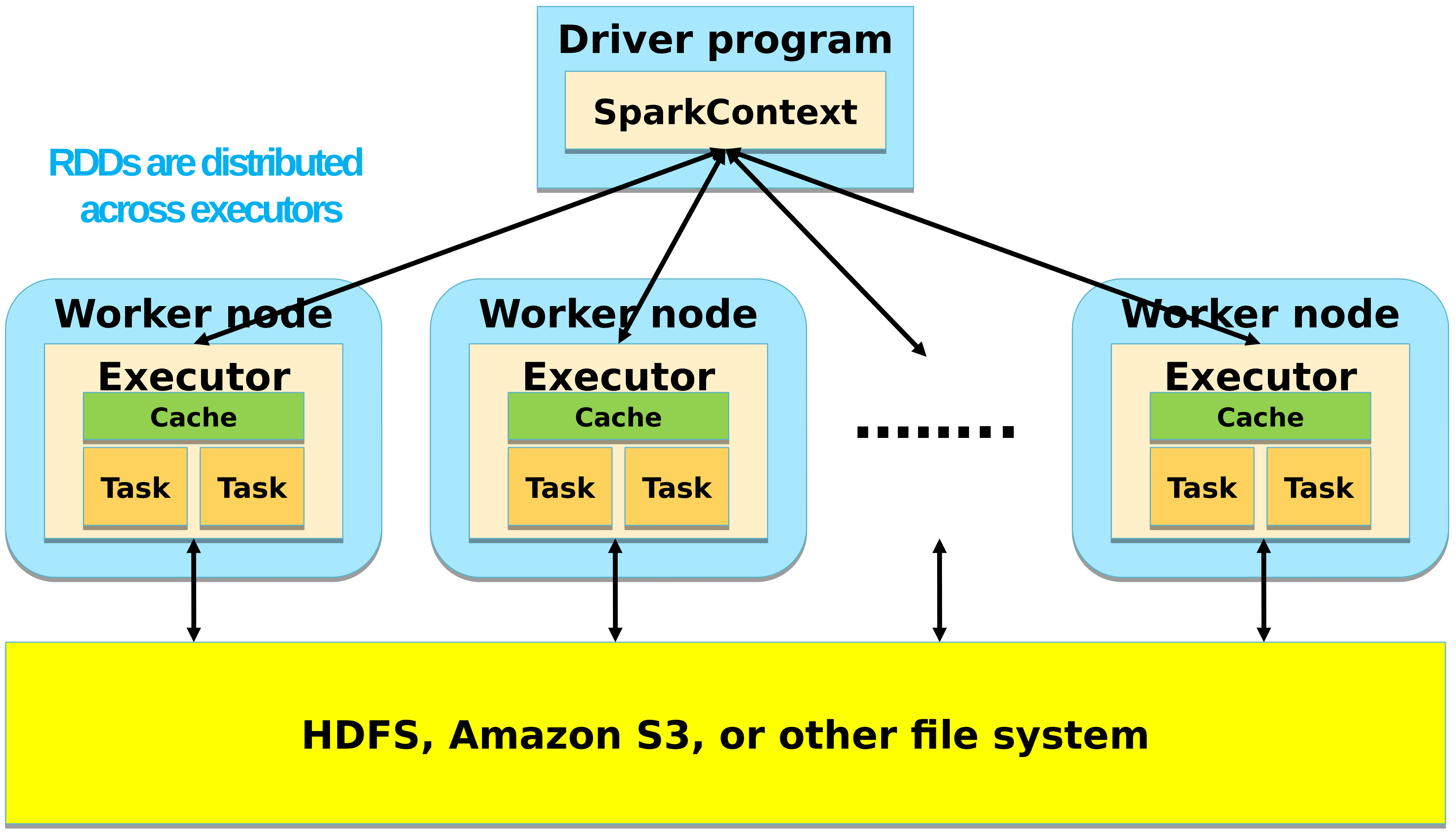
Structure of Spark programs

- The Driver program defines
  - Local variables
    - The standard variables of the Python programs
  - RDDs or Datasets
    - Distributed “variables” stored in the nodes of the cluster
  - The SparkContext object allows
    - Creating RDDs or Datasets
    - “Submitting” executors (processes) that execute in parallel specific operations on RDDs or Datasets
      - Transformations and Actions

Structure of Spark programs

- The worker nodes of the cluster are used to run your application by means of **executors**
- Each executor runs on its partition of the RDD(s) or Dataset(s) the operations that are specified in the driver

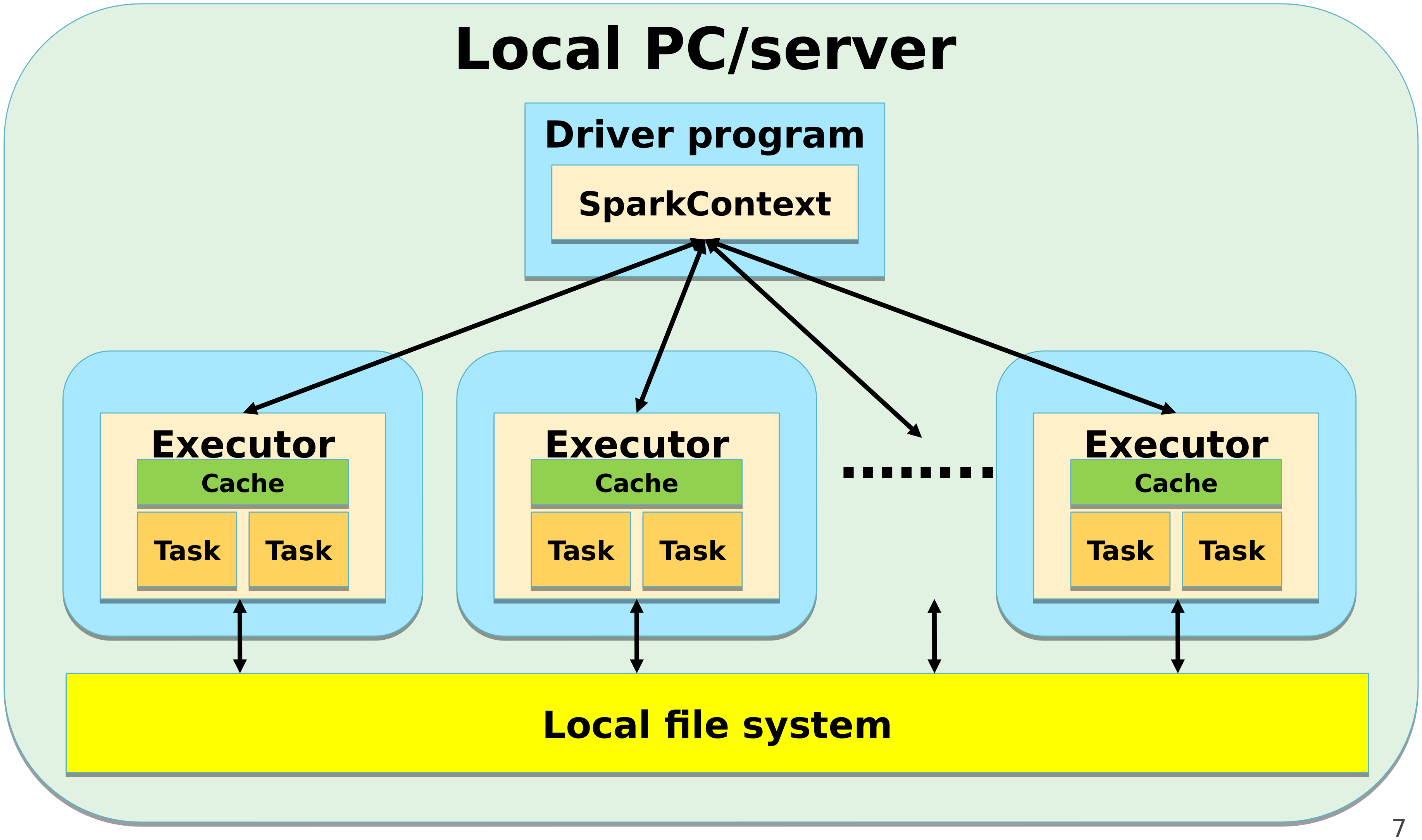
Distributed execution of Spark



Local execution of Spark

- Spark programs can also be executed locally
  - Local threads are used to parallelize the execution of the application on RDDs or Datasets on a single PC
    - Local threads can be seen are “pseudo-worker” nodes
  - It is useful to develop and test the applications before deploying them on the cluster
  - A local scheduler is launched to run Spark programs locally

Local execution of Spark



Spark official terminology

- Application
  - User program built on Spark
  - It consists of a driver program and executors on the cluster
- Driver program
  - The process running the main() function of the application and creating the SparkContext

Based on http://spark.apache.org/docs/latest/cluster-overview.html

Spark official terminology

- Cluster manager
  - An external service for acquiring resources on the cluster (e.g. standalone manager, Mesos, YARN)
- Deploy mode
  - Distinguishes where the driver process runs
    - In "cluster" mode, the framework launches the driver inside of the cluster
    - In "client" mode, the submitter launches the driver outside of the cluster
- Worker node
  - Any node of the cluster that can run application code in the cluster

Spark official terminology

- Executor
  - A process launched for an application on a worker node, that runs tasks and keeps data in memory or disk storage across them
  - Each application has its own executors
- Task
  - A unit of work that will be sent to one executor
- Job
  - A parallel computation consisting of multiple tasks that gets spawned in response to a Spark action (e.g. save, collect)

Spark official terminology

- Stage
  - Each job gets divided into smaller sets of tasks called stages that depend on each other (similar to the map and reduce stages in MapReduce)

How to submit a Spark application

Spark-submit

- Spark programs are executed/submitted on the cluster by using the spark-submit command
  - It is a command line program
  - It is characterized by a set of parameters
    - E.g., the name of the python file containing the Spark program we want to execute
    - The parameters of the Spark application
    - etc.

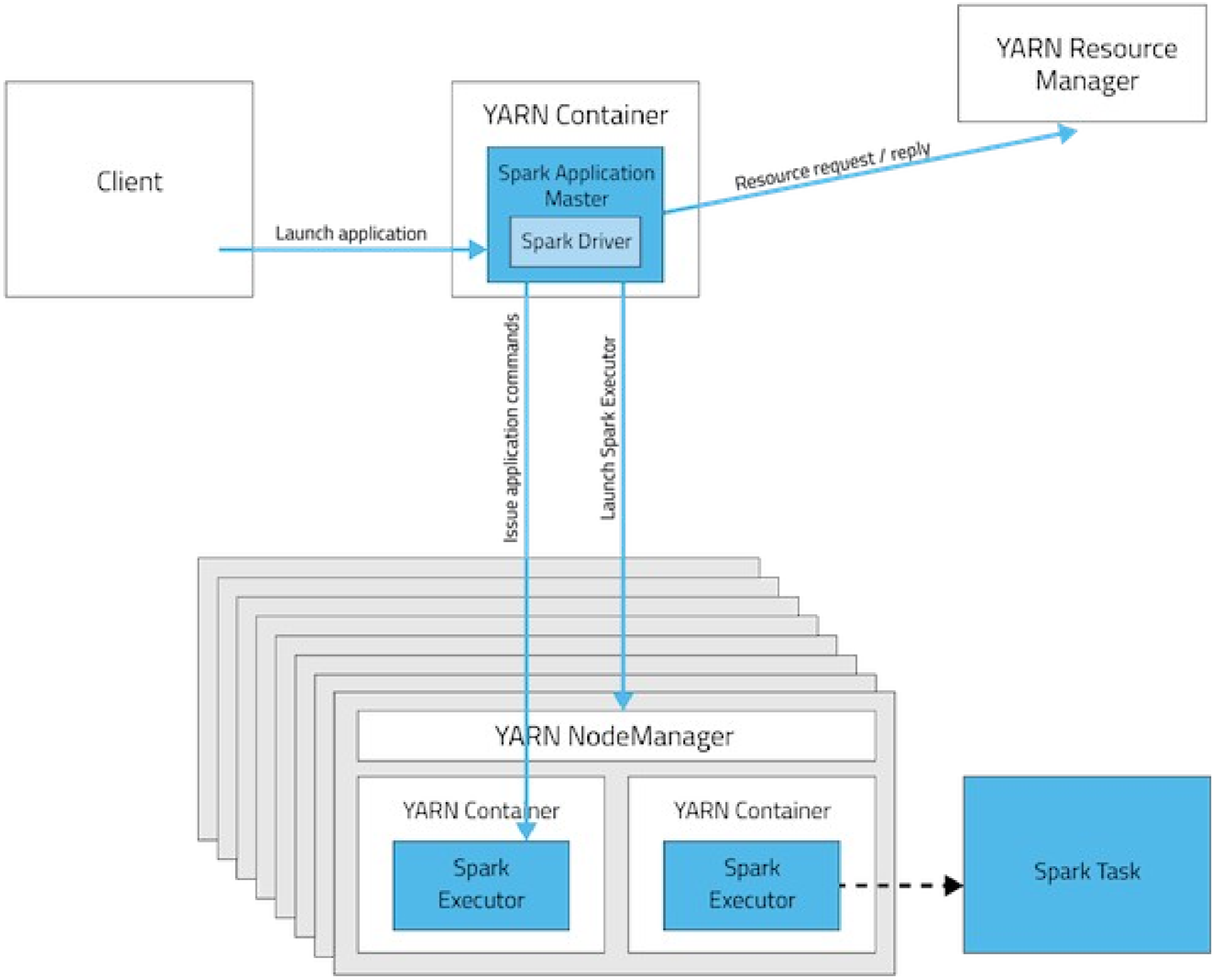
Spark-submit

- spark-submit has also two parameters that are used to specify which scheduler is used and where the driver is launched
  - **--master** option
    - Specify which environment/scheduler is used to execute the application
      - yarn The YARN scheduler (i.e., the one of Hadoop)
      - local The application is executed exclusively on the "local" PC

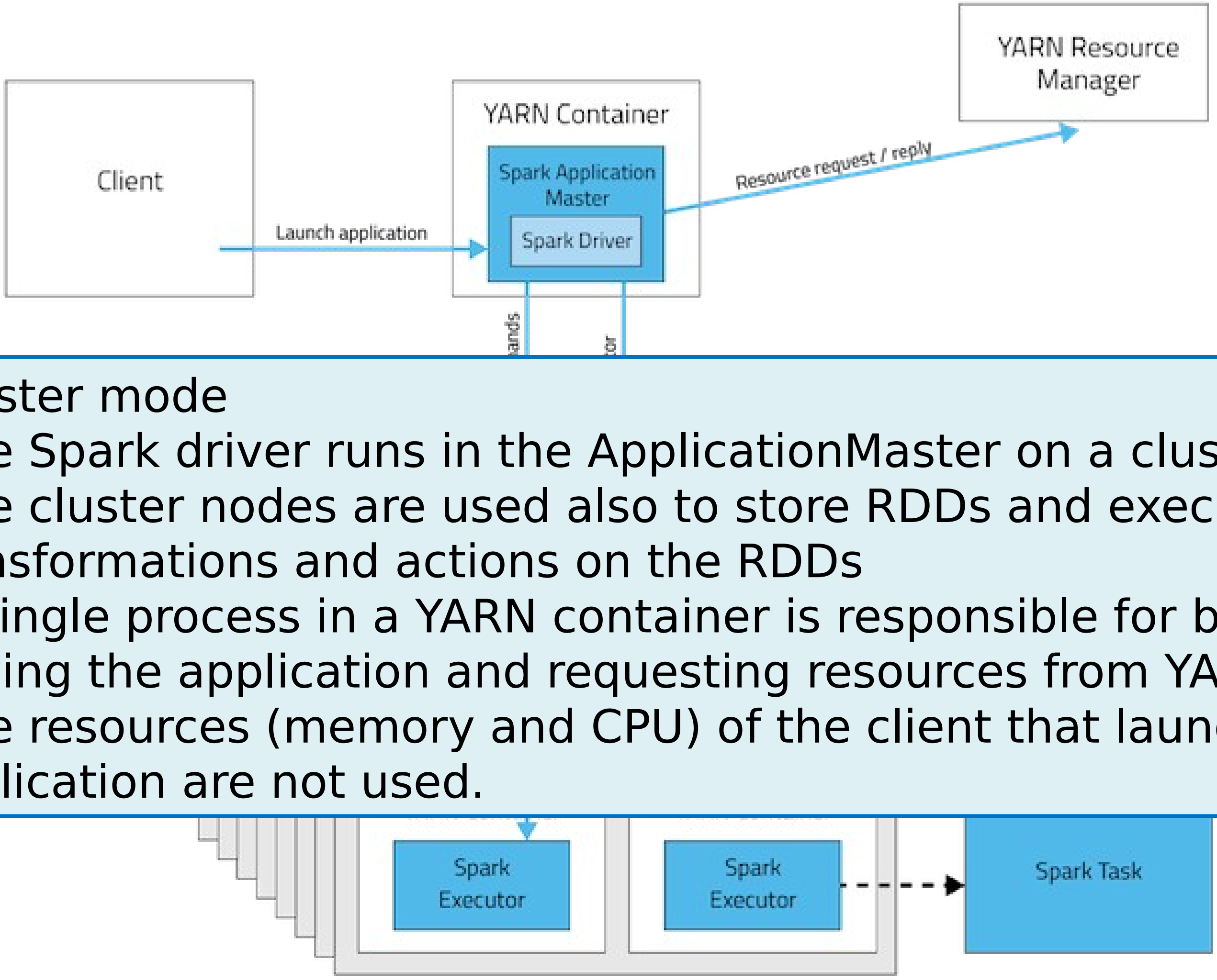
Spark-submit

- **--deploy-mode** option
  - Specify where the Driver is launched/executed
    - client The driver is launched locally (in the "local" PC executing spark-submit)
    - cluster The driver is launched on one node of the cluster

Cluster Deployment Mode



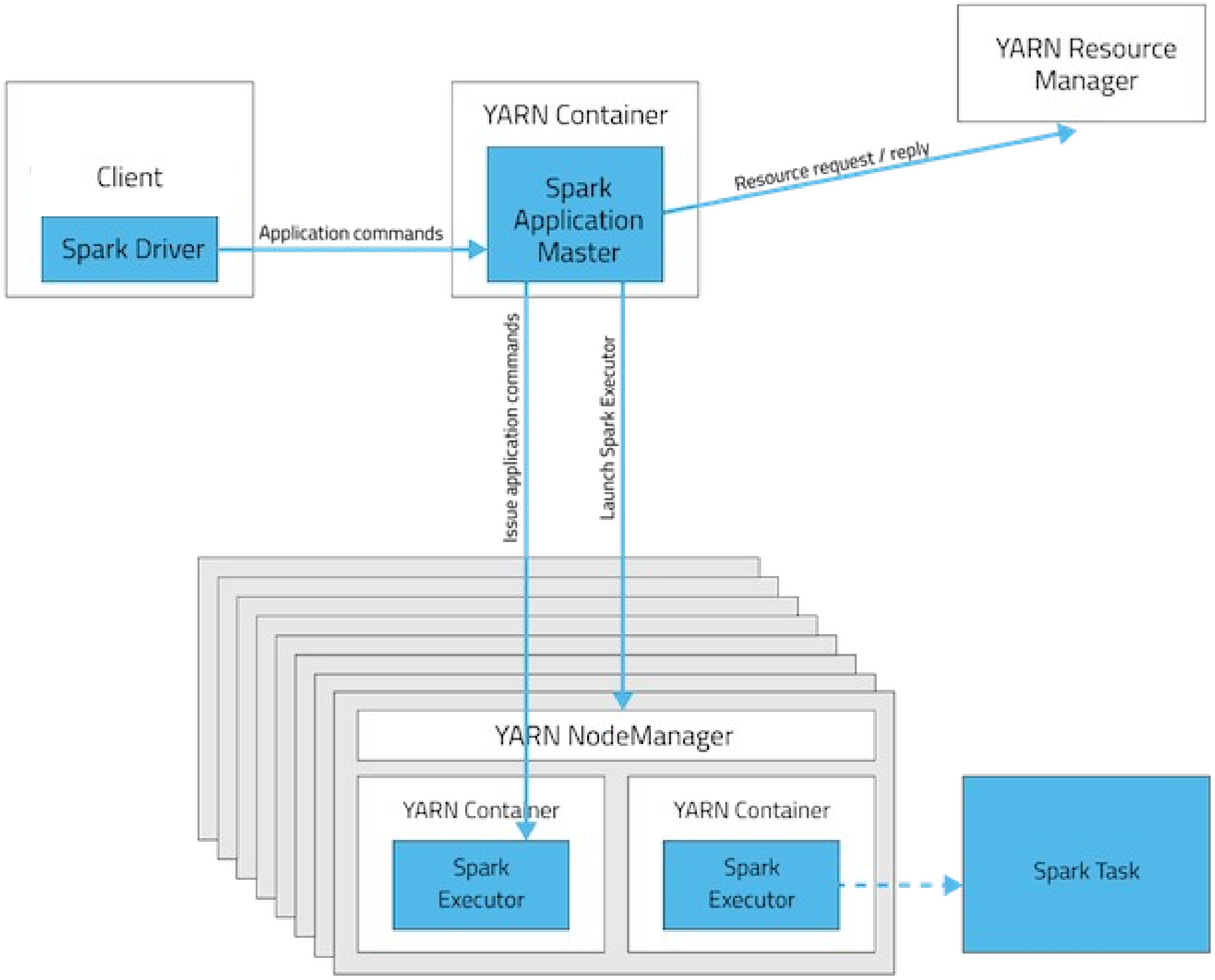
Cluster Deployment Mode



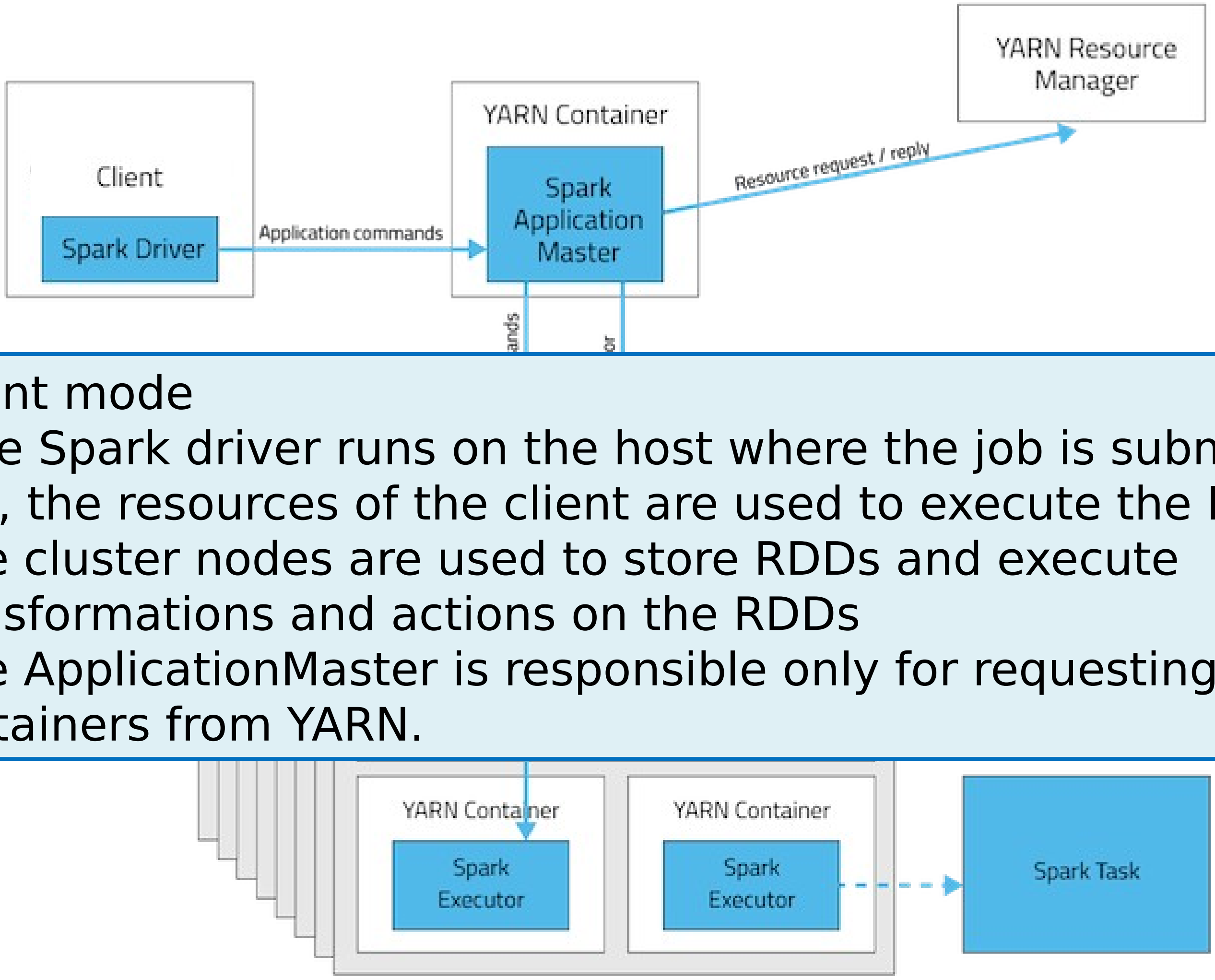
In cluster mode

- The Spark driver runs in the ApplicationMaster on a cluster node.
- The cluster nodes are used also to store RDDs and execute transformations and actions on the RDDs
- A single process in a YARN container is responsible for both driving the application and requesting resources from YARN.
- The resources (memory and CPU) of the client that launches the application are not used.

Client Deployment Mode



Client Deployment Mode



In client mode

- The Spark driver runs on the host where the job is submitted (i.e., the resources of the client are used to execute the Driver)
- The cluster nodes are used to store RDDs and execute transformations and actions on the RDDs
- The ApplicationMaster is responsible only for requesting executor containers from YARN.

Spark-submit: setting executors

- Spark-submit allows specifying
  - The number of executors
    - --num-executors NUM
      - Default value: NUM=2 executors
  - The number of cores per executor
    - --executor-cores NUM
      - Default value: NUM=1 core
  - Main memory per executor
    - --executor-memory MEM
      - Default value: MEM=1GB
- The maximum values of these parameters are limited by the configuration of the cluster

Spark-submit: setting driver

- Spark-submit allows specifying
  - The number of cores for the instance of the driver
    - --driver-cores NUM
      - Default value: NUM=1 core
  - Main memory for the instance of the driver
    - --driver-memory MEM
      - Default value: MEM=1GB
- Also the maximum values of these parameters are limited by the configuration of the cluster when the deploy-mode is set to cluster

Spark-submit: Execution on the cluster

- The following command submits a Spark application on a Hadoop cluster
  spark-submit --deploy-mode *cluster* --master *yarn MyApplication.py arguments*
- It executes/submits application contained in *MyApplication.py*
- The application is executed on a Hadoop cluster based on the YARN scheduler
  - Also the Driver is executed in a node of the cluster

Spark-submit: Local execution

- The following command submits a Spark application on a local PC
  spark-submit --deploy-mode *client* --master *local* *MyApplication.py arguments*
- It executes/submits the application contained in *MyApplication.py*
- The application is completely executed on the local PC
  - Both Driver and Executors
- Hadoop is not needed in this case
  - You only need the Spark software
  - The data are read from/store on the local file system

Spark-submit: Execution on the cluster, driver local

- The following command submits a Spark application on a Hadoop cluster, but the driver, is still "local"
  spark-submit --deploy-mode *client* --master *yarn MyApplication.py arguments*
- It executes/submits the application contained in *MyApplication.py*
- The application is executed on a Hadoop cluster based on the YARN scheduler
- The Driver is executed on the local PC