



Universidad del
Rosario

Big Data

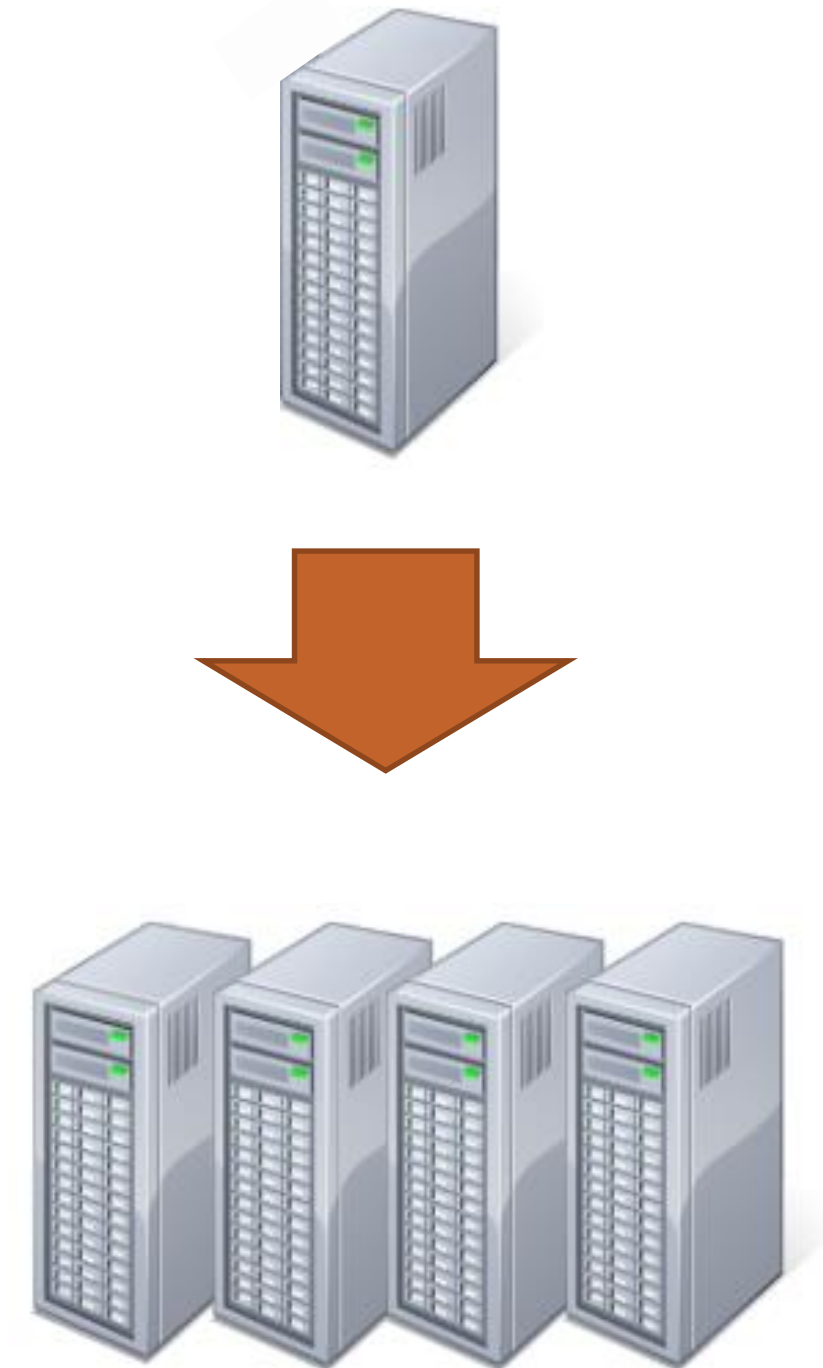
W3. Hadoop

FERNEY ALBERTO BELTRAN MOLINA
Escuela de Ingeniería, Ciencia y Tecnología
Matemáticas Aplicadas y Ciencias de la Computación

Recordando Computación paralela

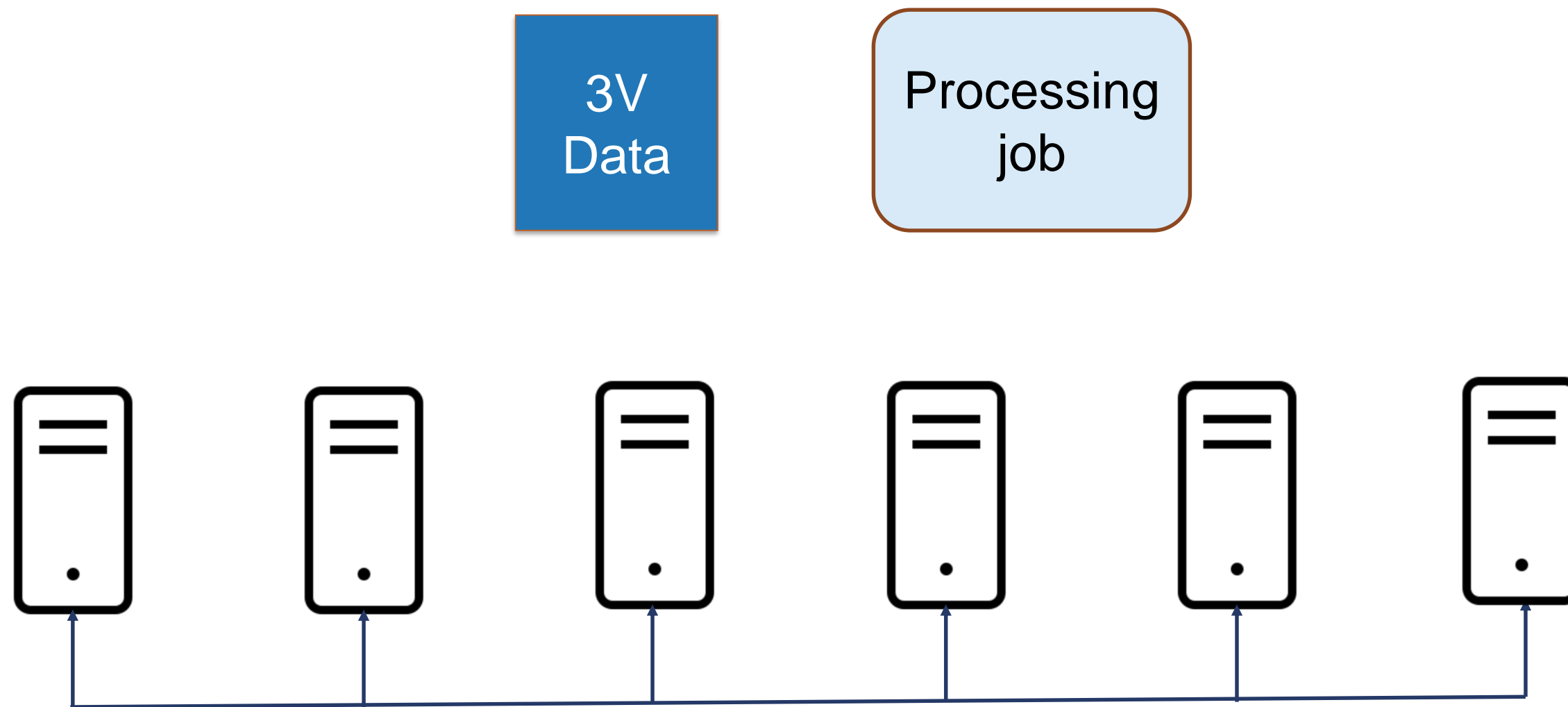
Escalamiento (vertical o horizontal), siempre es necesario:

- Paralelizar la tarea y distribuirla entre cada unidad de procesamiento
- Coordinar las unidades de procesamiento
- Sincronización de datos.
- Ancho de Banda Finito



Más allá de la computación convencional

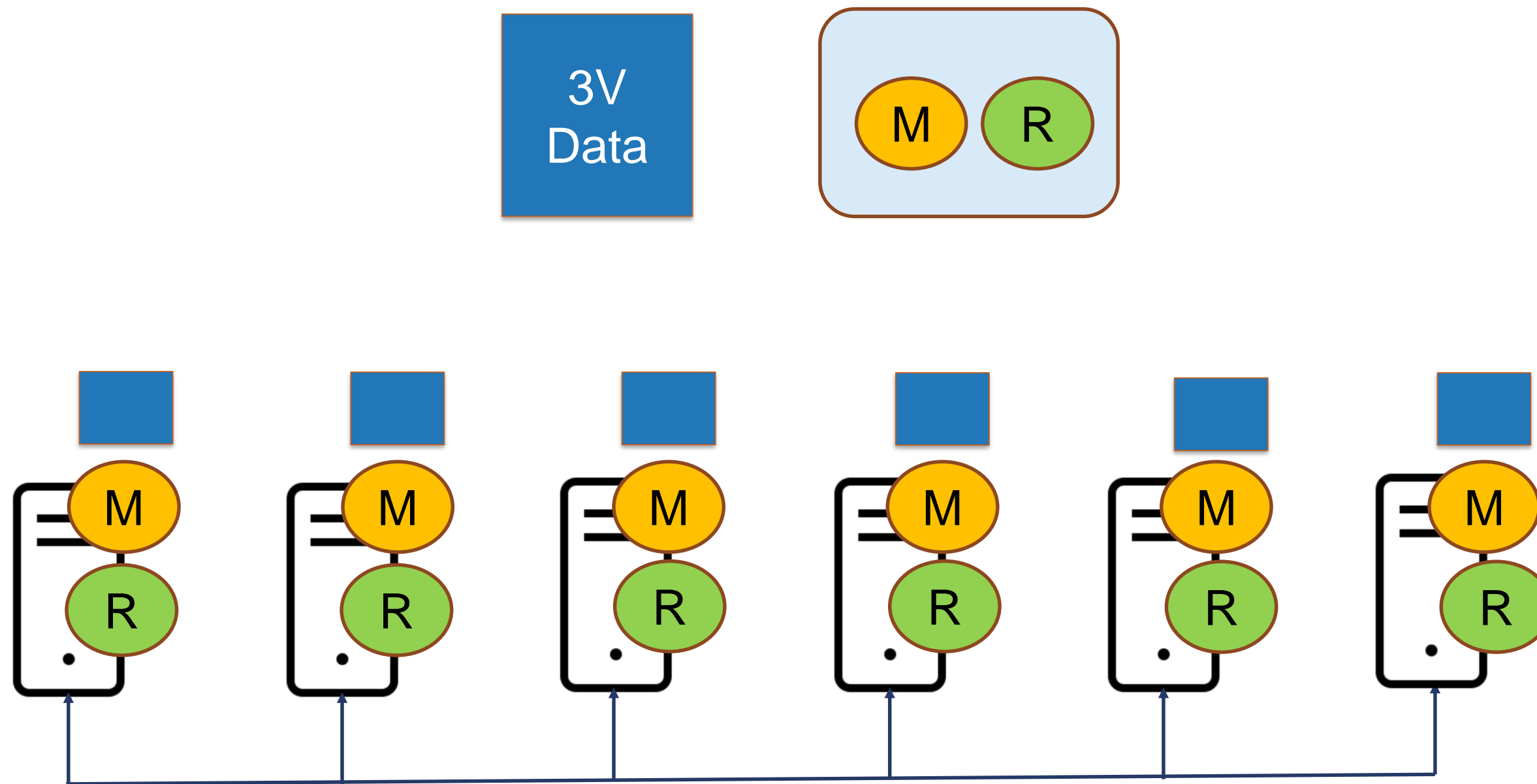
Para procesar datos de 3V, necesitamos ir más allá de la computación convencional. Al paralelizar el trabajo de procesamiento, podemos compartir la carga entre unidades de procesamiento individuales en un sistema de ingeniería de datos de alto rendimiento



Paralelizar MapReduce

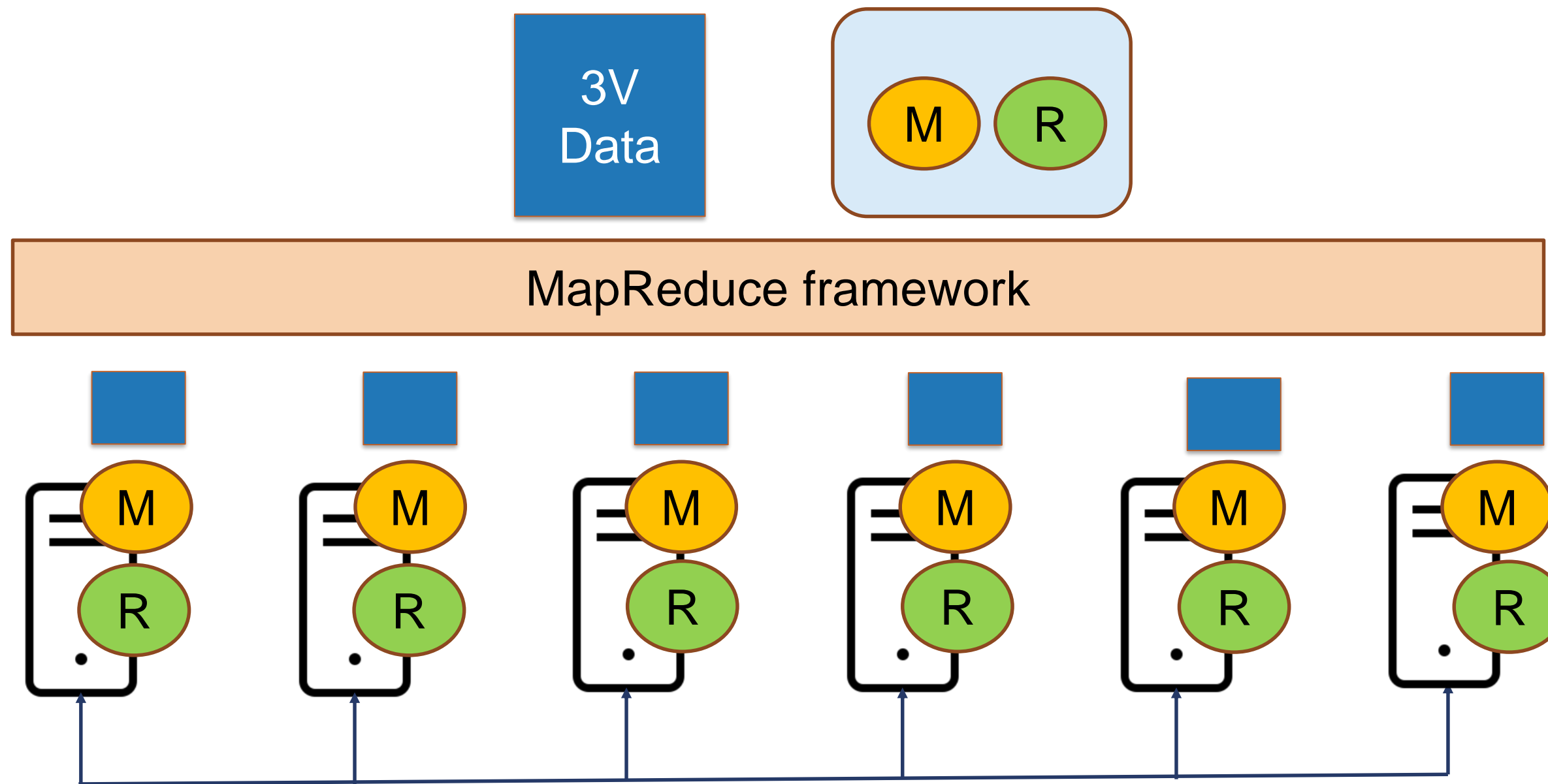
En MapReduce expresamos el trabajo de procesamiento como un mapa y una reducción.

MapReduce ofrece varias oportunidades para la paralelización del almacenamiento y el procesamiento.



MapReduce y Hadoop

Como científicos de datos, es nuestra responsabilidad definir el Mapper correcto y una buenas funciones en reduce. Luego el framework de MapReduce se encargará del resto.



Contenidos

- **Fundamentos de Hadoop**
- Despliegue de Hadoop
- HDFS

Apache Hadoop

Apache Hadoop es un **plataforma de big-data** open-source que comenzó como una implementación de MapReduce. Sus principales características son:

- Sistema de almacenamiento distribuido (HDFS) y capacidades de procesamiento distribuido (originalmente MapReduce, después de Hadoop 2 YARN).
- Reduce el tráfico de la red , Hadoop sigue el enfoque de diseño ***bring computation to data***.

HDFS

MapReduce
YARN



Almacenamiento, procesamiento y análisis
de datos **escalables y económicos**
Distribuido y tolerante a fallos
Aprovecha el poder del hardware estándar

Casos de Apache Hadoop

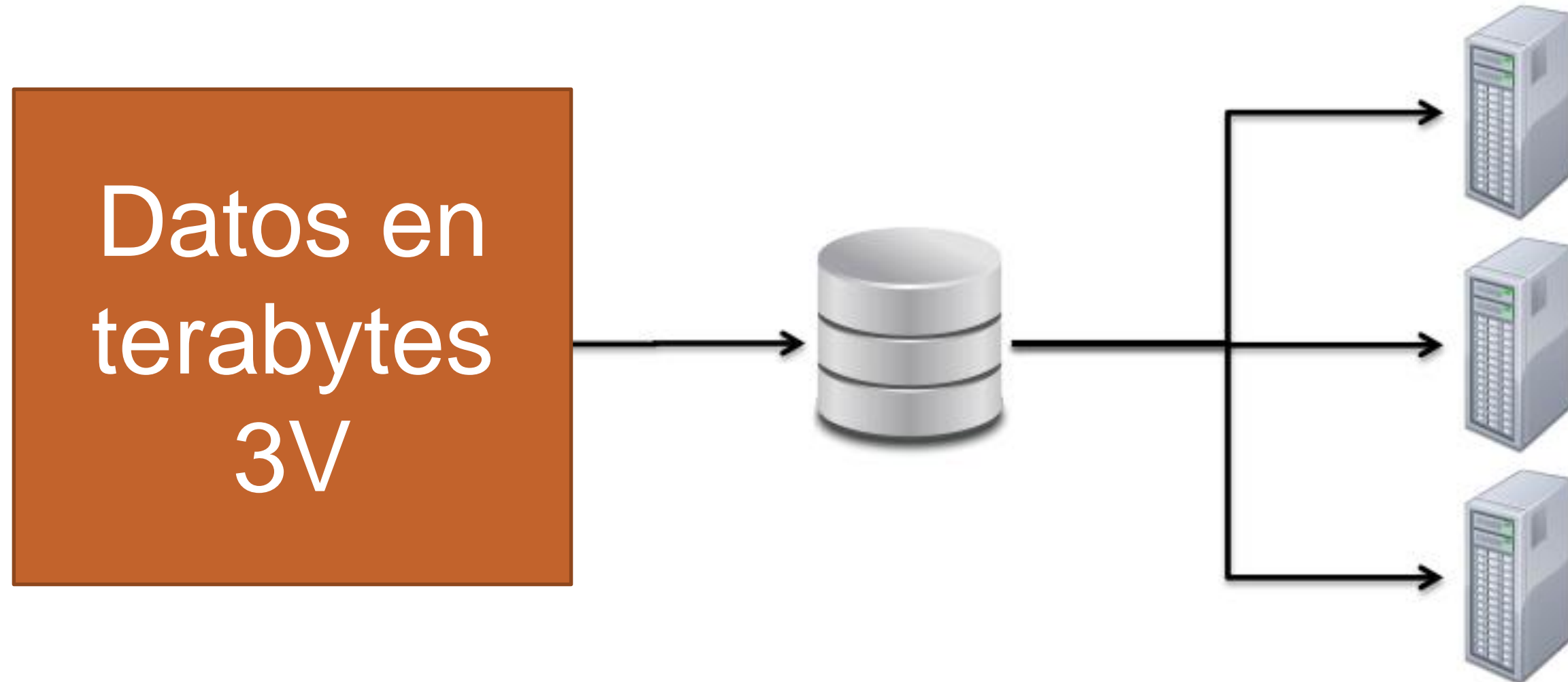
Apache Hadoop se puede implementar para problemas como :

- ETL (Extract, Transform, Load).
- La minería de texto.
- La creación de índices.
- La creación y análisis de gráficos.
- El reconocimiento de patrones.
- Modelos predictivos.
- Análisis de sentimientos.

Sistema de distribución

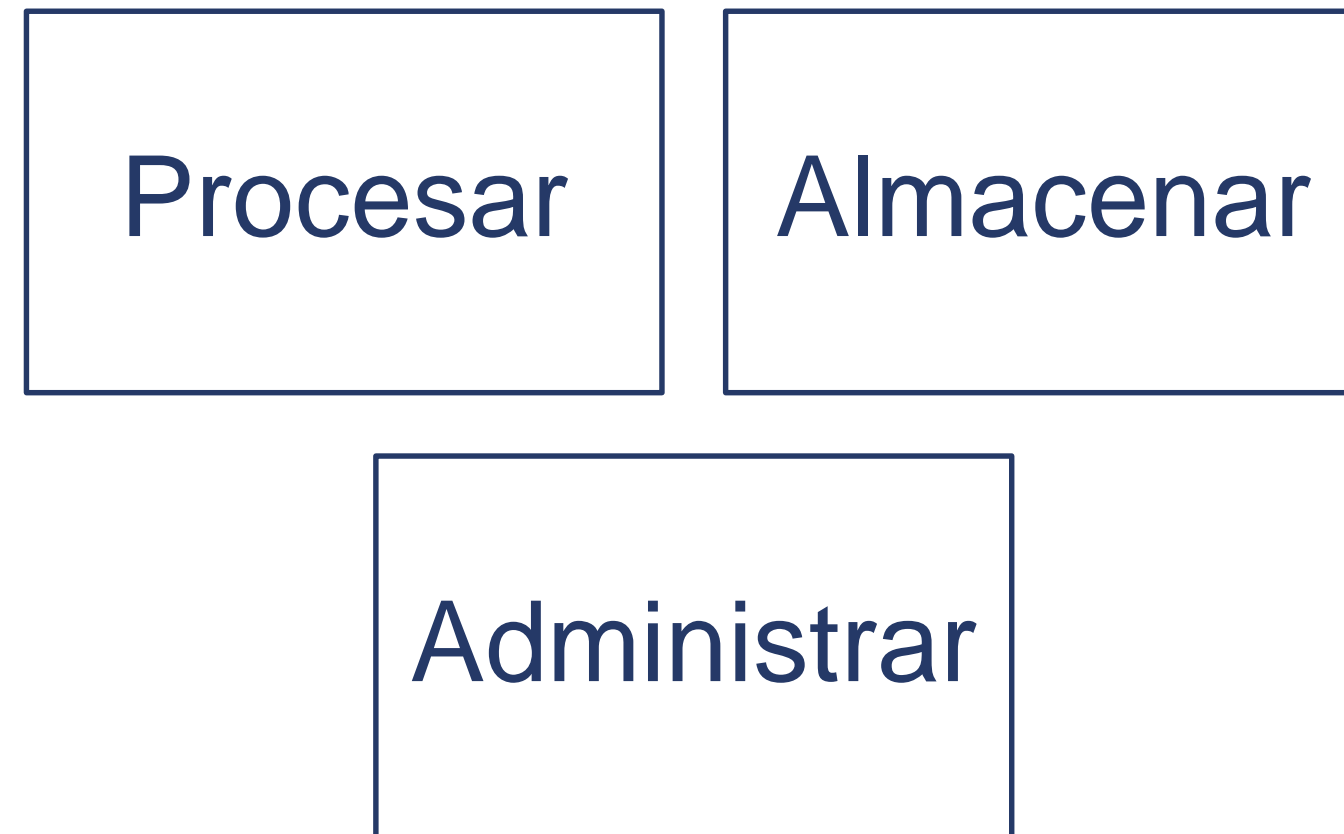
Principal resto , el cuello de botella de almacenamiento de datos :

- Los datos tradicionalmente son almacenados en una localización central
- Los datos son copiados en el tiempo de procesamiento .



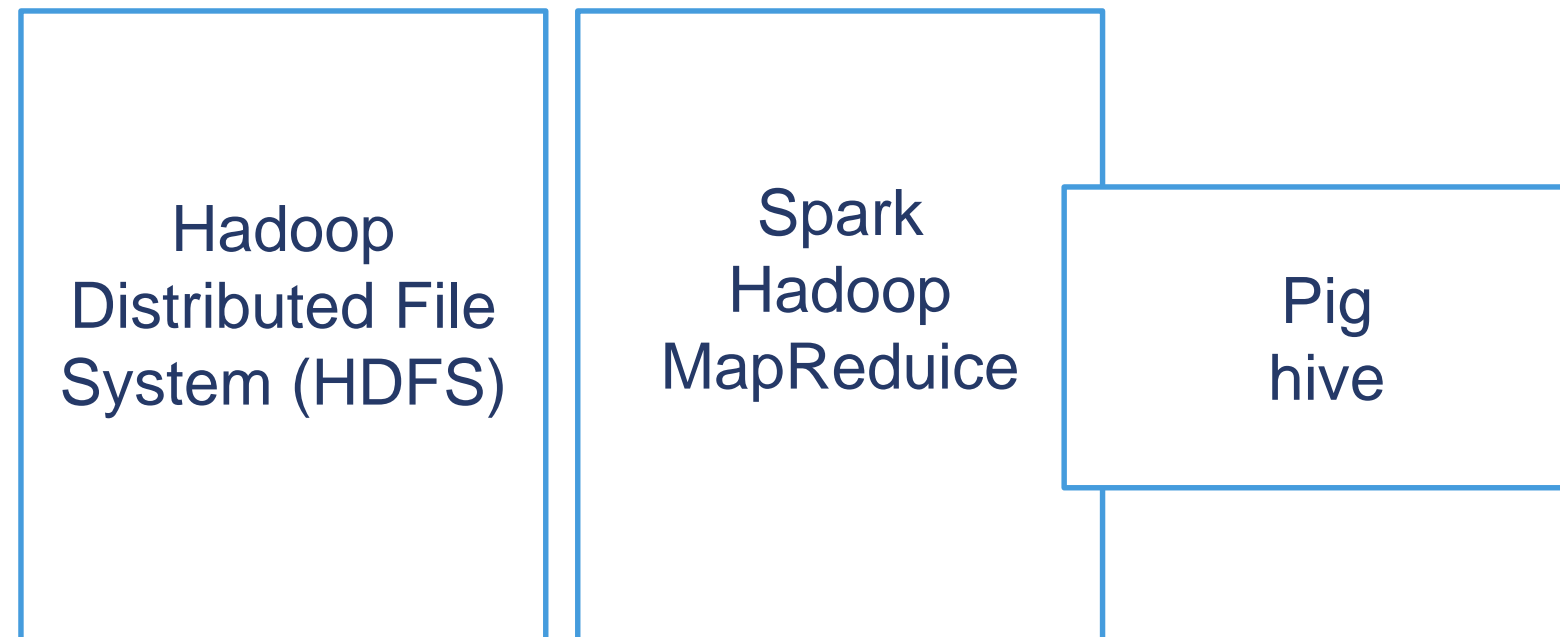
Solución

Traer el programa a los datos en lugar de llevar los datos al programa



Distribuir datos cuando los datos estén almacenado
Correr los cálculos computacionales donde estén los datos

Procesamiento Hadoop



Conceptos básicos

- **Cluster, Grupo de computadores trabajando en conjunto**

Administración de recursos, almacenamiento de datos y procesamiento

- **Nodo, un computador individual del cluster**

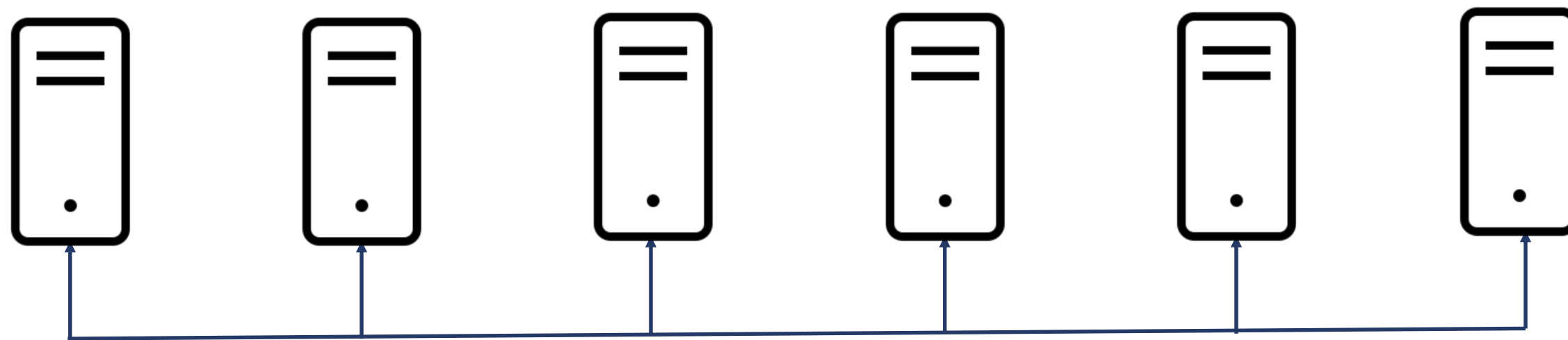
Computador que administran y computadores que realizan el trabajo

- **Daemon, es un programa**

Proceso que se ejecuta en segundo plano para hacer tareas esenciales

Hadoop cluster

Un clúster de Hadoop consiste en una colección de máquinas (**nodos**) que están conectadas en red para realizar cálculos paralelos en grandes cantidades de datos. Está diseñado para funcionar **en hardware de bajo costo** y permite escalar añadiendo o quitando nodos

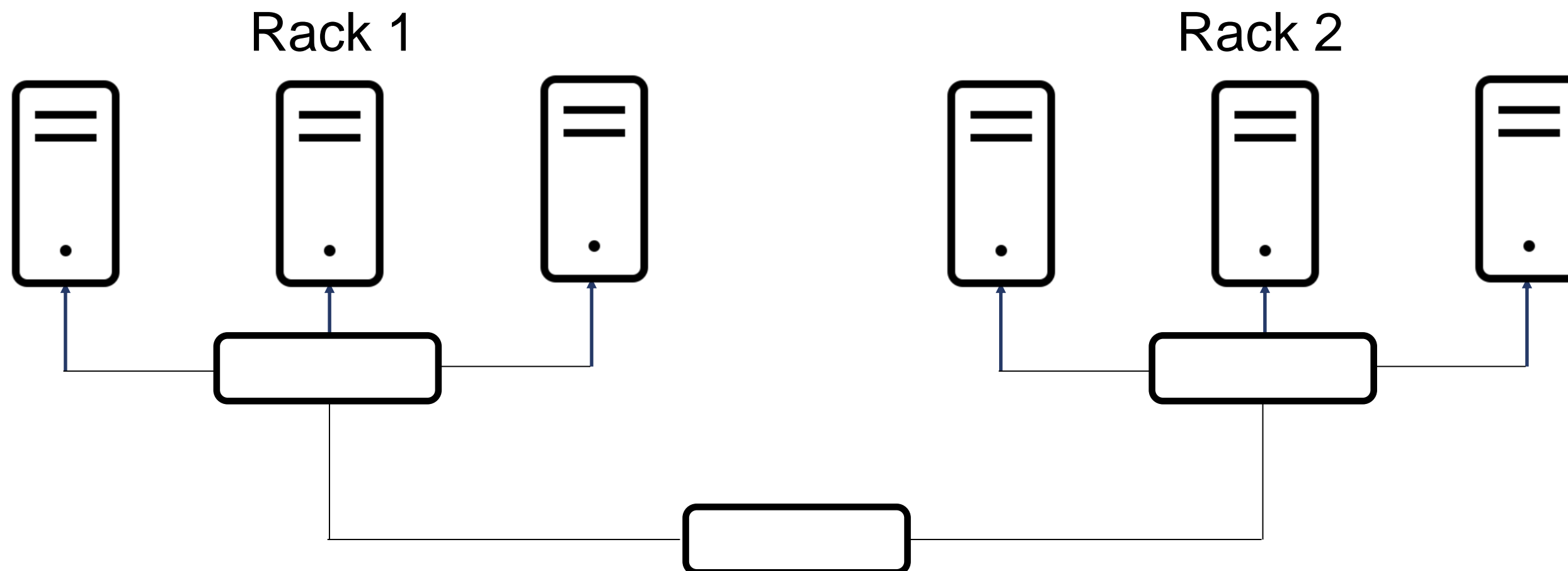


Hadoop cluster Topología de red

Los clústeres de Hadoop de tamaño mediano-grande, los nodos están conectados en una topología de red de dos o tres niveles::

- Los Nodos son instalados en un **rack** e interconectados usando a switch
- **Racks** son interconectados usando switch

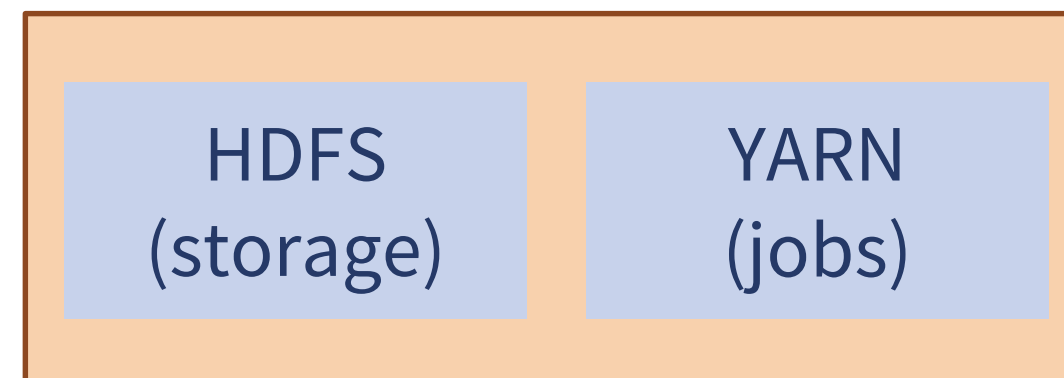
Hadoop es un **rack-aware**: Es importante entender la ubicación y la topología de red .



Componentes de Hadoop

Hadoop define 2 principales componentes:

- **HDFS** (*Hadoop Distributed File System*) Es un sistema de **archivos distribuido** altamente tolerante a fallos diseñado para ser implementado en hardware de bajo costo. Ofrece un alto rendimiento y es adecuado para grandes cantidades de datos.
- **YARN** (*Yet Another Resource Negotiator*) Un planificador de **trabajos distribuidos** encargado de programar y monitorear trabajos en todo el clúster. MapReduce es el tipo principal de trabajo, pero YARN soporta otros tipos.



Arquitectura de Hadoop cluster

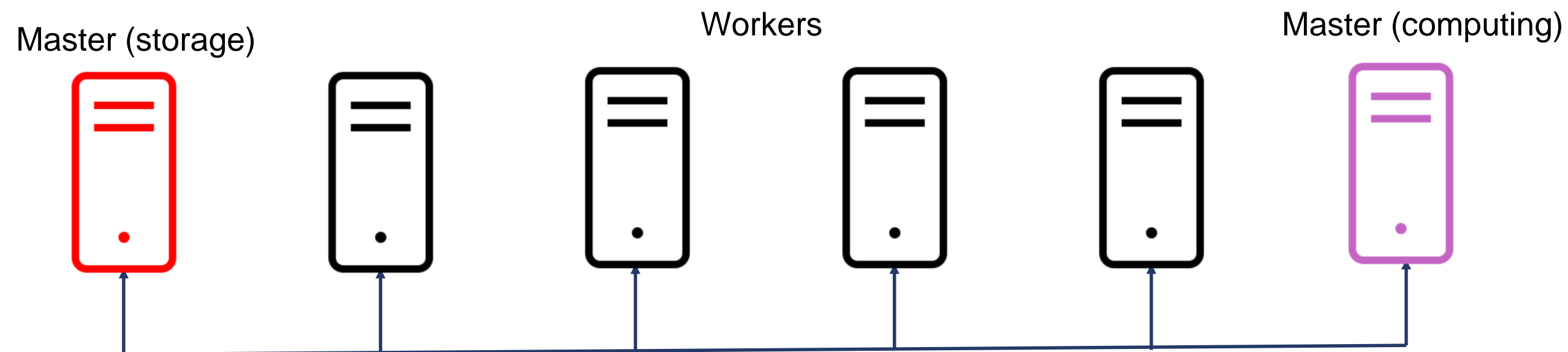
Los recursos de almacenamiento y cómputo en un clúster de Hadoop son gestionados mediante una arquitectura de maestro-trabajador (**master-worker**).

El **master**

- Es consciente del estado de los nodos trabajadores y se comunica con ellos,
- recibe solicitudes externas y decide quién ejecuta qué y cuándo.

Los **workers**

- Reportar el trabajo al maestro
- ejecutan las tareas ordenadas por maestro



Hadoop daemons

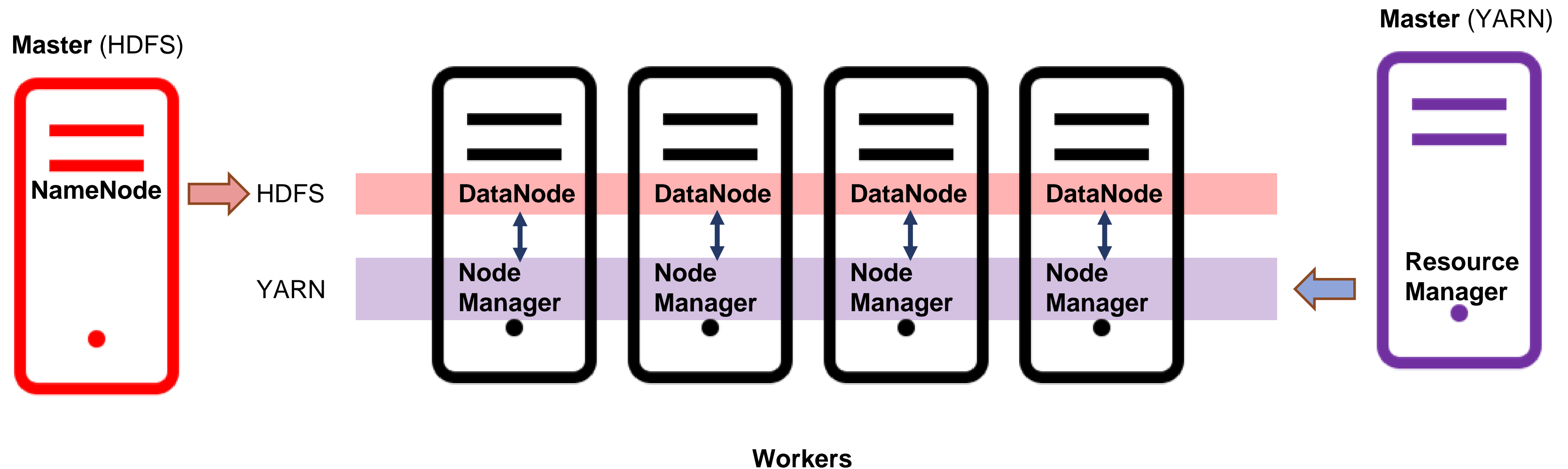
Hadoop define los demonios **master** y **worker** para HDFS y YARN.

In **HDFS** se tiene los siguientes demonios :

- **NameNode**: corre en el nodo HDFS master
- **DataNode**: corre en el nodo worker

In **YARN** tiene los siguientes demonios:

- **ResourceManager**: corre en el nodo YARN master
- **NodeManager**: corre en el nodo worker

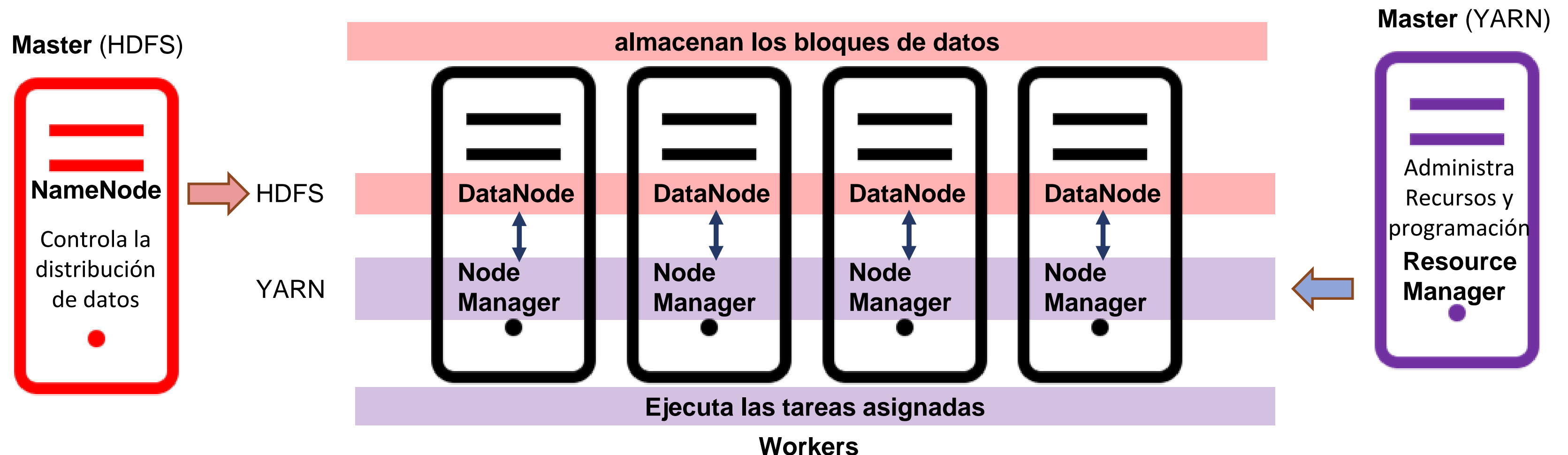


Hadoop daemons: “bring computation to data”

Los demonios **master** y **workers** se ejecutan en nodos diferentes.

Para implementar el principio de "traer cálculo a los datos", los **DataNode** ejecutan tanto:

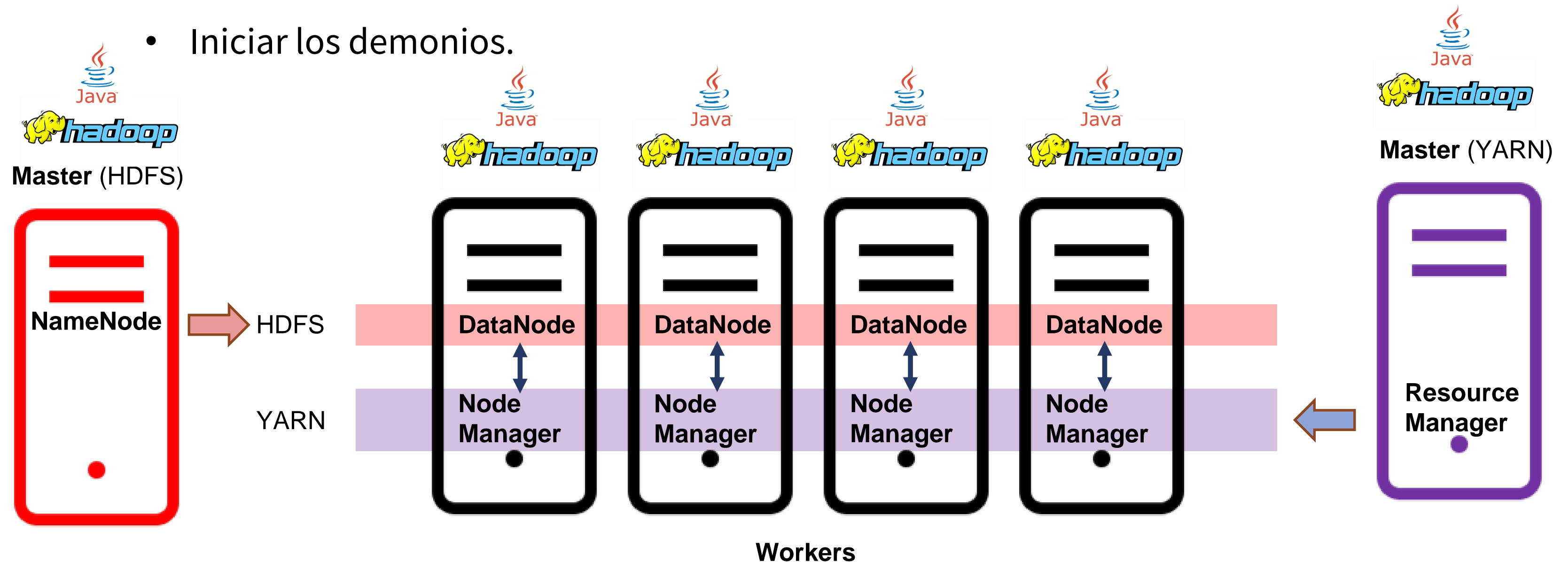
- El demonio HDFS DataNode (para el almacenamiento de datos)
- El demonio NodeManager YARN (para la ejecución de trabajos).



Crear un Hadoop cluster

Para tener un clúster de Hadoop en funcionamiento es necesario:

- Instalar Java, Hadoop está implementado en Java
- Descargar Hadoop en cada nodo
- Configurar Hadoop (IP y Cantidad de recursos)
- Iniciar los demonios.



Configuración de Hadoop cluster

Hadoop puede correr en tres modos diferentes:

Modo standalone: Hadoop se ejecuta en un único nodo y no está integrado con ninguna infraestructura distribuida.

Modo pseudo-distribuido: En este modo, Hadoop se ejecuta en un solo nodo pero simula una configuración de cluster distribuido. Pequeñas pruebas de concepto.

Modo distribuido: Este es el modo recomendado para producción y permite a Hadoop trabajar con un cluster de múltiples nodos. En este modo, los datos se dividen y se procesan en paralelo en diferentes nodos del cluster, lo que permite un procesamiento más rápido y escalable.

los tres modos difieren en cuanto a la configuración

Creando un Hadoop cluster

Hay dos formas principales de tener un clúster de Hadoop:

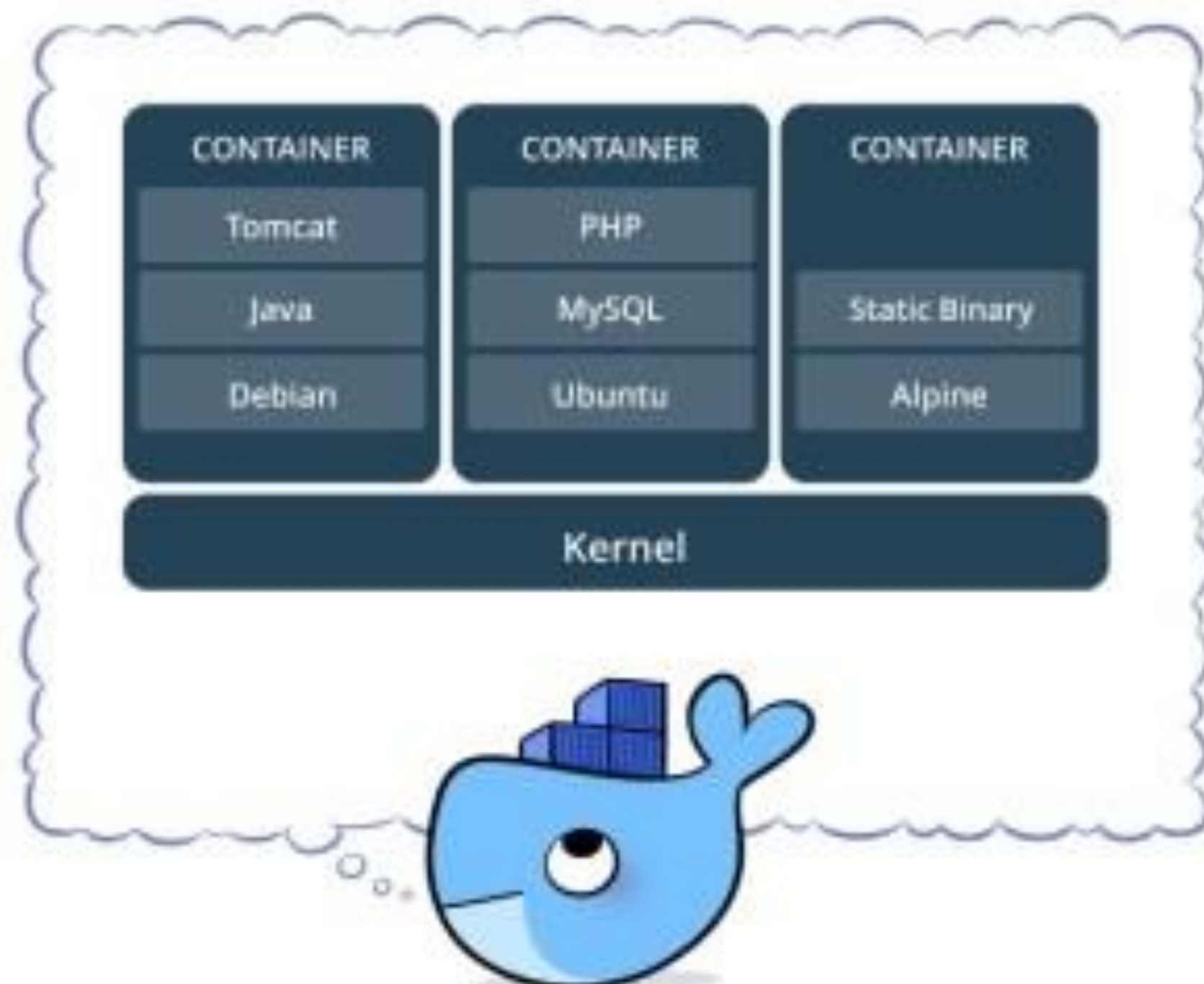
- Construirlo nosotros mismos (y acá de muchas maneras)
- Usar un servicio de clúster de gran datos basado en la nube (AWS EMR, Google Dataproc, Azure HDInsight). Pueden ser administrados, lo que significa que la plataforma en la nube hace todo esto por nosotros

Contenidos

- Fundamentos de Hadoop
- **Despliegue de Hadoop**
- HDFS

Docker para hadoop

1. Portable
2. Inmutable
3. Ligero



Fuente: <https://geekflare.com/es/docker-vs-virtual-machine/>

crear, desplegar y ejecutar aplicaciones en contenedores

Instalar Docker Engine

Tecnología que permite **la creación, el despliegue y la administración** de aplicaciones en contenedores, lo que facilita el desarrollo y la implementación de aplicaciones en entornos complejos y heterogéneo

1. En su equipo instalar **Docker Engine**, seguir la guía:

<https://docs.docker.com/engine/install/ubuntu/>

2. Configurar los permisos del Docker

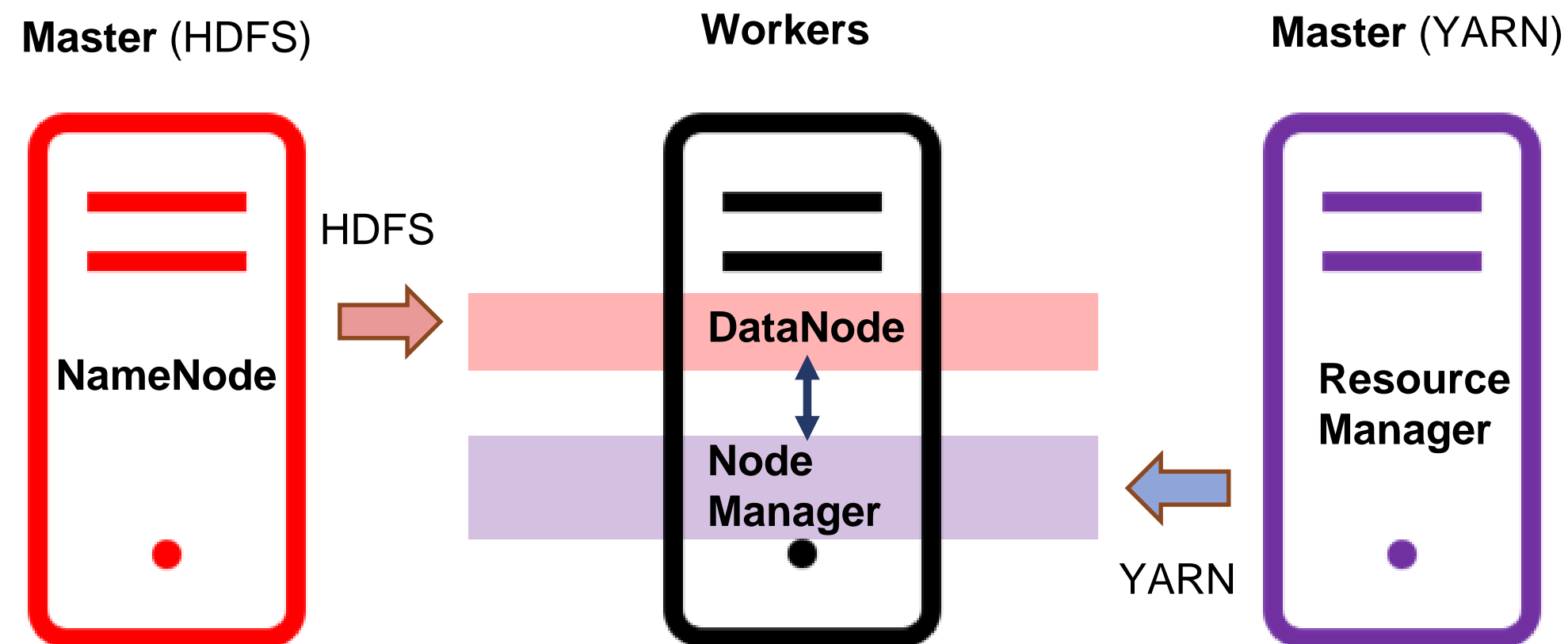
<https://docs.docker.com/engine/install/linux-postinstall/>

3. Reinicia el sistema operativo

5 minutos

Descargue el Docker Hadoop

1. Ingrese al Git del curso y descarga el docker: https://github.com/labDatascience/docker_hadoop_UR
2. Analizar el fichero **Dockerfile** ¿Qué observan?
3. Analizer el fichero **docker-compose.yml**
4. En la terminal ejecutar **docker-compose up -d** ¿Este comando que hace ?
5. Cuantos contenedores, están en ejecución? Que comando se uso ? El resultado es el de la imagen?



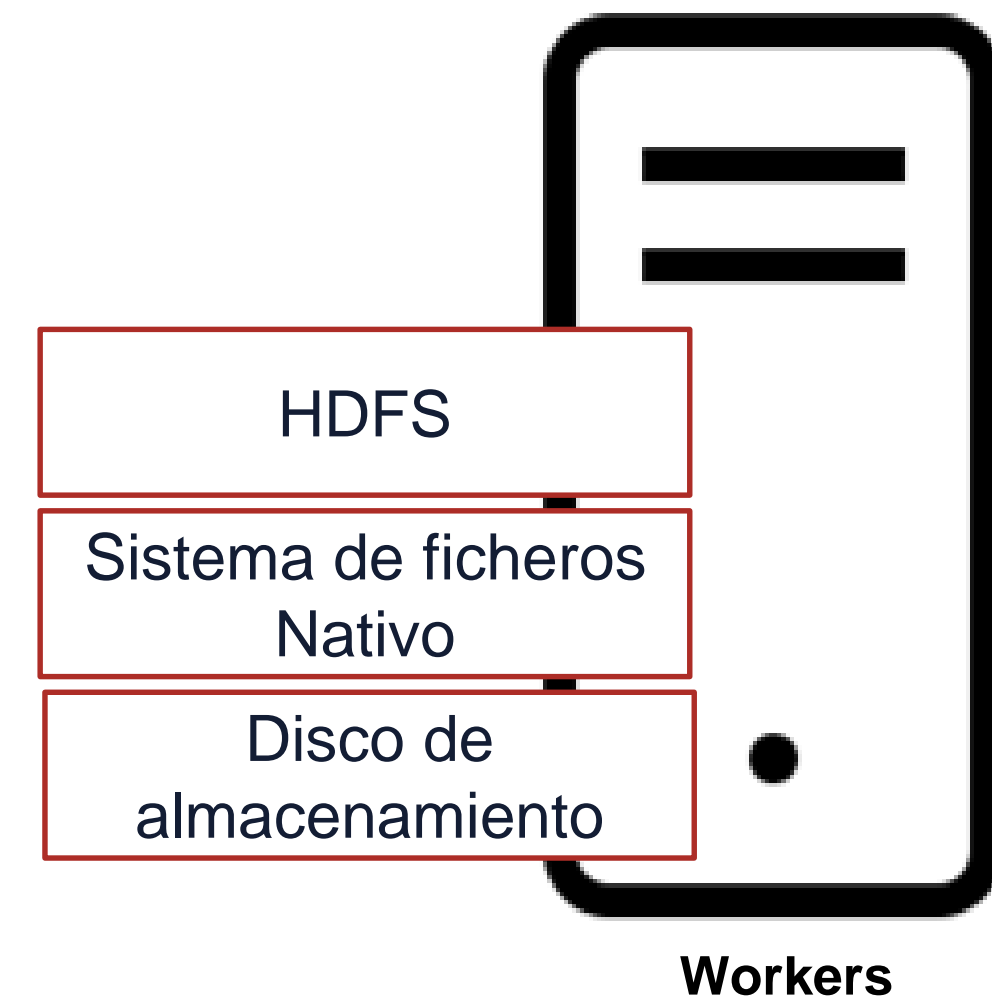
Contenidos

- Fundamentos de Hadoop
- Despliegue de Hadoop
- **HDFS**

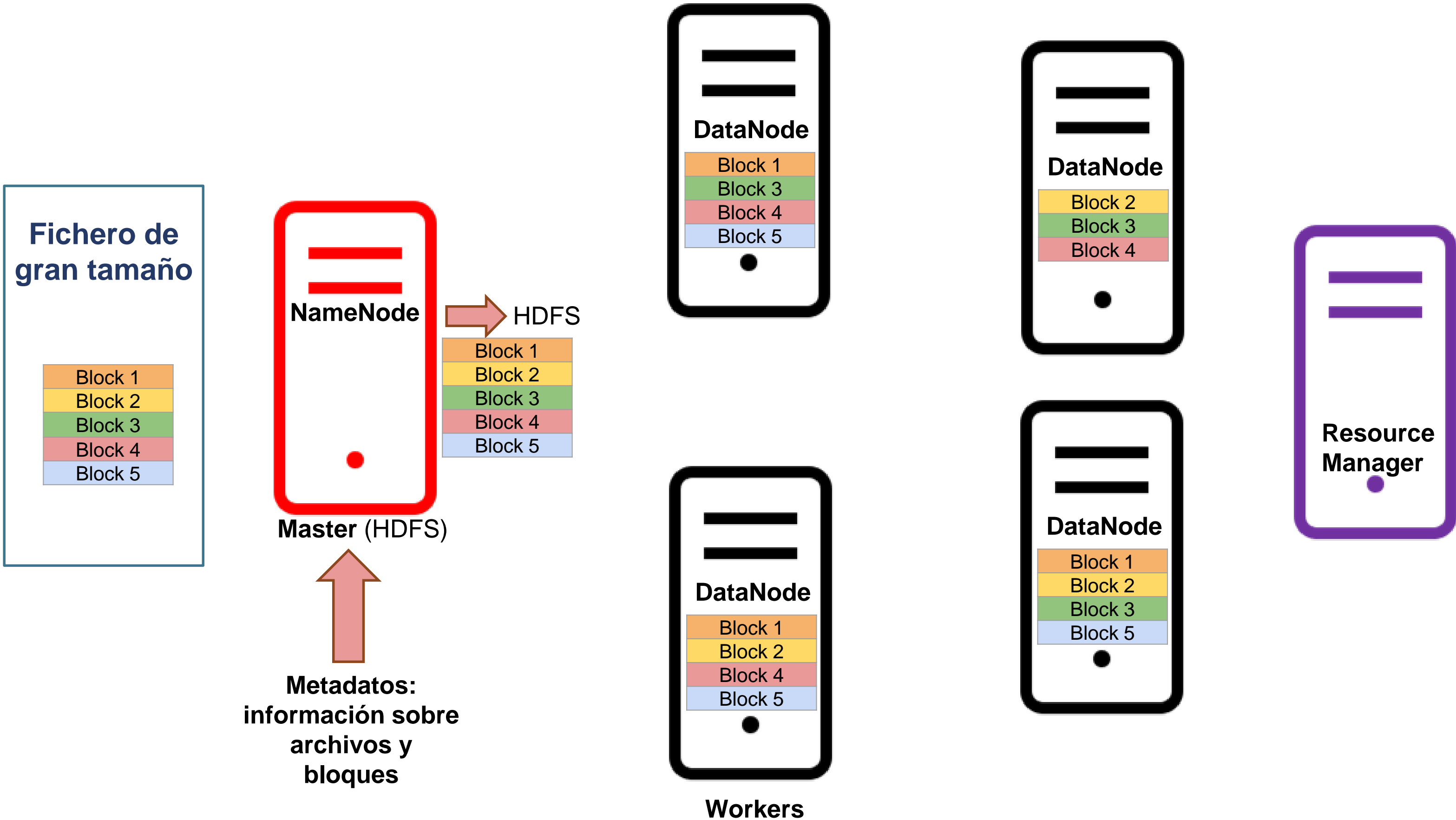
The Hadoop Distributed File System HDFS

Sistema de archivos **distributed** inspirado en el Sistema de Archivos de Google. Está diseñado para almacenar en hardware de bajo costo archivos muy grandes que se espera que se **escriban una vez y se lean muchas veces**. Un archivo en HDFS se divide en una secuencia de unidades conocidas como bloques:

- Los bloques se almacenan de manera **independiente** en nodos del clúster.
- Para reducir el número de búsquedas, los bloques son grandes, típicamente de 128 MB.
- Por defecto, los bloques **se replican tres veces en** diferentes nodos del clúster
- HDFS se sitúa encima de un sistema de archivos nativo. ext4, ext3



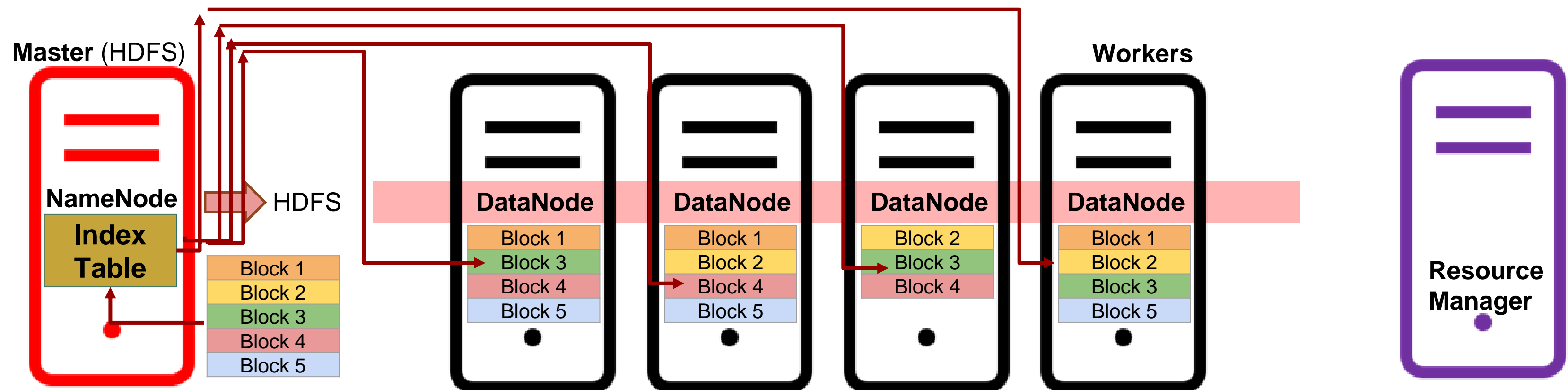
The Hadoop Distributed File System HDFS



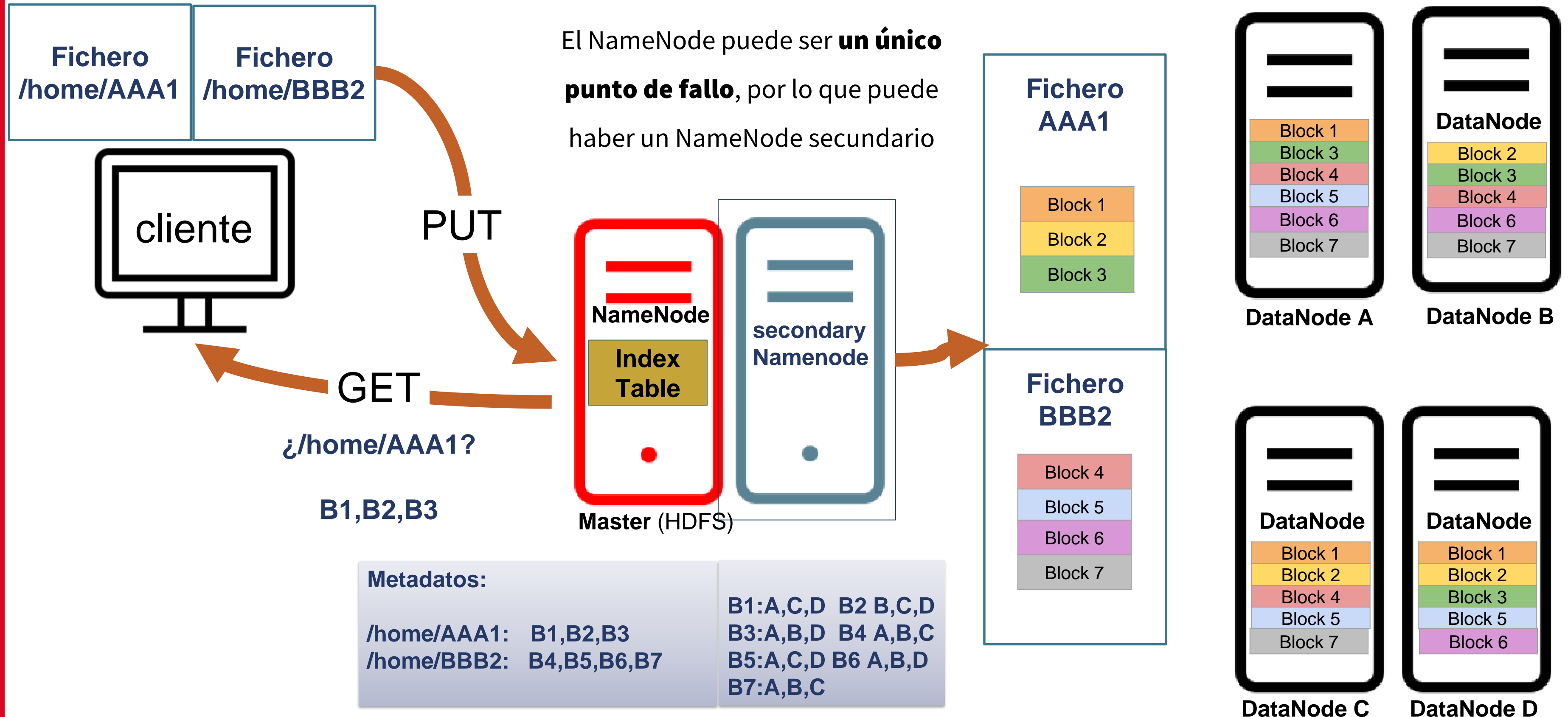
El demonio de HDFS master : NameNode

El demonio **NameNode** de HDFS es responsable de administrar el espacio de nombres del sistema de archivos. Esto incluye:

- Mantener el **árbol del sistema** de archivos
- Tener la **Tabla de índice** que contiene la ubicación (DataNodes) de cada bloque de archivo.
- Responsable de controlar el **acceso a los archivos y directorios**, y de asegurar que se mantenga la coherencia del sistema de archivos.



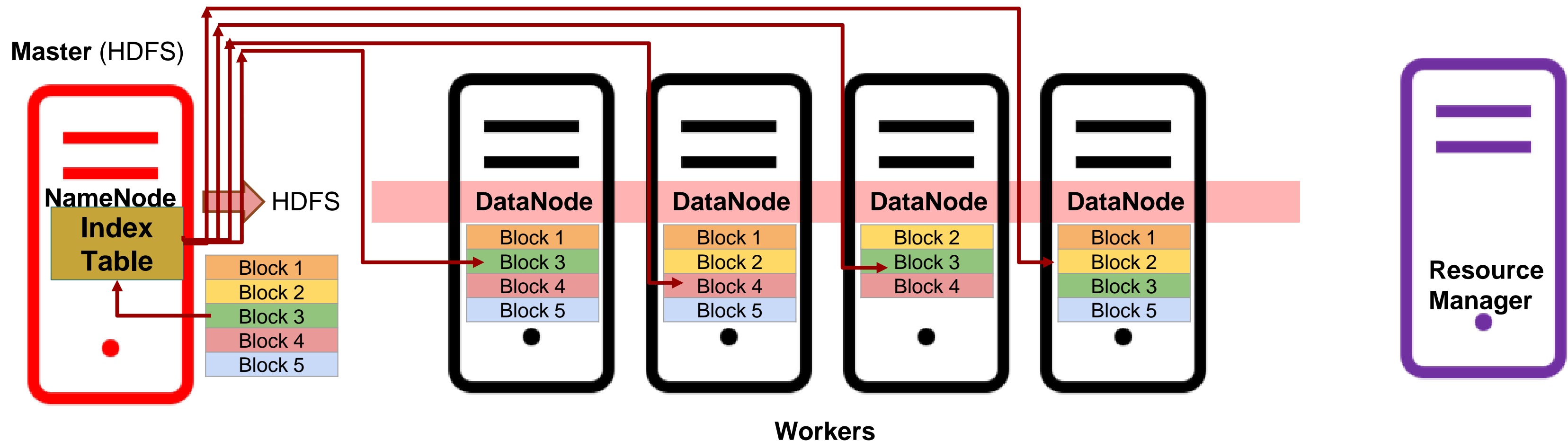
El demonio de HDFS master : NameNode



El demonio de HDFS worker : DataNode

El HDFS **DataNode** es el demonio de worker:

- **Almacena y recupera** bloques
- Informa periódicamente al NameNode con una lista de bloques que almacena
- Implementa almacenamiento en caché de bloques que se acceden con frecuencia



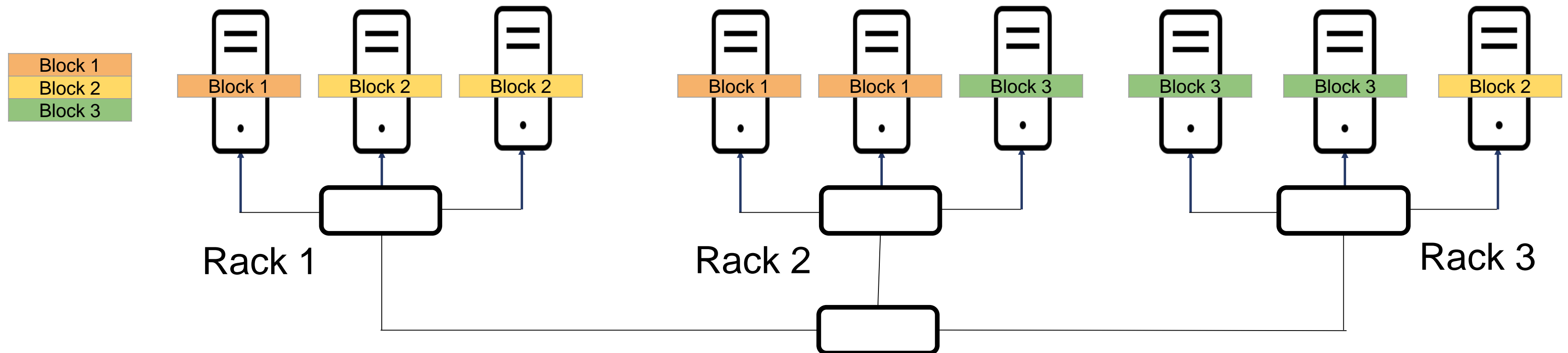
Los bloque de replicas de HDFS

Por defecto, se almacenan 3 réplicas de cada bloque de datos de archivo en diferentes nodos del clúster.

La ubicación de estas réplicas se determina mediante el algoritmo de colocación de réplicas de HDFS.

Por lo general:

1. La primera réplica en el nodo donde se está escribiendo el archivo (nodo de escritor)
2. Las otras dos réplicas en nodos diferentes, preferiblemente en un rack diferente. Disponibilidad y tolerancia a fallos en caso de fallo del rack o problemas de red.



Acceso a HDFS

El sistema de ficheros de Hadoop (HDFS) se puede manejar a través de tres interfaces:

- Interfaz de línea de comandos, mediante el comando **hadoop fs [opciones]**.
- Interfaz web (puerto 50070 del NameNode).
- El API de programación
 - Java API: for java applications
 - HTTP REST API : for non-java applications

La shell del sistema de archivos (FS) es una aplicación que proporciona a los usuarios la CLI típica para interactuar con HDFS, por ejemplo

```
bin/hadoop fs -mkdir /foodir
```

CLI a HDFS

Comando	Acción
<code>hadoop fs -ls <path></code>	Lista ficheros
<code>hadoop fs -cp <src> <dst></code>	Copia ficheros HDFS a HDFS
<code>hadoop fs -mv <src> <dst></code>	Mueve ficheros HDFS a HDFS
<code>hadoop fs -rm <path></code>	Borra ficheros en HDFS
<code>hadoop fs -rmr <path></code>	Borra recursivamente
<code>hadoop fs -cat <path></code>	Muestra fichero en HDFS
<code>hadoop fs -mkdir <path></code>	Crea directorio en HDFS
<code>hadoop fs -chmod ...</code>	Cambia permisos de fichero
<code>hadoop fs -chown ...</code>	Cambia propietario/grupo de fichero
<code>hadoop fs -put <local> <dst></code>	Copia de local a HDFS
<code>hadoop fs -get <src> <local></code>	Copia de HDFS a local

Operaciones HDFS

Para leer un archivo en HDFS:

- Las aplicaciones obtienen del NameNode las direcciones de los DataNodes que tienen los bloques de archivo requeridos.
- Los DataNodes se ordenan según su proximidad: si la aplicación se ejecuta en un nodo, obtendrá bloques locales.

Para escribir un archivo en HDFS:

- El NameNode selecciona una lista de DataNodes para almacenar las réplicas.
- Para cada bloque, se copia la primera réplica (generalmente en el nodo de la aplicación).
- La segunda réplica se copia desde la primera y la tercera desde la segunda. La segunda y tercera réplicas se encuentran en el mismo rack para reducir el tráfico de red.

Ejercicio con HDFS en el docker

Objetivo

1. Comprender cómo acceder a HDFS para manejar la entrada y salida de trabajos en el clúster Hadoop
2. Ejecutar trabajos en un clúster Hadoop

Documentación: <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/FileSystemShell.html>

Proceso.

1. Lance hadop docker, si no recuerda ver diapositiva 23
2. Interactúe con la CLI de HDFS, invocando la forma.

¿Cuál es la configuración del cluster?

```
hadoop fs -<command> options
```

La lista completa de comandos la puedes encontrar usando

```
hadoop fs -help
```

3. Igual que el lab MapReduce, cree en HDFS la carpeta input

```
hadoop fs -mkdir input
```

4. Copia el dataSet del lab MapReduce. La transferencia de archivos desde un sistema de archivos local a Hadoop se realiza utilizando el comando '-put' o '-copyFromLocal'

Donde están los datos en local ?
Cuál es el comando ?
Como sabe que ya tiene el texto en HDFS

WordCount con HDFS en el docker

Objetivo

1. Correr un trabajo del lab MapReduce en el cluster de Hadoop

Proceso.

1. Una vez que se hayan copiado sus datos en HDFS, puede ejecutar su trabajo MapReduce . Usa las carpetas Input/Out

```
python3 wordcount.py -r hadoop --output-dir out --no-cat-output hdfs://user/root/input
```

2. Analiza el comando anterior, ¿se ejecuto el trabajo?
3. Verifique el contenido de la carpeta 'out' con el comando 'ls'
4. ¿Cuántos Reduce han intervenido en el proceso?
5. ¿Cómo retornamos el resultado a la carpeta local ?