Paper Title： Bigtable: A Distributed Storage System for Structured Data

Student Name：康智詠

Department/Year：資工四甲

Student ID：406261523

Date: 2020/11/4

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

主要在介紹由Bigtable所提供的資料模型，可以給與客戶端動態控制資料以及格式，以及敘述Bigtable的設計和實作方法

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

1. Boxwood project某些部分與Chubby、GFS、Bigtable相同，例如分散式協議、locking、分散式chunk storage、分散式B-tree storage
2. Oracle’s Real Application Cluster database 共享磁碟來儲存資料以及分散式lock管理者
3. IBM’DB2 Parallel Edition 的架構與Bigtable相似，是依據shared-nothing

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

1. Boxwood project的目標是提供基礎設施給高階的服務
2. 藉由分散式B-tree或分散式hash table所實作出的key-value模型有許多限制

Q4. What is the authors' approach or solution?

**Data Model**

Map由row key,column key, timestamp來進行索引

**Rows**

Bigtable利用row來進行字典序排序資料。Table的row range是動態分割的。每個row range是進行負載平衡以及分散的基本單位，也稱作tablet

**Column Families**

Column key打包成一個集合稱為column family，是訪問控制的基本單位。通常在同一個column family是同樣的型態。Column key必須在資料儲存在此family的任何column key前被創造。在family被創造完成後，任何的column key都可以被使用。此外，Column key以syntax:family:qualifier的方式進行命名

**Timestamp**

利用Timestamp使Bigtable可以對相同資料保存多個版本，並且被索引。不同版本的資料會將timestamp以遞減的方式進行存取，這樣最近的版本就可以先被讀取。為了減少資料的版本，提供兩個per-column-family設定來自動告訴Bigtable將版本進行garbage-collect，使client可以要求保存前n個版本的檔案，或者僅保存最新的版本

**API**

1. Bigtable API提供函式來創造或刪除table和column family，以及修改cluster,table,column family,metadata
2. Client應用程式可以在Bigtable中寫入、刪除值，或者查詢單獨row的值，也可以在table中遍歷資料的子集合
3. Bigtable可以讓使用者用複雜的方式對資料進行運算。幫助Single-row transactions、使用integer counter、幫助運行在server中address space的client-supplied script

**Building Blocks**

1. Bigtable使用GFS來儲存log和檔案資料。Bigtable cluster通常在機器的共享池當中進行操作，並且程序會共享機器來運行不同的應用程式。
2. Bigtable的資料使用Google SSTable的檔案格式來儲存。SSTable提供持久且一成不變將key映射到value。可以特定的key來查詢特定的值或者在特定範圍內遍歷所有的key/value
3. Bigtable依賴Chubby，來完成許多任務。Chubby提供高度可用性且持久的分散式lock服務。如果Chubby停止服務，那麼Bigtable也將無法運行

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

**Row**

可以有效率讀取小範圍的row，並且通常僅需要與少數量的機器進行交流

**Column Families**

可以管理許多不同種類的應用程式

Q6. How does it perform?

**Tablet Location**

第一個等級的檔案儲存在Chubby的root tablet，Chubby當中有root tablet的位置。root tablet當中有所有tablet在METADATA table位置。每個METADATA table儲存一個集合的user tablet位置

**Tablet Assignment**

每個tablet一次會被指派到一個tablet server。Master會追蹤存活tablet server的進度以及最近被指派的tablet，還有沒被指派的tablet。當tablet沒被指派，而tablet server可以負荷的時候，master會藉由傳送tablet載入請求到tablet server來指派tablet

**Tablet Serving**

更新會被上傳到commit log，commit log用來儲存redo record。在更新的時候，最近上傳的資料會被儲存在sorted buffer的記憶體，被稱為memtable；較舊的更新資料會被淳在SSTable。

為了復原tablet，tablet server會讀取METADATA table當中的metadata。Metadata包含一個鏈結的SSTable和一個集合的redo point，SSTable會指向任何包含tablet資料的commit log。Server會將所需的一塊SSTable讀進記憶體並藉由執行所有已經提交的更新來重建memtable

當寫入操作傳送到tablet server, server檢查是否完好無缺，並且傳送者會被授權更改的能力。藉由來自Chubby file讀取被同意寫作者的清單來決定是否進行授權。

當讀取操作抵達tablet server，通樣地進行檢查是否完好無缺以及是否有適當的授權。如果是被許可的讀取操作將會在連續的SSTable以及memtable上運行

**Compactions**

當寫入操作運行時，memtable的大小將會增加。如果memtable的大小達到臨界點，那麼memtable將會被凍結，之後會創造一個新的memtable。而被凍結的memtable將會被轉換成SSTable並且被寫入GFS

Q7. Why is this work important?

可以滿足許多專案儲存資料的需求，可以可靠擴展到petabytes的資料以及上千個機器。達到許多目標，廣泛應用、可擴展、高性能、高可用性。提供多個不同的介面並提供client簡單的資料模型來輔助動態控制資料的版面及格式，且允許client選擇模式控制本地資料

Q8. Can any improvement be done?

目前沒想到