Paper Title： The Hadoop Distributed File System

Student Name：康智詠

Department/Year：資工四甲

Student ID：406261523

Date: 2020/11/25

Q1. What is the problem the authors are trying to solve?

介紹Hadoop Distributed File System(HDFS)的設計理念以及每個結構的功能和運作方式

Q2. What other approaches or solutions existed at the time that this work was done?

PVFS,Lustre,GFS

Q3. What was wrong with the other approaches or solutions?

並沒有介紹

Q4. What is the authors' approach or solution?

**NameNode**

HDFS的命名空間有分層制度，文件和目錄即是NameNode的代表，會記錄許多內容。檔案的內容會被切割成許多的block並被複製於許多的DataNode。NameNode會維持命名空間樹以及檔案和DataNodes之間的映射關係

**DataNodes**

每個block在DataNode的複製品會有兩個檔案。第一個為資料本身，第二個為block的metedata，包含block資料的checksum和block的generation stamp

**HDFS Client**

與常見的檔案系統一樣，提供讀、寫、刪除檔案以及創造、刪除目錄的指令，藉由在命名空間中的路徑來進行索引。

而與常見的檔案系統不同之處在於HDFS提供API來找尋檔案block的位置

**Image and Journal**

Namespace image是檔案系統metadata用來描述應用程式資料的組織。image持續不斷紀錄而被寫入disk稱為checkpoint。Journal是一個預寫提交日誌，所有檔案系統的更動將會被記錄在此當中

**CheckpointNode**

CheckpointNode定期結合現存的checkpoint以及journal來產生新的checkpoint以及journal

**BackupNode**

BackupNode有能力定期創造新的checkpoint，並維持檔案系統命名空間的記憶體中以及最新的image

**Upgrades, File System Snapshots**

藉由創造snapshot來減少在更新時存放在系統的資料損壞的風險，進而讓administrator持續儲存最近檔案系統的狀況

Q5. Why is it better than the other approaches or solutions?

檔案的內容複製到多個DataNode，使得可靠性很高以及有更多的機會使需要的資料就在計算地方的附近

Q6. How does it perform?

**File Read and Write**

1. 藉由創造新的檔案以及寫入資料來為應用程式添加資料到HDFS。HDFS允許單個寫入者或多個寫入者的模式
2. 當client打開檔案寫入時必須取得該檔案的許可證，正在寫入的client 將會藉由傳送heartbeat到NameNode來定期更新許可證。當檔案關閉時，許可證將會被撤銷。
3. 許可證會有軟限制以及硬限制。當超過軟限制的時間，寫入者沒有關閉檔案或沒有更新許可證，那麼許可證將可以被其他client佔有。如果超過硬限制的時間，檔案將會自動被關閉。
4. 在DataNode的複製品可能會損毀，HDFS為每個檔案的資料block產生並保存checksum，可以用此來檢驗是否有錯誤產生

**Block Placement**

1. NameNode可以解決每個DataNode的rack位置。當DataNode向NameNode註冊時，NameNode會運行以配置的腳本來決定哪個rack是屬於它的
2. 當新的block被創造，HDFS會將第一個複製品放在寫入者的節點，第二及第三個複製品將會放在不同的rack。
3. 在所以有的目標節點都被選定後，節點會被組成pipeline來將複製品盡量靠近第一個複製品

**Replication management**

1. NameNode會努力確保每個blcok有想要數量的複製品。當NameNode收到來自DataNode的回報，block的複製品有超過過低於想要的數量，NameNode會進行檢測。如果超過想要的數量，會傾向刪除擁有較少儲存空間的DataNode。
2. 如果block低於想要的數量，會推入一個優先序列，低於需求比例愈少的將會擁有較高的權重。背景執行序將會定期對優先序列的前頭進行掃描來決定要將新的複製品放於何處

**Balancer**

Balancer是一個平衡disk空間的工具，藉由群集的管理員部屬為應用程式。會將複製品從高使用率的DataNode搬移到低使用率的DataNode。並且會藉由最小化內部rack資料的複製來最佳化

**Block Scanner**

每個DataNode會運行block掃描器來定期掃描block複製品並驗證儲存的checksum是否有符合block資料，每個block驗證時間將會儲存在人類可以讀取的日誌文件。無論client或block的掃描器何時偵測到腐敗的block將會提醒NameNode，NameNode將會把腐敗的block進行標記，但並不會馬上將複製品放入排程進行刪除。而是開始將block複製好，達到預期的數量才會進行搬移

**Decommissioing**

群集的管理員藉由列出那些節點被同意可以註冊以及不被同的可以註冊的主機地址才進行加入群集的篩選。管理員可以命令系統重新計算包含以及排除的清單，原本有被包含而轉為不被包含的將會被標記為退役，將不會成為複製品放置的目標，但會繼續服務讀取請求。一旦NameNode偵測到所有的block在退役的DataNode被複製，節點將會進入退役狀態，才可以被安全刪除。

**Inter-Cluster Data Copy**

HDFS提供DistCp，給大型inter/intra-cluster平行複製。進行MapReduce的工作，每個map任務會複製來源資料的一部份到目的地檔案系統，MapReducec的架構將會自動處理平行任務排程、錯誤偵測及復原

Q7. Why is this work important?

HDFS可以可靠儲存大資料集合，並且可以在高頻寬傳遞給使用者。在大群集中，成千個伺服器可以直接在主機附加儲存空間並且運行使用者的任務。藉由分散式儲存以及運算可以有效利用資源。

Q8. Can any improvement be done?

目前沒想到