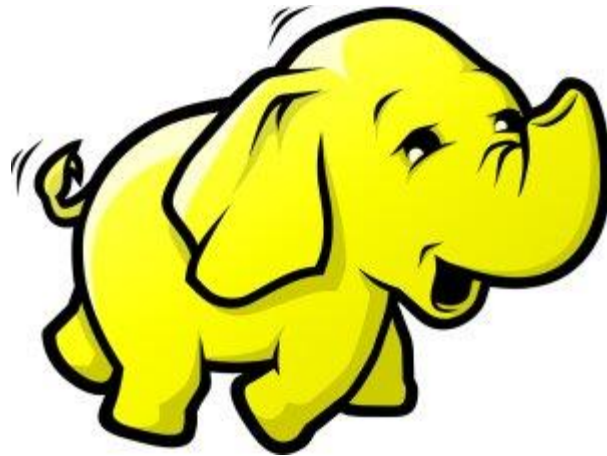




Say hello to Hadoop



HDFS

مقیاس پذیری:

- مزیت مهم هادوپ است
- مقیاس پذیری افقی

Hadoop

مقیاس پذیری:

- **مقیاس پذیری عمودی (Scale up):** ظرفیت سخت افزاری را افزایش می دهیم. به بیان دیگر ، RAM, CPU و یا دیسک بیشتری به سیستم اضافه می کنیم.
- چالش ها:
 - همیشه در مورد افزایش سخت افزار محدودیت هست و نمی توان منابع دستگاه را افزایش داد
 - باید در اکثر مواقع ابتدا ماشین را خاموش کرد. سپس برای داشتن مجموعه سخت افزاری منابع را افزایش داد و ماشین را دوباره راه اندازی کرد.

Hadoop

مقیاس پذیری:

- **مقیاس پذیری افقی (Scale out):** به جای افزایش ظرفیت سخت افزار هر دستگاه، گره های بیشتر را اضافه می کنیم.

- مزایا:

- می توانیم دستگاه های بیشتری که در حال فعالیت هستند را بدون متوقف کردن سیستم اضافه کرد.
- هیچ زمانی نیاز نیست به منظور افزایش یا کاهش مقیاس کلاستر، عملکرد آنرا متوقف کرد.

HDFS

- در HDFS می توان از یک سخت افزار ارزان قیمت مانند دسکتاپ استفاده کرد تا ویژگی های سرورهای گران قیمت اجرا شوند.
- چون از سخت افزار ارزان قیمت استفاده می کنیم برای مقیاس پذیری افقی نیاز به هزینه زیاد نداریم.
- به بیان دیگر:
اضافه کردن گره های بیشتر به HDFS (مقیاس پذیری افقی) از نظر اقتصادی مقرون به صرفه است.

Hadoop Distributed File System

- یک سیستم فایل توزیع شده که با جاوا نوشته شده برای فرم ورک Hadoop
- فایل های بزرگ (در حد گیگا تا ترابایت) را در ماشین های مختلف ذخیره می کند
- **قابلیت اطمینان:** بوسیله (replicate) داده در چند میزبان، قابلیت اطمینان حاصل می شود
- به صورت پیش فرض داده ها روی ۳ گره ذخیره می شود که دو تا روی یک rack و دیگری در یک rack متفاوت واقع شده اند.

معماری HDFS

- **HDFS** یک سیستم فایل با ساختار بلاکی است که در آن هر فایل به بلاک‌هایی با اندازه‌های از پیش تعیین شده تقسیم می‌شود. این بلاک‌ها بصورت توزیع شده در چندین ماشین خوشه ذخیره می‌شوند.



معماری HDFS

- گره Master : سرویس Name node
- گره های Slave : سرویس Data node



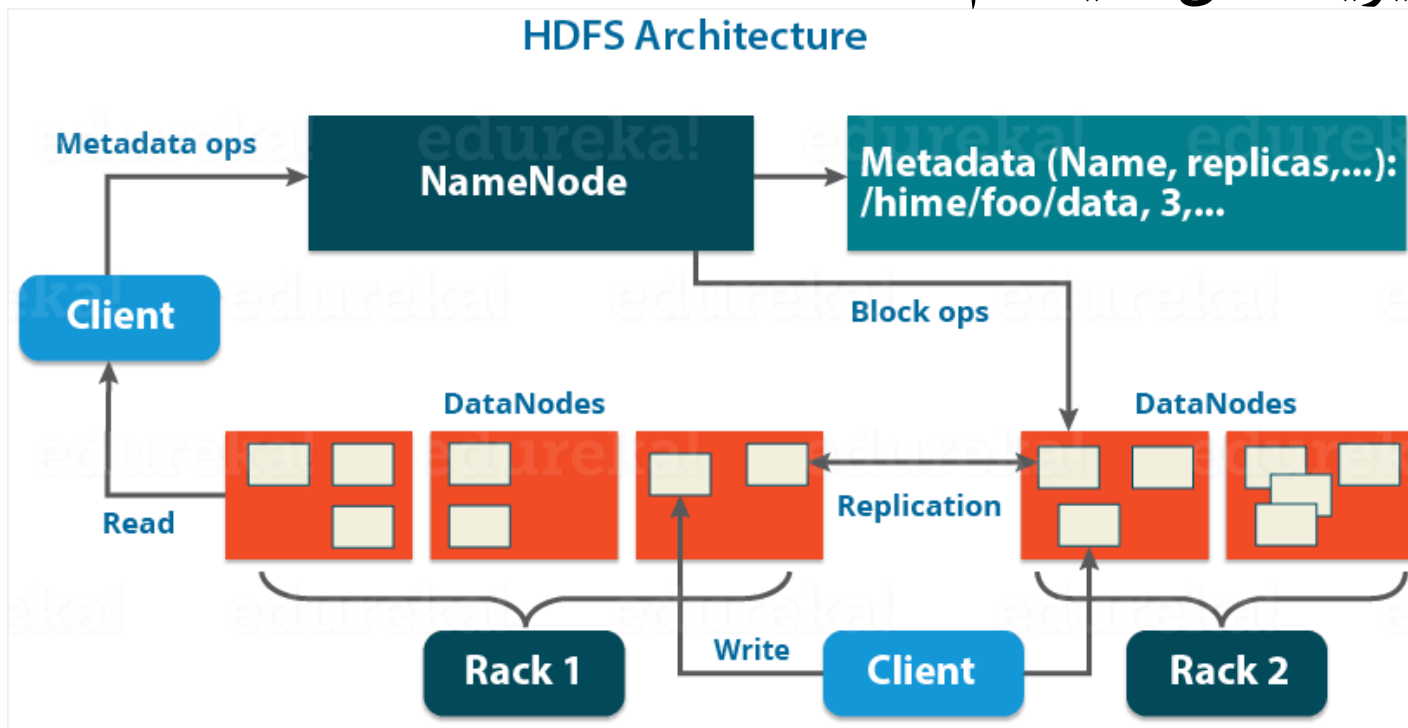
Hadoop Distributed File System

- یک سیستم فایل توزیع شده که با جاوا نوشته شده برای فرم ورک Hadoop
- فایل های بزرگ (در حد گیگا تا ترابایت) را در ماشین های مختلف ذخیره می کند
- **قابلیت اطمینان:** بوسیله (replicate) داده در چند میزبان، قابلیت اطمینان حاصل می شود
- به صورت پیش فرض داده ها روی ۳ گره ذخیره می شود که دو تا روی یک rack و دیگری در یک rack متفاوت واقع شده اند.

معماری HDFS

:Name node

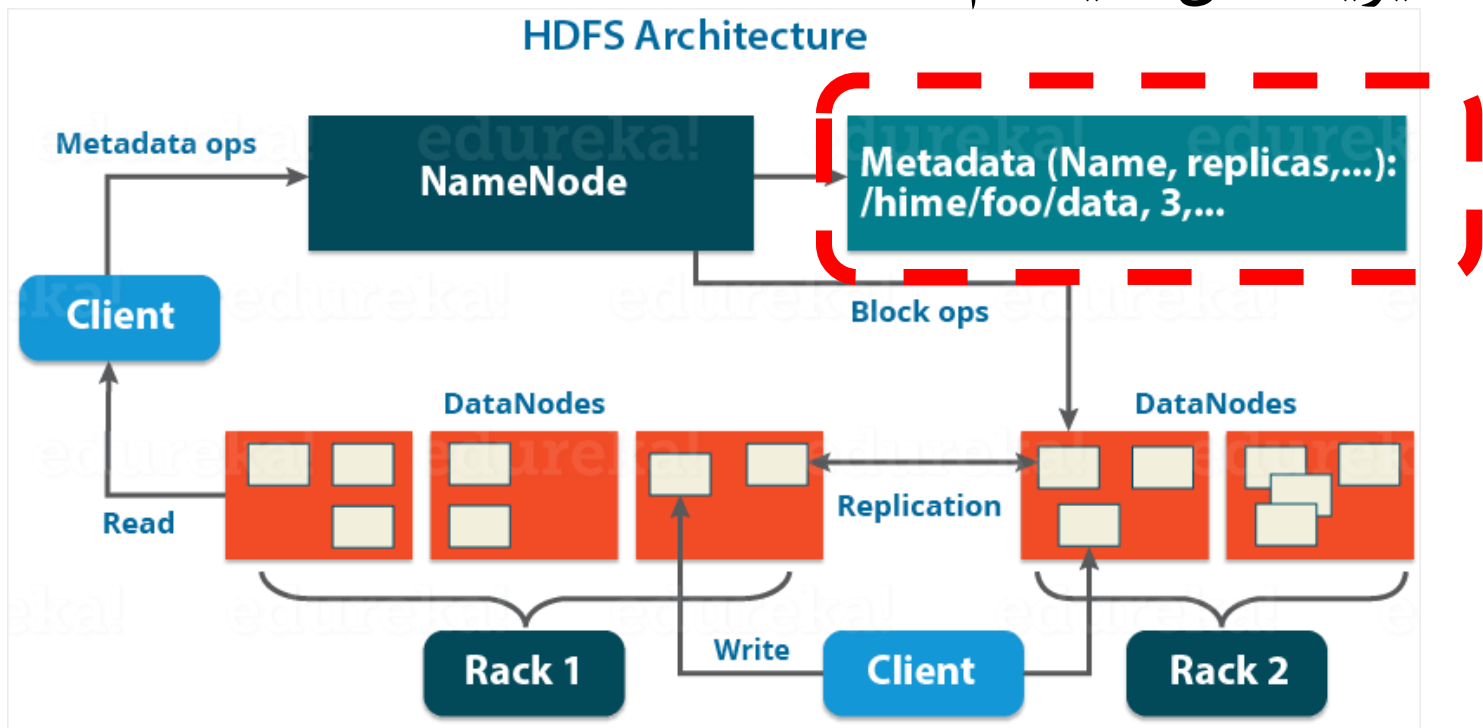
- سرویس مدیریتی در HDFS
- مدیریت تمام گره های slave
- مدیریت کل سیستم HDFS



معماری HDFS

:Name node

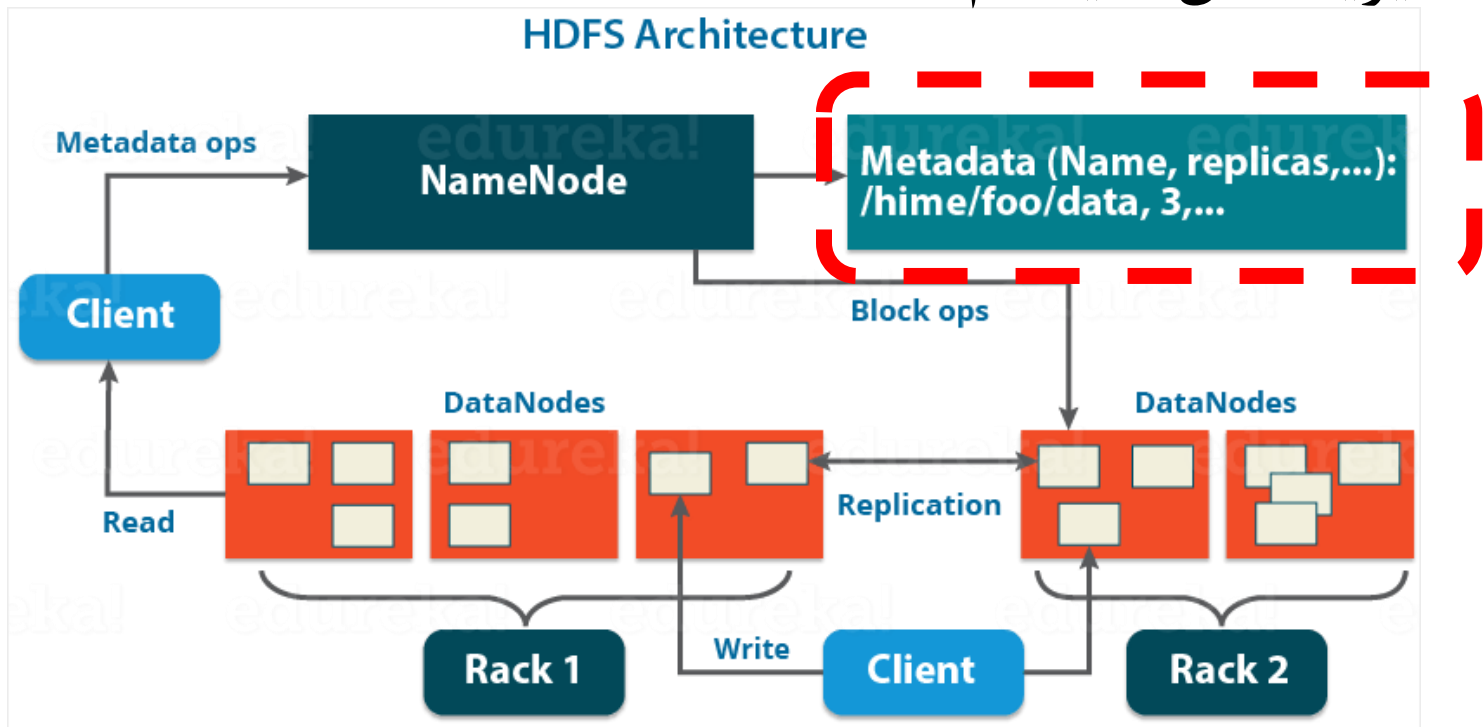
- سرویس مدیریتی در HDFS
- مدیریت تمام گره های slave
- مدیریت کل سیستم HDFS



معماری HDFS

:Name node

- سرویس مدیریتی در HDFS
- مدیریت تمام گره های slave
- مدیریت کل سیستم HDFS



معماری HDFS

وظایف NameNode:

- بصورت یک سرویس در گره Master بوده که مدیریت و نگهداری از DataNode ها را انجام می دهد.
- ذخیره سازی متاداده های تمام فایل ها، برای مثال محل ذخیره سازی بلاک ها ، اندازه فایل ها، دسترسی ها، ساختار درختی و غیره.
- از تمام تغییرات در سیستم فایل باخبر می شود و **متاداده** مرتبط با آن **تغییرات** را **بروزرسانی** می کند.
- این سرویس دائماً پیام های گزارش را از گره های Slave در کلاستر را دریافت می کند تا از وضعیت لحظه ای آنها باخبر شود.
- سرویس NameNode لیست تمام بلاک های سیستم فایل را در خود دارد و می داند هریک از این بلاک ها در کدام گره ذخیره شده اند.

معماری HDFS

وظایف NameNode:

متادیتا:

– Fslmage: این فایل شامل وضعیت کامل فضای نام سیستم فایل از ابتدای راه اندازی خوشه HDFS است.

– EditLogs: این فایل شامل تمام تغییرات ایجاد شده در سیستم فایل HDFS پس از آخرین تغییرات موجود در فایل Fslmage می باشد.

معماری HDFS

وظایف NameNode:

- دائماً پیام‌های گزارش بلاک و **ضربان قلب** را از گره‌های Slave در خوشه دریافت می‌کند تا از **وضعیت لحظه‌ای** آنها باخبر شود.
- **لیست** تمام بلاک‌های سیستم فایل را در خود دارد و می‌داند هریک از این بلاک‌ها در **کدام گره** ذخیره شده‌اند.
- **مدیریت تکرار** تمام بلاک‌ها در سیستم فایل را برعهده دارد.

معماری HDFS

وظایف NameNode:

– در صورت خرابی هریک از گره‌های Slave، NameNode تصمیم می‌گیرد که در یک Slave جدید یک کپی دیگر از داده‌های از دست رفته ایجاد کند تا سطح تکرار بلاک‌ها حفظ شود. همچنین برقرار سطح توازن داده و ترافیک شبکه در سطح خوشه نیز از دیگر وظایف سرویس NameNode بشمار می‌رود.

معماری HDFS

: DataNode

- DataNode ها، گره‌های Slave در سیستم HDFS هستند.
- برخلاف NameNode؛ برای گره‌های Slave می‌توان از سخت‌افزارهای **مقرون بصرفه** استفاده کرد.
- در واقع گره Slave یک ماشین سرور است که داده‌ها را در سیستم فایل **محلی** خود ذخیره می‌کند.

معماری HDFS

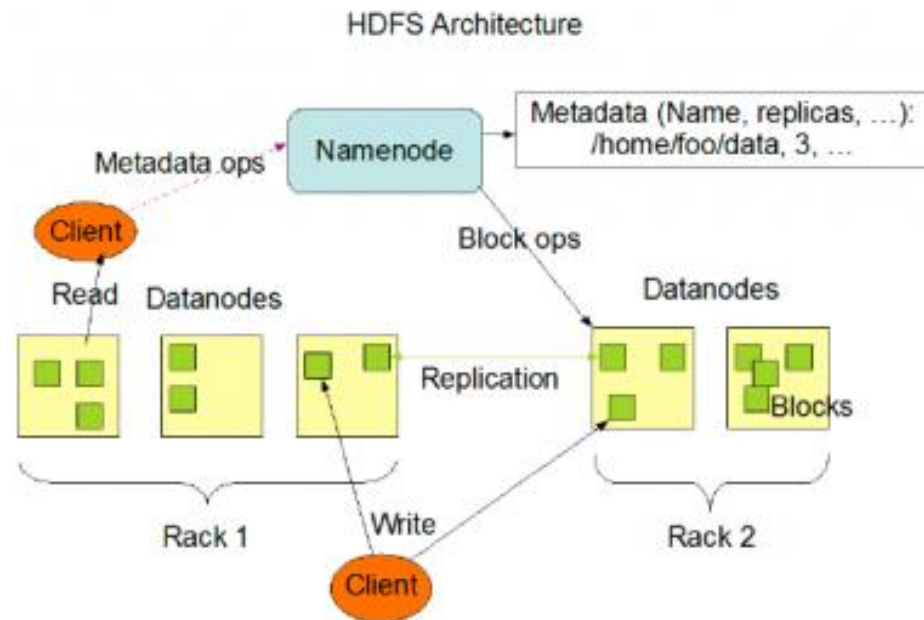
عملکردهای سرویس DataNode :

- DataNode سرویسی است که بروی هرکدام از ماشین‌های Slave اجرا می‌شود.
- داده‌های واقعی HDFS در گره‌های Slave ذخیره می‌شوند.
- سرویس DataNode درخواست‌های خواندن یا نوشتن از کلاینت‌ها را اجرا می‌کند.
- بصورت متناوب **ضربان قلب** را به NameNode ارسال می‌کند تا سلامتی کلی HDFS را گزارش کند. بصورت پیش‌فرض، این تناوب برای هر ۳ ثانیه تعیین شده است.

HDFS

HDFS Terminology

- Namenode
- Datanode
- DFS Client
- Files/Directories
- Replication
- Blocks



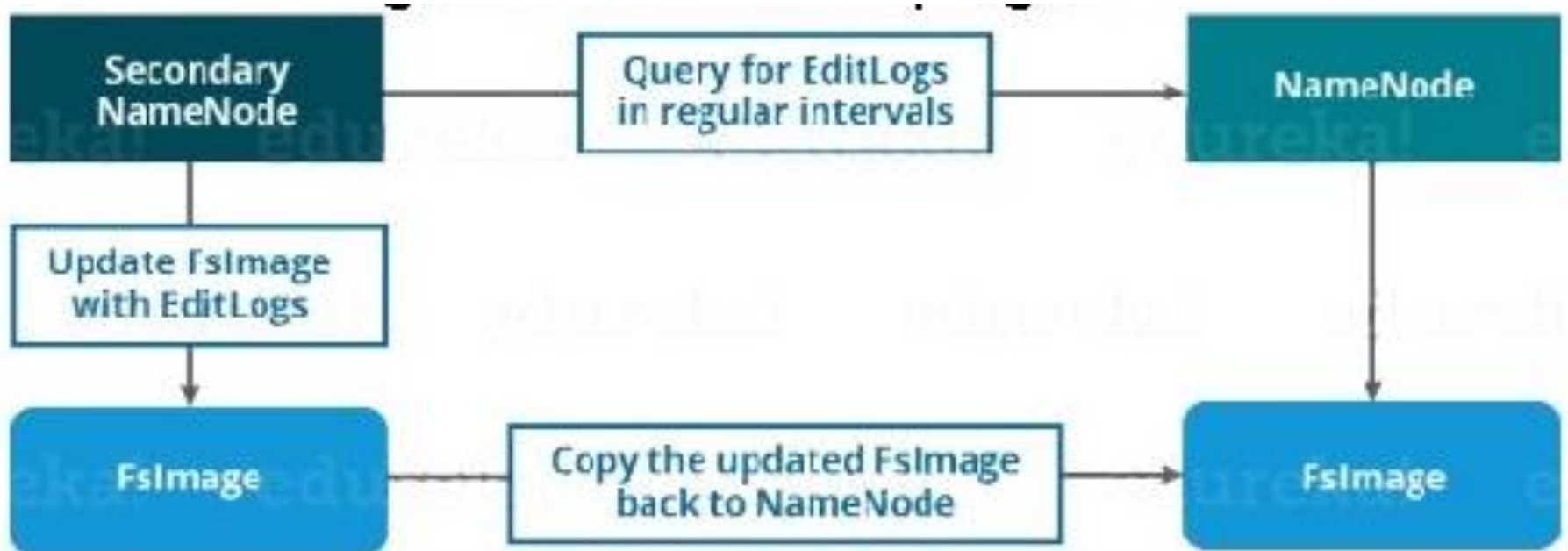
HDFS

Single Point of Failure

- با افزودن یک نود به نام **Secondary Name Node** قابلیت دسترسی پذیری بالاتری ارائه شده است.
- نگهداری یک تصویر (Snapshot) از Primary Name Node
- استفاده از آن در صورت بروز مشکل در Name Node اولیه

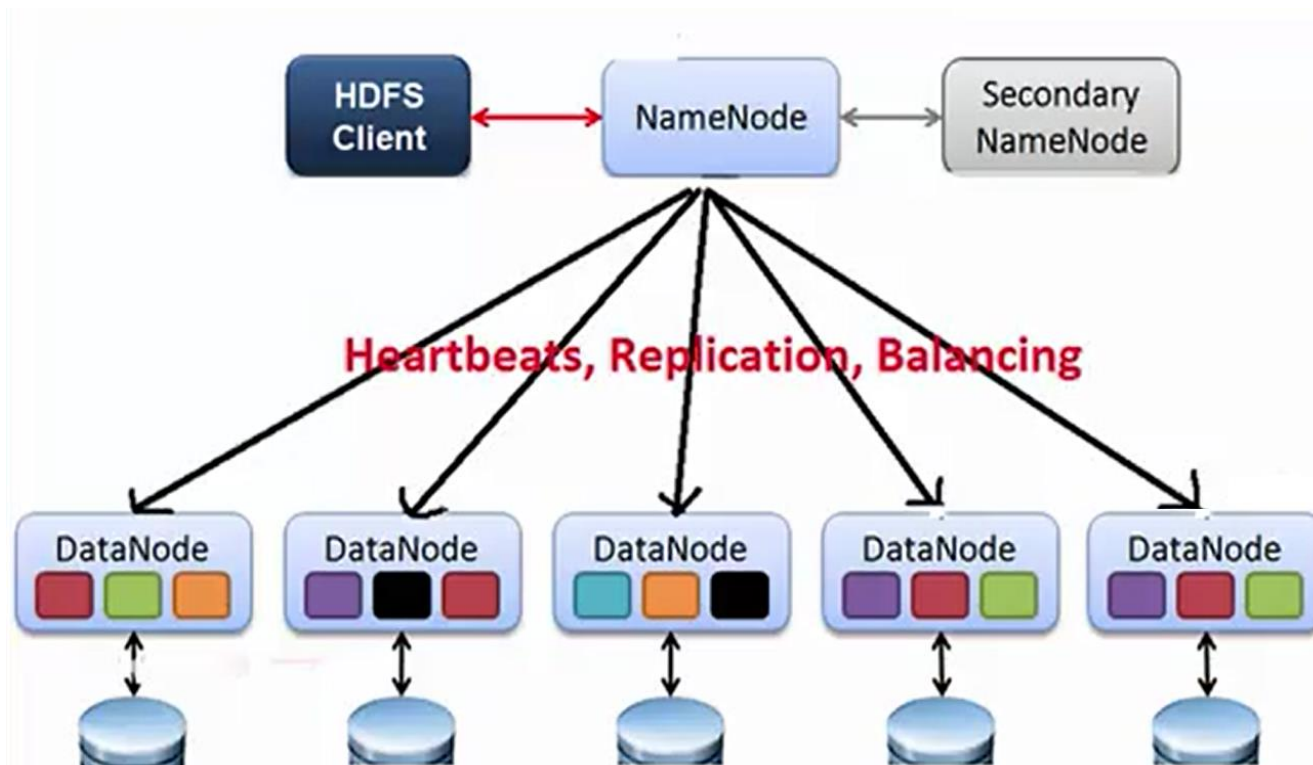
HDFS

- High-Availability :
- با افزودن یک نود به نام **Secondary Name Node** قابلیت دسترسی پذیری بالاتری ارائه شده است.



HDFS

: High-Availability •



HDFS

بلاک:

- کوچک ترین واحد ذخیره سازی داده ها در HDFS هستند

HDFS

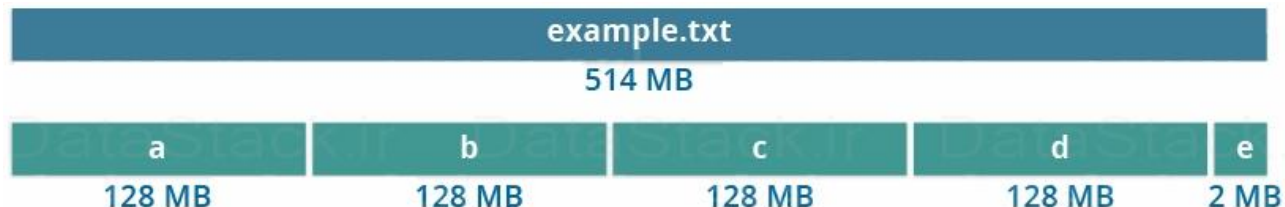
بلاک:

- HDFS هر فایل را بصورت بلاک‌ها ذخیره می‌کند که این بلاک‌ها در میان خوشه آپاچی هدوپ بصورت پراکنده ذخیره می‌شوند.

HDFS

بلاک:

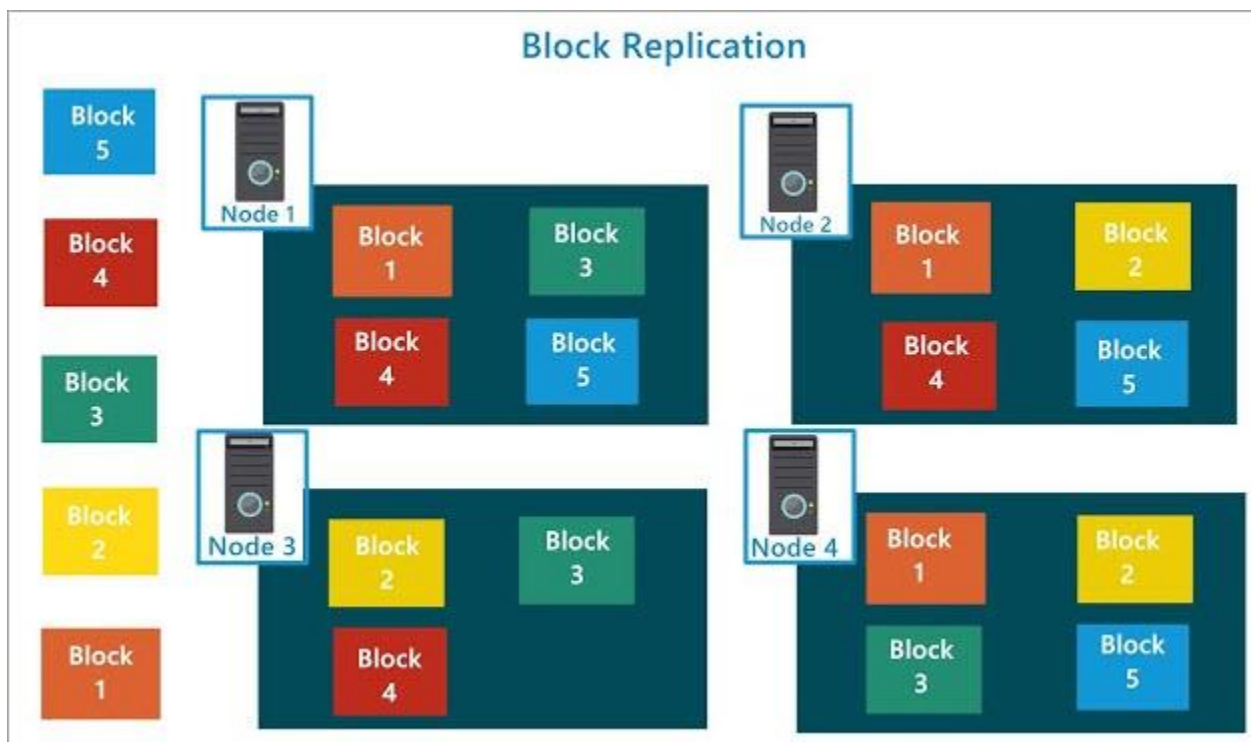
- HDFS هر فایل را بصورت بلاک‌ها ذخیره می‌کند که این بلاک‌ها در میان خوشه آپاچی هدوپ بصورت پراکنده ذخیره می‌شوند.



HDFS

مدیریت Replication:

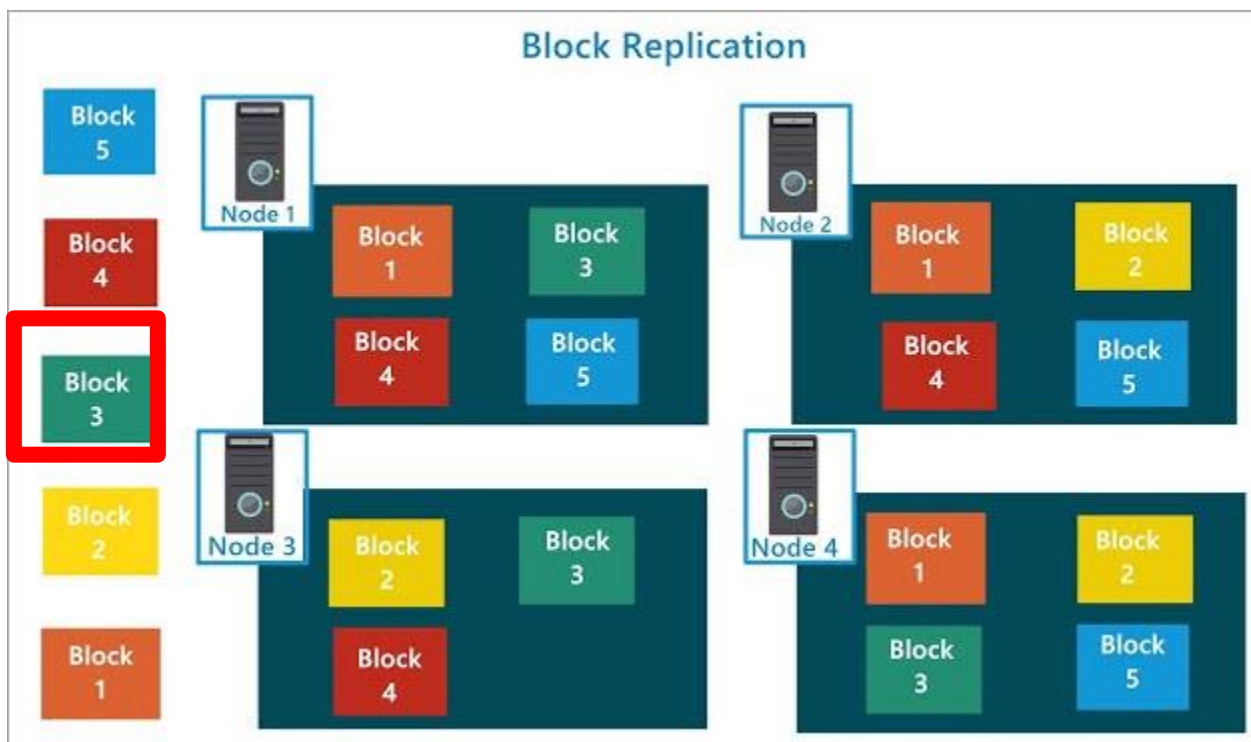
- HDFS جهت فراهم آوردن قابلیت تحمل پذیری در برابر خطا، این بلاک‌ها در **چندین** گره از خوشه کپی می‌شوند.



HDFS

مدیریت Replication:

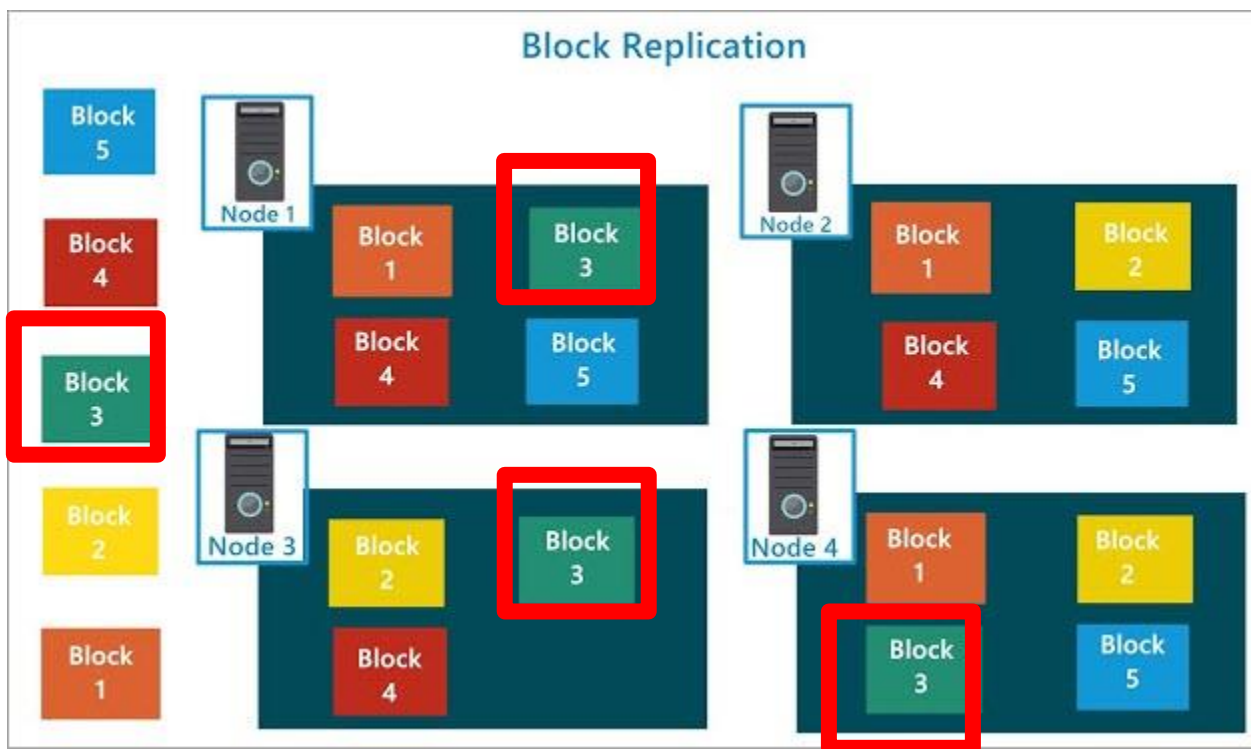
- HDFS جهت فراهم آوردن قابلیت تحمل پذیری در برابر خطا، این بلاک‌ها در چندین گره از خوشه کپی می‌شوند.



HDFS

مدیریت Replication:

- HDFS جهت فراهم آوردن قابلیت تحمل پذیری در برابر خطا، این بلاک‌ها در چندین گره از خوشه کپی می‌شوند.



HDFS

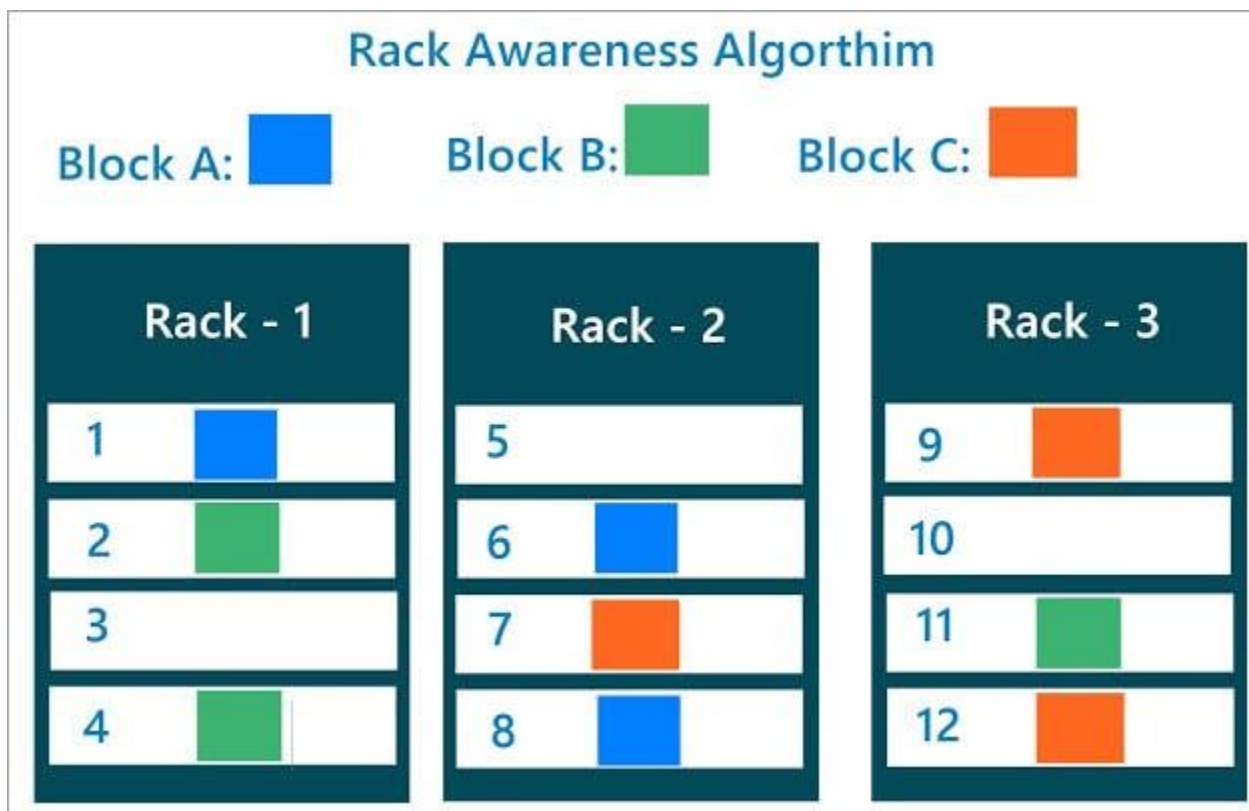
مدیریت Replication:

- ذخیره یک فایل ۱۲۸ مگابایتی در HDFS با استفاده از ضریب پیش فرض، فضایی به میزان ۳۸۴ مگابایت ($۱۲۸ * ۳$ مگابایت) اشغال خواهد شد؛ زیرا بلاک ها سه بار کپی شده و هر کپی در DataNode متفاوتی ساکن می شود..

HDFS

افزایش قابلیت تحمل پذیری خطا:

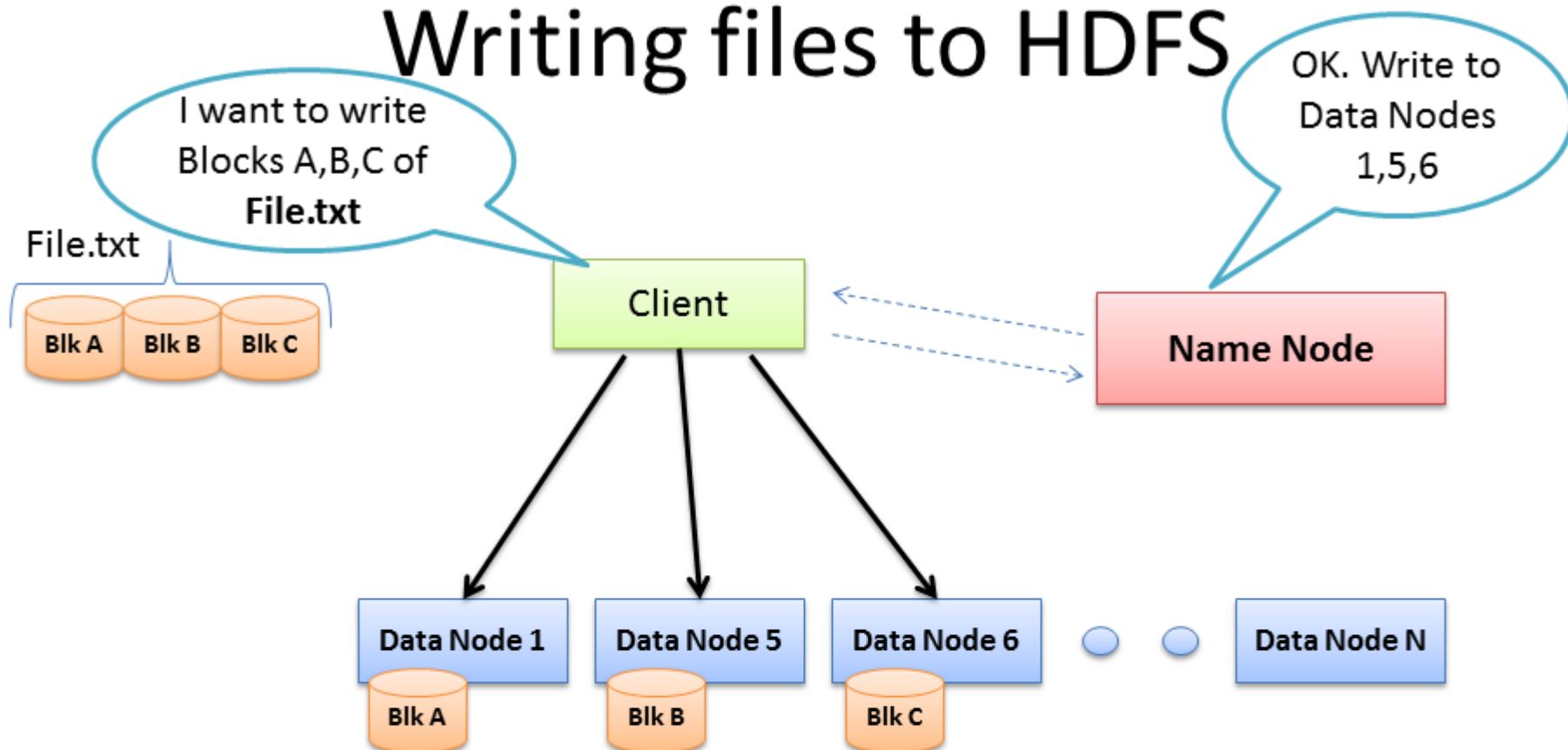
در صورت امکان بیشتر از دو بلاک بروی یک رک یکسان قرار نگیرند



HDFS

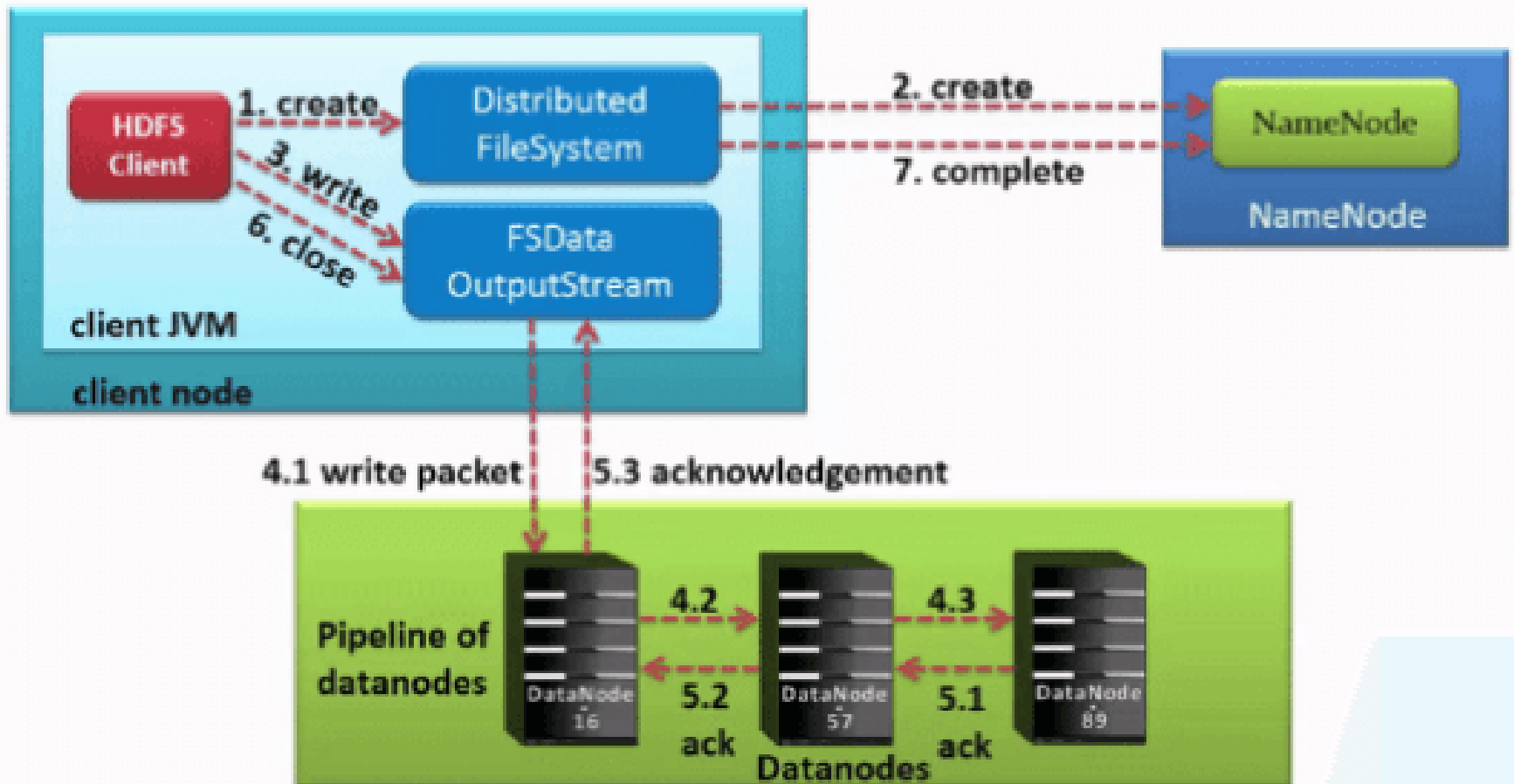
خواندن و نوشتن در HDFS :

Writing files to HDFS



HDFS

نوشتن در HDFS :



HDFS

نوشتن در HDFS :

- ایجاد Pipeline
- ارسال جریان داده ها و تکرار آنها
- خاتمه دادن Pipeline (دریافت Ack صحت کپی)
-

HDFS

نوشتن در HDFS :

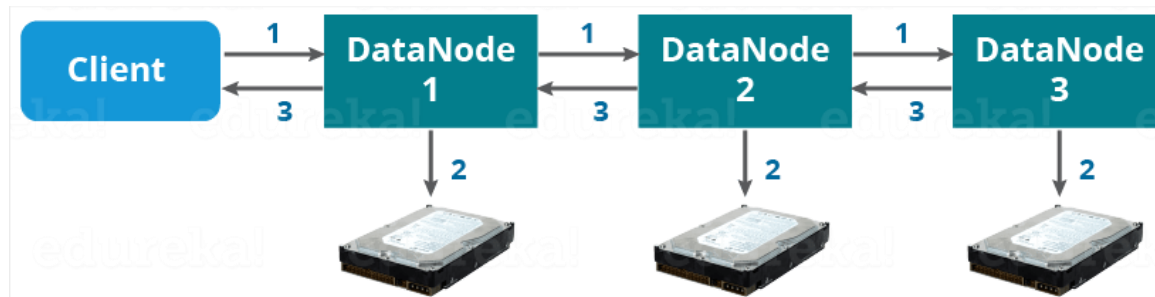
- ایجاد Pipeline
 - قبل از نوشتن بلاک، کلاین چک می کند که Data Node های اعلام شده از فهرست IP ها آماده دریافت باشند.
 - کلاین یک Pipeline برای هر کدام از این بلاک ها ایجاد می کند که این کار توسط ارتباط دادن Data Node ها در فهرست مربوط به همان بلاک انجام می شود.
 - برای مثال برای بلاک A :
-
- Fro Blok A:
 - List A={IP of DataNode1,
IP of DataNode4,
IP of DataNode6}

HDFS

نوشتن در HDFS :

— ایجاد Pipeline

- For Block A, list A = {IP of DataNode 1, IP of DataNode 4, IP of DataNode 6}
- For Block B, set B = {IP of DataNode 3, IP of DataNode 7, IP of DataNode 9}
- Each block will be copied in three different DataNodes to maintain the replication factor consistent throughout the cluster.
- Now the whole data copy process will happen in three stages:



HDFS

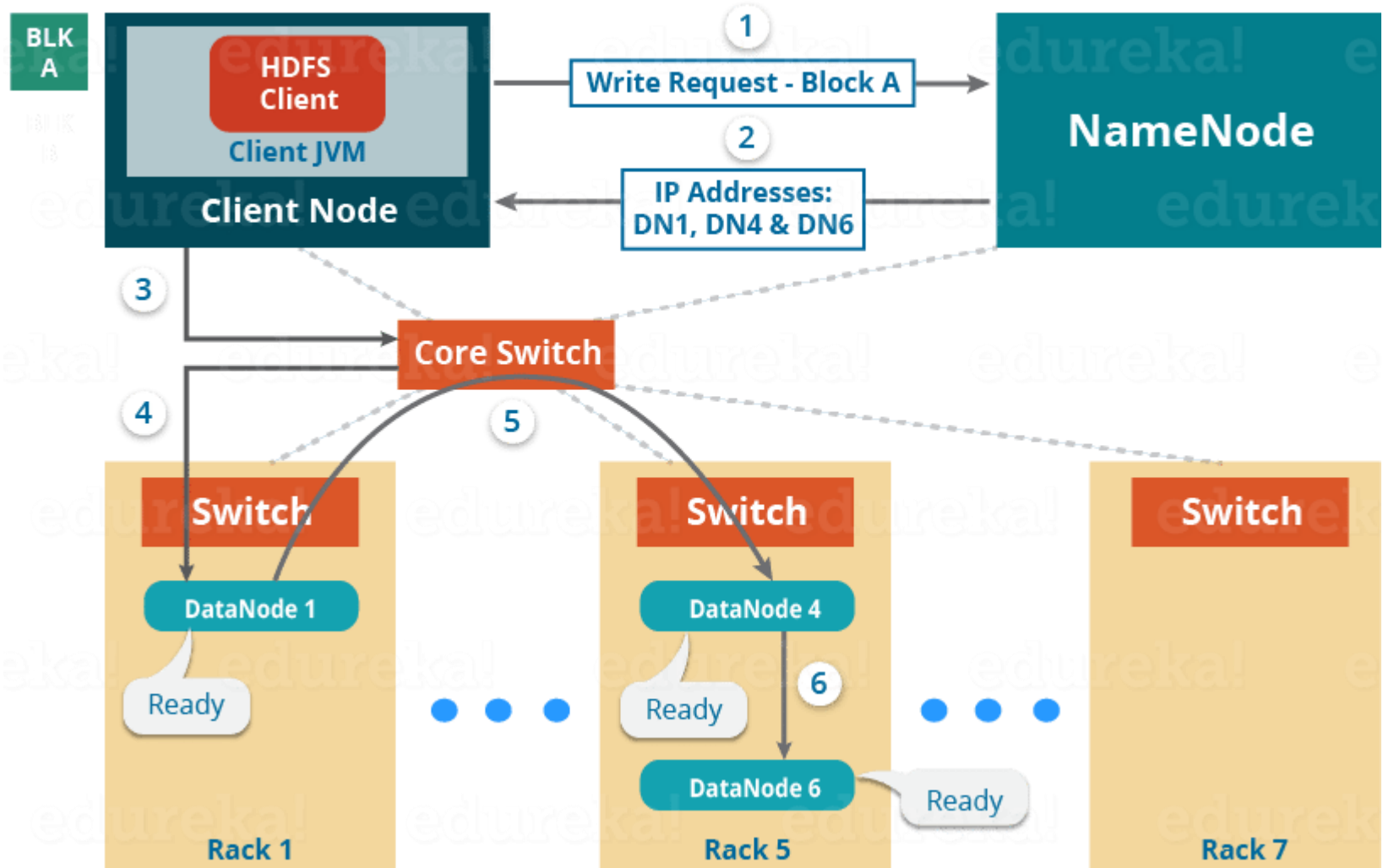
نوشتن در HDFS :

بلاک A

- کلاینت اولین DataNode را که در فهرست IP های موجود برای بلاک A را انتخاب می کند.
- یک ارتباط TCP/IP شکل می دهد.
- کلاینت نود DataNode1 را برای دریافت بلاک آگاه می کند.
- ادرس IP دو DataNode بعدی را هم برای آنها می فرستد

HDFS

Setting up HDFS - Write Pipeline



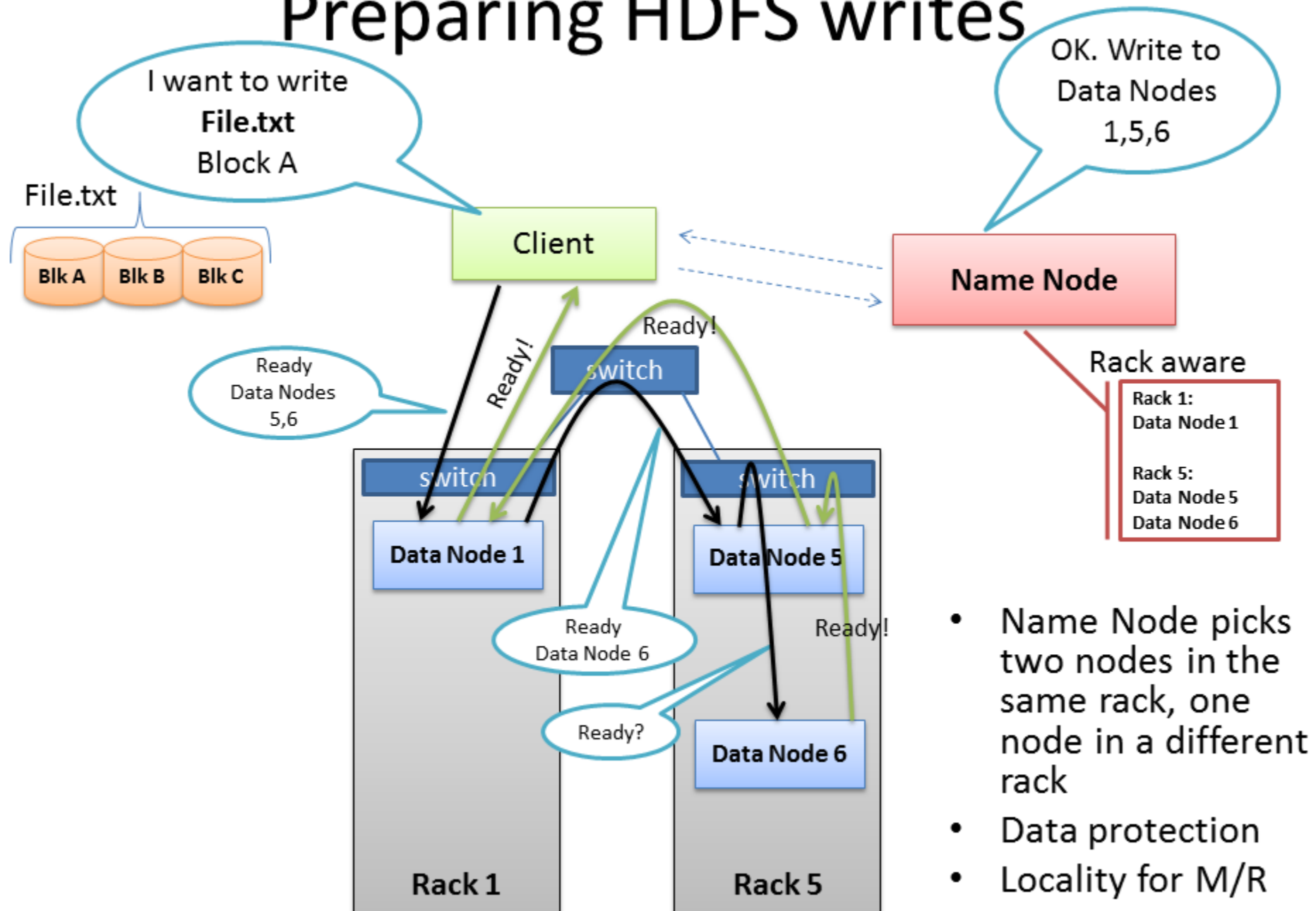
HDFS

نوشتن در HDFS :

بلاک A

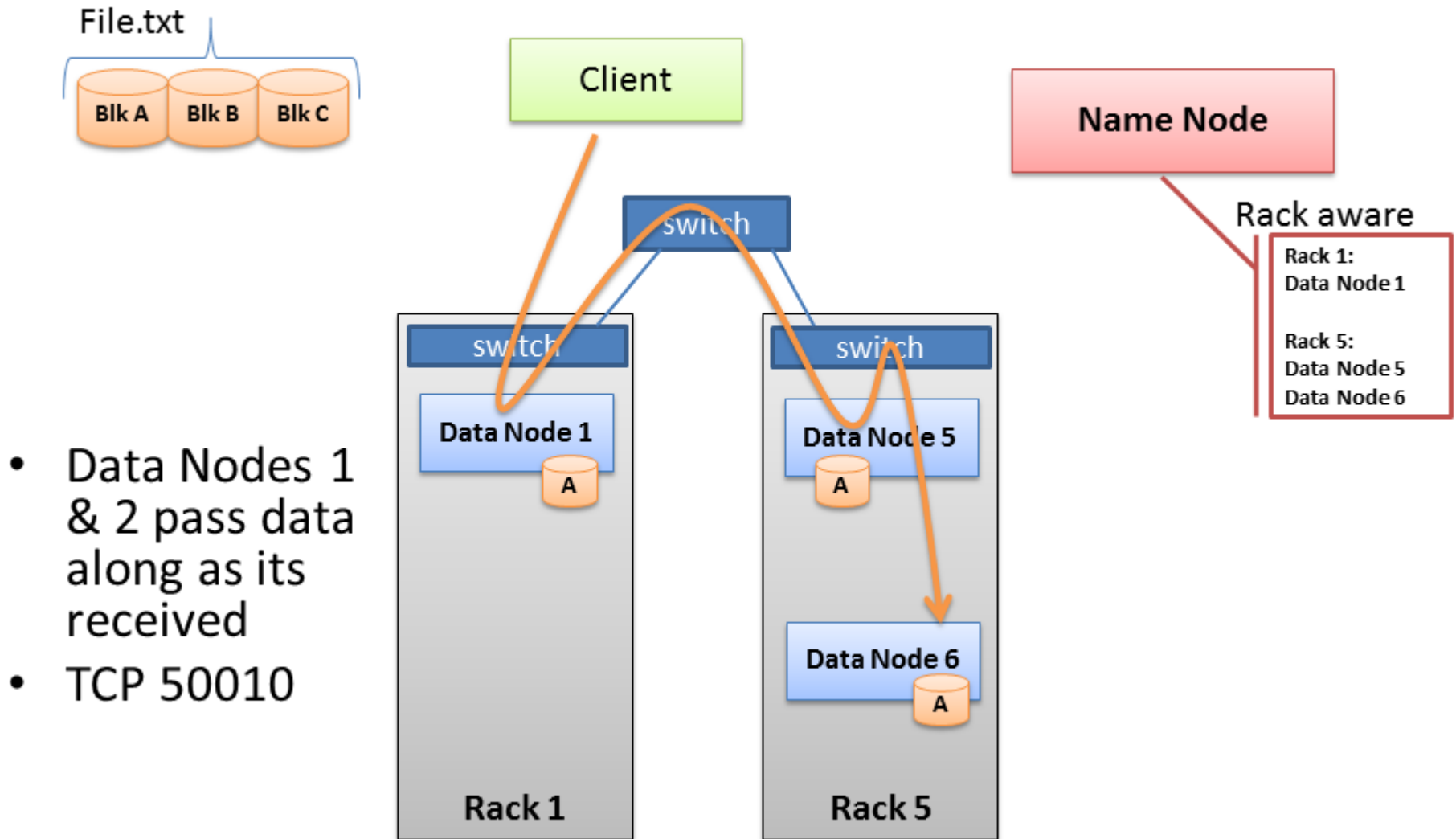
- کلاینت اولین DataNode را که در فهرست IP های موجود برای بلاک A را انتخاب می کند.
- یک ارتباط TCP/IP شکل می دهد.
- کلاینت نود DataNode1 را برای دریافت بلاک آگاه می کند.
- ادرس IP دو DataNode بعدی را هم برای آنها می فرستد
- مرحله بعد، مرحله ارسال Ack با توالی معکوس است یعنی از DataNode 6 تا ۴ و سپس ۱،
- سرانجام DataNode 1 به کلاینت اعلام می کند که تمام DataNode ها آماده بوده و خط لوله بین کلاینت، DataNode 1، ۴ و ۶ ایجاد می شود.
- حال ایجاد خط لوله کامل شده و کلاینت سرانجام کپی جریان داده ها را آغاز می کند.

Preparing HDFS writes



HDFS

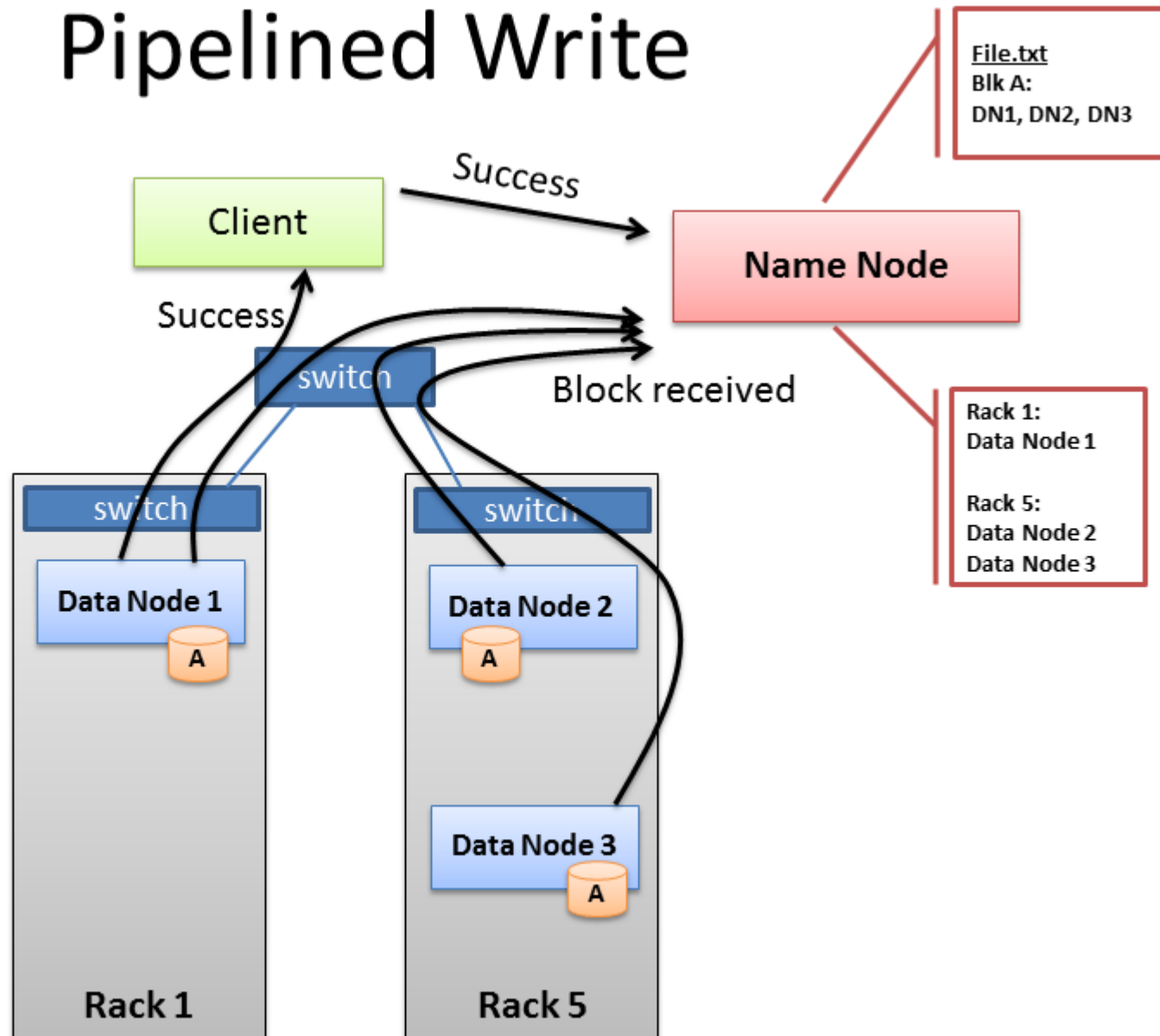
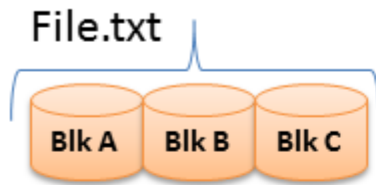
Pipelined Write



- Data Nodes 1 & 2 pass data along as its received
- TCP 50010

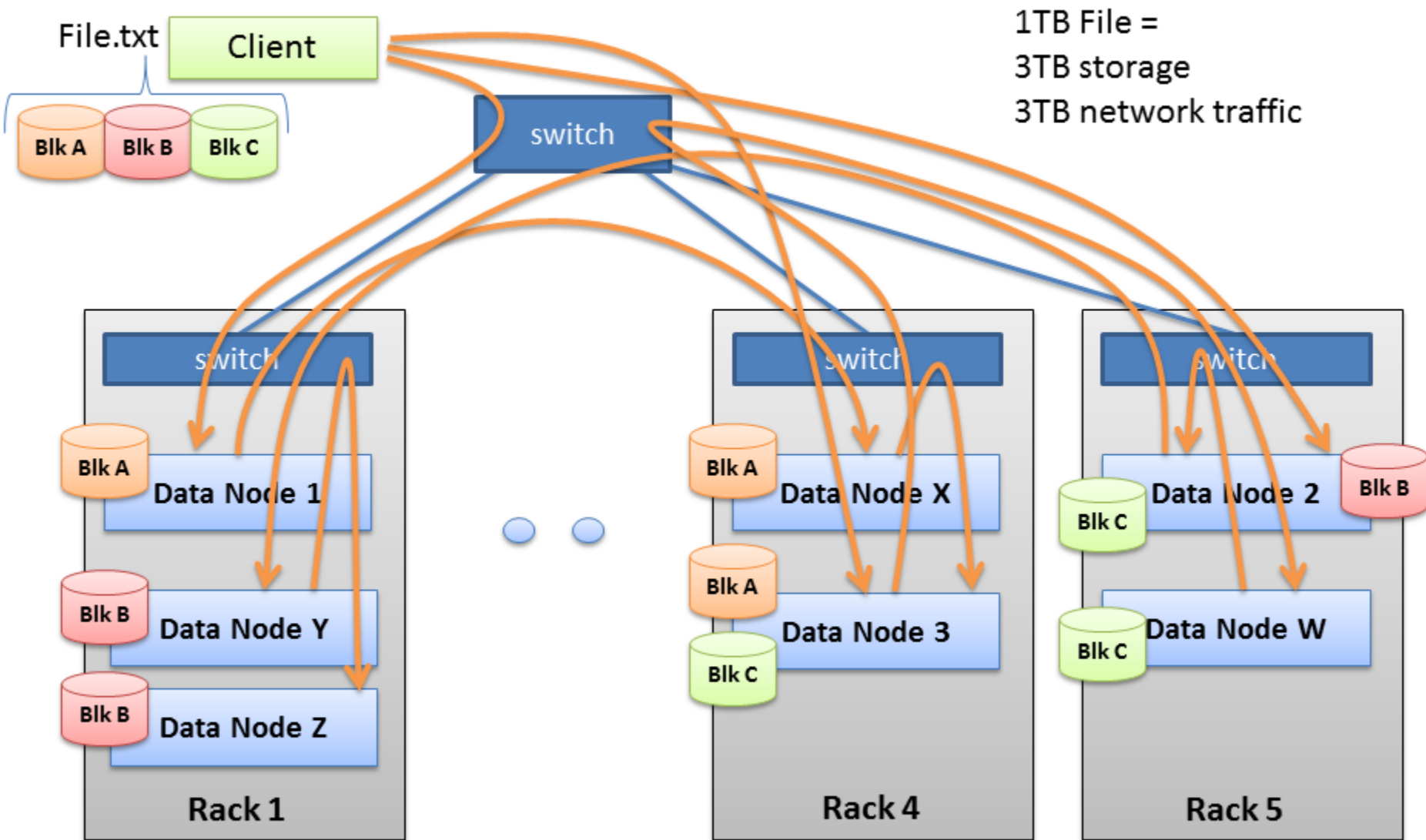
HDFS

Pipelined Write



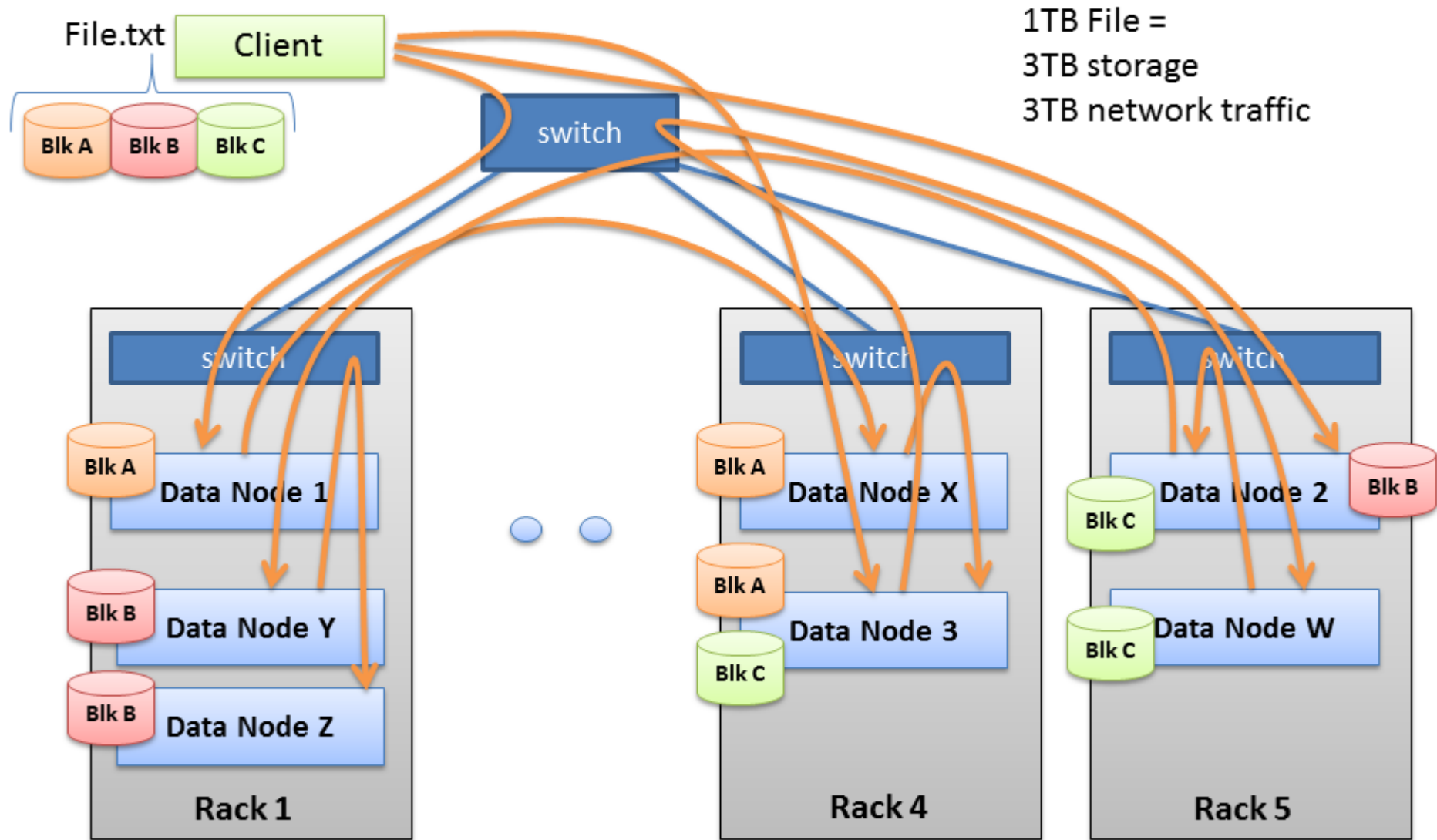
HDFS

Multi-block Replication Pipeline



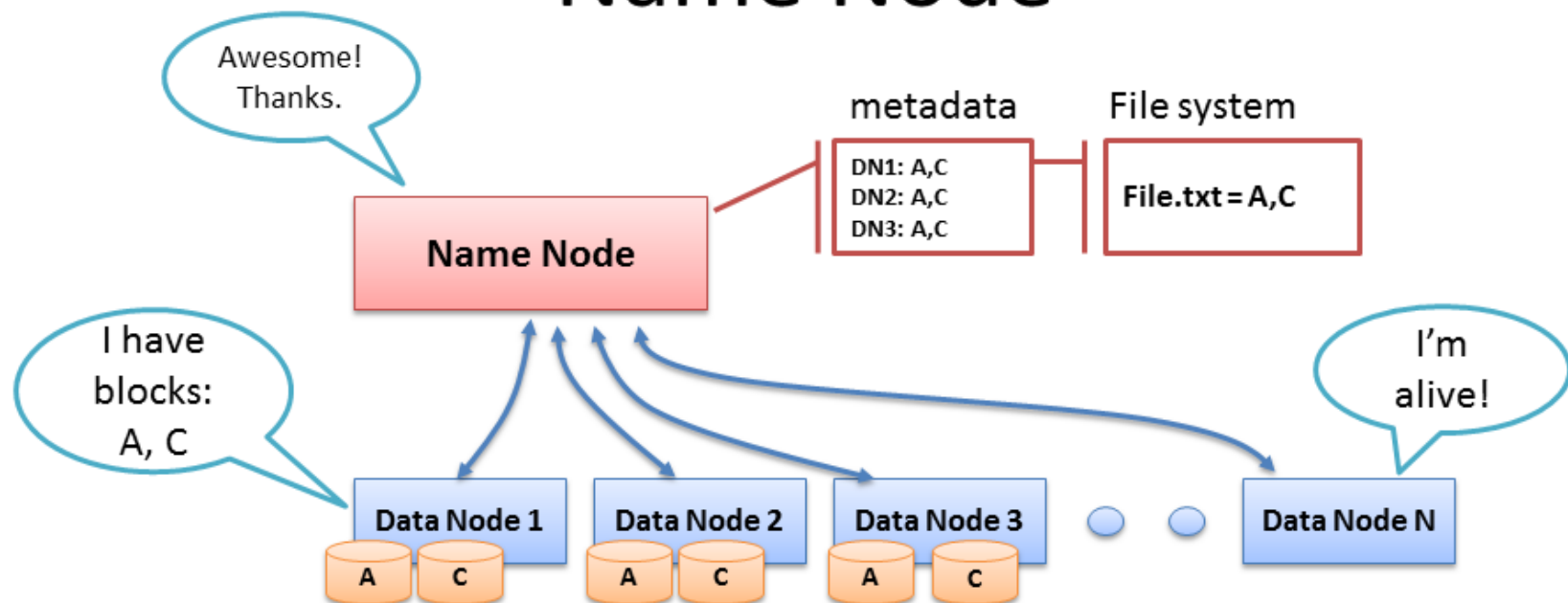
HDFS

Multi-block Replication Pipeline



HDFS

Name Node

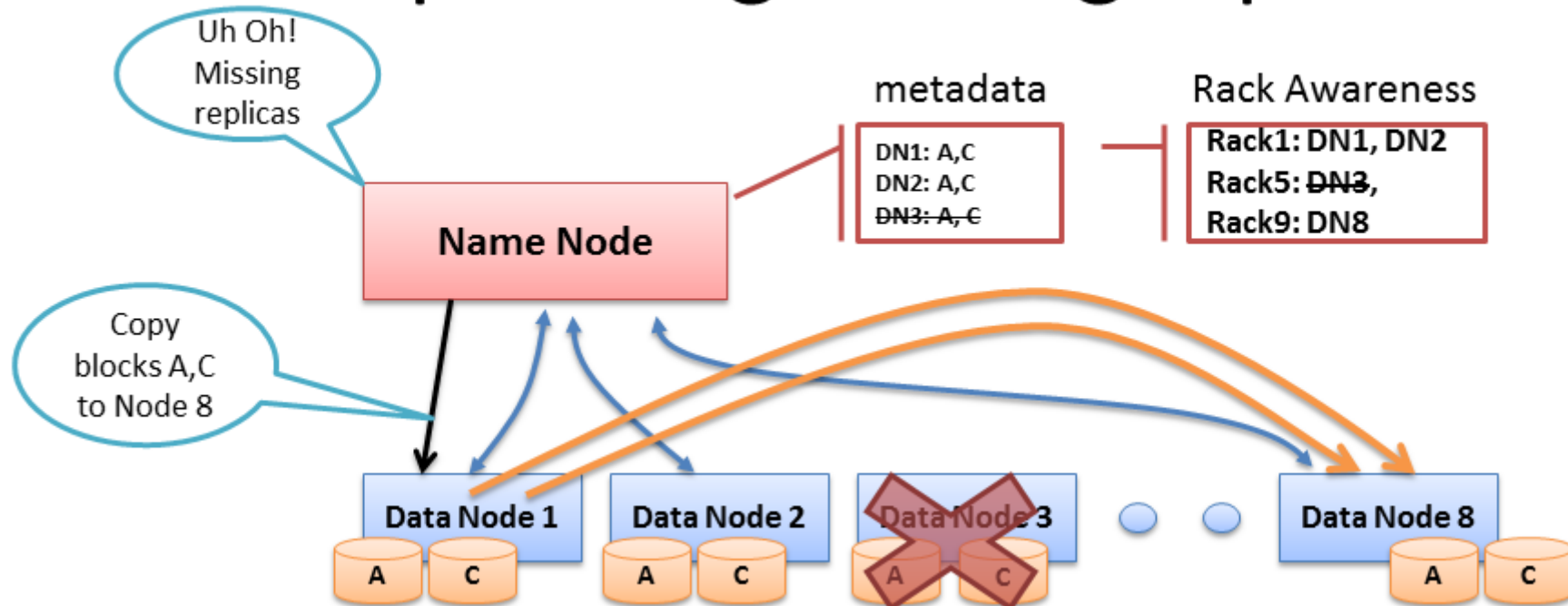


- Data Node sends Heartbeats
- Every 10th heartbeat is a Block report
- Name Node builds metadata from Block reports
- TCP – every 3 seconds
- If Name Node is down, HDFS is down

HDFS

جایگزینی کپی های از دست رفته در HDFS :

Re-replicating missing replicas



HDFS

خواندن از HDFS :

Client reading files from HDFS

