

奠定大數據的基石

# NoSQL 資料庫技術

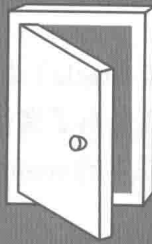
皮雄軍 編著  
蔡桂宏 博士 審閱

騰訊前資深後台工程師力作，大數據時代必讀佳作！

本書範例，  
可至佳魁資訊官網下載



佳魁資訊



Knowledge is power

「資料是 21 世紀最有價值的資產，它比黃金和石油更有價值」。隨著大數據時代的來臨，傳統的關聯式資料庫在可擴充性、資料模型和可用性方面遇到了難以克服的障礙。此時各種 NoSQL 系統出現了。它們的特點各不相同，分別應用於不同的場景並迅速獲得了極大的成功。作為一名從事後台開發多年的工作者，我對每一種新技術的出現與應用都充滿了渴望與期待，其中 NoSQL 解決了我實際工作中遇到的許多問題。NoSQL 具有下面幾方面的優點：

## 1. 靈活的可擴充性

多年以來，資料庫管理員們都是透過「垂直擴充」的方式（當資料庫的負載增加的時候，購買更大型的伺服器來承載增加的負載）來進行擴充的，而非透過「水平擴充」的方式（當資料庫負載增加的時候，在多台主機上分配增加的負載）來進行擴充。但是，隨著請求量和可用性需求的增加，資料庫也正在移轉到雲端或虛擬化環境中，「水平擴充」的經濟優勢變得更加明顯了，對各大企業來說，這種「誘惑」是無法抗拒的。

要對 RDBMS（關聯式資料庫，例如 Oracle）做「水平擴充」，並不是很容易。但是各種新類型的 NoSQL 資料庫主要是為了進行透明的擴充，來利用新節點而設計的，而且，它們通常都是為了低成本的 commodity hardware 而設計的。

## 2. 輕鬆應對大數據

在過去的十年裡，正如請求量發生了天翻地覆的增長一樣，需要儲存的資料量也發生了急劇的膨脹。為了滿足資料量增長的需要，RDBMS 的容量也在日益增加，但是，對一些企業來說，隨著請求量的增加，單一資料庫能夠管理的資料量也變得越來越讓人無法忍受了。現在，大量的「大數據」可以透過 NoSQL 系統（如 MongoDB）來處理，它們能夠處理的資料量遠遠超出了最大型的 RDBMS 所能處理的極限。

## 3. 維護簡單

在過去的幾年裡，雖然一些 RDBMS 供應商們聲稱在可管理性方面做出了很多的改進，但是高階的 RDBMS 系統維護起來十分昂貴，而且還需要訓練有素的 DBA 們的協助。DBA 們需要親自參與高階的 RDBMS 系統的設計、安裝和最佳化。

NoSQL 資料庫從一開始就是為了降低管理方面的要求而設計的：從理論上來說，自動修復，資料分配和簡單的資料模型的確可以讓管理和最佳化方面的要求降低很多。

## 4. 經濟

NoSQL 資料庫通常使用廉價的 Commodity Servers 叢集來管理膨脹的資料和請求量，而 RDBMS 通常需要依靠昂貴的專有伺服器 and 儲存系統來做到這一點。使用 NoSQL，每 GB 的成本或每秒處理的請求的成本都比使用 RDBMS 的成本少很多，這可以讓企業花費更低的成本儲存和處理更多的資料。

## 5. 靈活的資料模型

對大型的生產性 RDBMS 來說，變更管理是一件很令人頭痛的事情。即使只對一個 RDBMS 的資料模型做出很小的改動，也必須要十分小心的管理，也許還需要停機或降低服務水準。NoSQL 資料庫在資料模型約束方面是更加寬鬆的，甚至可以說並不存在資料模型的約束。NoSQL 的 Key/Value 資料庫和文件類型資料庫可以讓應用程式在一個資料元素裡儲存任何結構的資料。即使是規定更加嚴格的以「大表」為基礎的 NoSQL 資料庫（如 HBase）通常也允許建立新列，這並不會造成什麼麻煩。

應用程式變更和資料庫綱要的變更並不需要作為一個複雜的變更單元來管理。從理論上來說，這可以讓應用程式反覆運算的更快，但是，很明顯，如果應用程式無法維護資料的完整性，那麼這也會帶來一些不良的副作用。

## ● 本書的誕生

目前還沒有一本全面而深入介紹 NoSQL 系統的圖書。不是只有理論內容，就是千篇一律把某個 NoSQL 系統的操作一一羅列。為了讓許多的 NoSQL 學習者能夠更加全面和深入地學習 NoSQL 技術，筆者撰寫了本書。從系統的角度出發，既深入說明 NoSQL 的產生原因與理論基礎，又對三種典型 NoSQL 系統引用了豐富的實戰，讓讀者可以全面而深入地了解各種 NoSQL，知道各個 NoSQL 和 MySQL 系統的不同，根據自己的情況進行正確的技術選型。

## ● 本書特色

### 1. 內容全面、新穎

本書內容全面而新穎，既對 NoSQL 系統的理論進行深入淺出的分析，又深入說明了列式的、Key/Value 的和文件類型這三種 NoSQL 系統，最後還說明了 MySQL 的效能最佳化。

## 2. 說明由淺入深，循序漸進

如果你從來沒有用過 NoSQL 系統，本書正適合你。並且由淺入深將各種 NoSQL 系統聯繫起來，並分析各自的相同點和不同點，讀完之後你一定會對 NoSQL 系統有更多的認識。

## 3. 理論與實作並重

對 NoSQL 系統的產生原因和理論基礎做了廣泛而深入的分析，讓讀者知其然，更知其所以然。書中儘量消除初學者學習 NoSQL 系統時容易遇到的障礙，變抽象為實際，變複雜為簡單。而且重點對 Hadoop/HBase、MongoDB 和 Redis 這三種 NoSQL 系統都列出了實際實作。

## 4. 圖文並茂，容易了解

針對 NoSQL 系統中的一些架構和較難了解的概念，提供了大量的插圖，並結合實際文字來說明，非常直觀，更有利於讀者的學習與了解。

## 5. 語言通俗容易

不使用讓人難以了解的語言分析問題，而是採用通俗易懂的敘述說明 NoSQL 系統的相關知識，讓讀者能夠真正了解 NoSQL 技術，繼而在開發中使用這些技術。

## ● 本書內容

### 第 1 篇 NoSQL 的興起與理論基礎（第 1~3 章）

介紹了大數據時代 NoSQL 的興起原因及 NoSQL 的理論基礎，包含 NoSQL 與大數據簡介，NoSQL 系統的分類和特點，各種資料一致性模型的定義、原理、實現、舉例及適用場景，NoSQL 系統水平擴充的方法，主要是複製和分區技術的困難、在實際系統中的運用及和資料一致性的關係，最後簡單介紹了其他知識，例如五分鐘法則等。這些內容都是學習後續章節所必須要掌握的基礎知識，後續章節中的實際 NoSQL 系統如 Redis 可以看成是這些理論知識的運用。

### 第 2 篇 列式 NoSQL 系統（第 4、5 章）

主要介紹了列式 NoSQL 系統。首先以 Google 的 BigTable 為例介紹了列式 NoSQL 系統的特點、原理和應用。然後實戰演練開放原始碼的 HBase 系統。由於 HBase 與 Hadoop、Zookeeper 等具有密切的關係，因此對其一併進行了介紹。

### 第 3 篇 Key/Value NoSQL 系統（第 6~8 章）

主要介紹了 Key/Value NoSQL 系統。首先以 Amazon 的 Dynamo 為例介紹了 Key/Value 系統的特點、原理和應用場景，並詳細描述了這個系統是如何運用本書第 1 篇中的理論知識而建置的。然後介紹另一個來自 Google 的 LevelDB 系統，最後實戰演練被廣泛使用的 Redis 系統。

### 第 4 篇 文件型 NoSQL 系統（第 9、10 章）

主要介紹文件型 NoSQL 系統。首先以 CouchDB 為例介紹文件型 NoSQL 的特點、原理和應用場景，然後實戰演練了被廣泛使用的 MongoDB 系統。

### 第 5 篇 MySQL 基礎與效能最佳化（第 11、12 章）

主要介紹了目前在網際網路公司被廣泛使用的 MySQL 關聯式資料庫。雖然在大數據時代 NoSQL 將佔據資料處理技術的主流，但是傳統的 MySQL 在一些應用場景仍具有優勢。所以本篇先介紹了一些 MySQL 的基礎知識，然後詳細描述了其進階特性，最後介紹 MySQL 的效能最佳化、複製技術、垂直擴充、水平擴充和綜合應用。

#### ● 本書適合讀者群

- NoSQL 初學者；
- 網際網路公司高平行處理系統的後台開發人員；
- 大數據、NoSQL 開發同好；
- 剛就職的中級程式設計師；
- 大專院校師生；
- 相關教育訓練班的學員。

#### ● 本書原始程式碼取得方式

本書有關的原始程式碼，請讀者自行到 [www.topteam.cc](http://www.topteam.cc) 上資源下載區下載。

#### ● 本書作者

雖然筆者花費了大量精力寫作，並力圖將疏漏減少到最少，但仍恐百密一疏。如果您在閱讀本書的過程中發現有任何疏漏，或對本書的說明有任何疑問，請發送電子郵件到 [bookservice2008@163.com](mailto:bookservice2008@163.com)。

編者

1  
Part

NoSQL 的興起與理論基礎

Chapter 01 NoSQL 與大數據簡介

1.1	引子 —— NoSQL 在中國大陸使用的案例 .....	1-2
1.1.1	新浪微博.....	1-2
1.1.2	淘寶資料平台.....	1-3
1.1.3	視覺中國網站.....	1-4
1.1.4	優酷營運資料分析.....	1-5
1.1.5	飛信空間.....	1-6
1.1.6	豆瓣社區.....	1-7
1.2	大數據 .....	1-9
1.2.1	大數據的度量單位.....	1-9
1.2.2	大數據的特點.....	1-10
1.3	大數據相關技術 .....	1-12
1.3.1	大數據擷取技術 .....	1-12
1.3.2	大數據前置處理技術 .....	1-12
1.3.3	大數據儲存及管理技術 .....	1-13
1.3.4	大數據分析及採擷技術 .....	1-13
1.3.5	大數據展現與應用技術 .....	1-14
1.4	NoSQL 簡介.....	1-15
1.4.1	什麼是 NoSQL .....	1-15
1.4.2	關聯式資料庫簡史.....	1-15
1.4.3	資料庫分類 .....	1-15
1.4.4	關聯式資料庫的優勢 .....	1-17
1.4.5	不擅長的處理.....	1-18
1.4.6	NoSQL 資料庫 .....	1-21
1.5	NoSQL 資料庫的類型 .....	1-23
1.5.1	鍵值 (Key/Value) 儲存 .....	1-23
1.5.2	針對文件的資料庫.....	1-25
1.5.3	針對列的資料庫 .....	1-25

1.6	如何使用和學習 NoSQL 資料庫 .....	1-26
1.6.1	始終只是一種選擇.....	1-26
1.6.2	在何種程度上信賴它 .....	1-27
1.7	雲端資料管理 .....	1-27

## Chapter 02 NoSQL 的資料一致性

2.1	傳統關聯式資料庫中的 ACID .....	2-2
2.1.1	最小性 .....	2-2
2.1.2	一致性 .....	2-2
2.1.3	隔離性 .....	2-3
2.1.4	持久性 .....	2-3
2.1.5	舉例.....	2-3
2.2	CAP 理論 .....	2-4
2.2.1	NoSQL 系統是分散式系統 .....	2-4
2.2.2	CAP 理論說明 .....	2-5
2.3	AP 的實例 —— DNS 系統.....	2-7
2.3.1	DNS 系統.....	2-7
2.3.2	DNS 域名解析過程 .....	2-7
2.3.3	DNS 系統是最後一致性的 .....	2-10
2.4	資料一致性模型與 BASE .....	2-10
2.4.1	資料一致性模型 .....	2-10
2.4.2	BASE ( Basically Available , Soft-state , Eventual consistency ) ...	2-12
2.5	資料一致性實現方法.....	2-12
2.5.1	Quorum 系統 NRW 策略.....	2-13
2.5.2	時間戳記策略.....	2-14
2.5.3	向量時鐘.....	2-18

## Chapter 03 NoSQL 的水平擴充與其他基礎知識

3.1	所有資料儲存在一個伺服器上.....	3-2
3.2	分片 (Sharding).....	3-3
3.3	主從複製.....	3-5
3.4	對等 (Peer To Peer) 複製 .....	3-6



3.5	複製和分片的同時使用 .....	3-7
3.6	資料水平擴充的方法歸納 .....	3-8
3.7	分片對資料的劃分方式 .....	3-8
	3.7.1 Range-Based Partitioning .....	3-9
	3.7.2 Round-Robin .....	3-12
3.8	一致性 hash 演算法 (Consistent Hashing) .....	3-13
	3.8.1 基本場景 .....	3-13
	3.8.2 hash 演算法和單調性 .....	3-14
	3.8.3 Consistent Hashing 演算法的原理 .....	3-14
	3.8.4 虛擬節點 .....	3-17
3.9	磁碟的讀寫特點及五分鐘法則 .....	3-18
	3.9.1 磁軌、磁區、磁柱和磁頭數 .....	3-19
	3.9.2 固態硬碟 (SSD)：隨機讀寫速度快 .....	3-20
	3.9.3 記憶體：讀寫速度極快 .....	3-20
	3.9.4 五分鐘法則 .....	3-21
3.10	不要刪除資料 .....	3-21

## 2 Part

## 列式 NoSQL 系統

### Chapter 04 BigTable 與 Google 雲端運算原理

4.1	雲端運算 .....	4-2
	4.1.1 雲端運算的概念 .....	4-2
	4.1.2 雲端運算發展現狀 .....	4-6
	4.1.3 雲端運算實現機制 .....	4-7
	4.1.4 網格計算與雲端運算 .....	4-9
4.2	Google 檔案系統 GFS .....	4-13
	4.2.1 系統架構 .....	4-14
	4.2.2 容錯機制 .....	4-17
	4.2.3 系統管理技術 .....	4-18
4.3	平行資料處理 MapReduce .....	4-19

4.3.1	產生背景.....	4-19
4.3.2	程式設計模型.....	4-20
4.3.3	實現機制.....	4-21
4.4	分散式鎖服務 Chubby.....	4-25
4.4.1	Paxos 演算法.....	4-25
4.4.2	Chubby 系統設計.....	4-26
4.4.3	Chubby 檔案系統.....	4-29
4.4.4	通訊協定.....	4-30
4.4.5	正確性與效能.....	4-33
4.5	分散式結構化資料表 BigTable.....	4-34
4.5.1	設計動機與目標.....	4-35
4.5.2	資料模型.....	4-37
4.5.3	系統架構.....	4-37
4.5.4	主要伺服器.....	4-38
4.5.5	子表伺服器.....	4-39
4.5.6	效能最佳化.....	4-43

## **Chapter 05** Google 雲端運算的開放原始碼版本——Hadoop

5.1	Hadoop 簡介.....	5-2
5.1.1	Hadoop 發展史.....	5-2
5.1.2	Apache Hadoop 專案及系統結構.....	5-4
5.2	Hadoop 產生的原因.....	5-5
5.2.1	巨量的資料.....	5-5
5.2.2	資料的儲存和分析.....	5-7
5.3	Hadoop 和其他系統的比較.....	5-8
5.3.1	和關聯式資料庫管理系統 (RDBMS) 的比較.....	5-9
5.3.2	和網格計算的比較.....	5-10
5.4	HDFS 的架構設計.....	5-12
5.4.1	前提和設計目標.....	5-12
5.4.2	Namenode 和 Datanode.....	5-13
5.4.3	檔案系統的 Namespace.....	5-13
5.4.4	資料複製.....	5-14

5.4.5	檔案系統中繼資料的持久化 .....	5-15
5.4.6	通訊協定.....	5-16
5.4.7	穩固性 .....	5-17
5.4.8	資料組織.....	5-18
5.4.9	可存取性.....	5-19
5.4.10	空間的回收 .....	5-19
5.5	安裝 Hadoop .....	5-20
5.5.1	安裝 JDK 1.7 .....	5-20
5.5.2	安裝 Hadoop.....	5-22
5.6	HDFS 操作.....	5-28
5.6.1	使用 FS Shell 指令操作 HDFS .....	5-28
5.6.2	程式設計讀寫 HDFS.....	5-33
5.7	Hadoop 中的 MapReduce 模型 .....	5-35
5.7.1	MapReduce 計算模型.....	5-35
5.7.2	Hadoop 中的 Hello World 程式.....	5-36
5.7.3	執行 MapReduce 程式.....	5-42
5.7.4	Hadoop 中的 Hello World 程式——新的 API .....	5-44
5.7.5	MapReduce 的資料流程和控制流.....	5-46
5.8	Zookeeper .....	5-48
5.8.1	Zookeeper 設定安裝 .....	5-48
5.8.2	Zookeeper 的資料模型.....	5-50
5.8.3	Zookeeper 的基本使用.....	5-51
5.8.4	ZooKeeper 典型的應用場景 .....	5-54
5.8.5	統一命名服務 (Name Service).....	5-54
5.8.6	共用鎖 (Locks).....	5-57
5.8.7	佇列管理.....	5-58
5.8.8	Zookeeper 歸納 .....	5-61
5.9	HBase.....	5-61
5.9.1	簡介.....	5-61
5.9.2	邏輯視圖.....	5-62
5.9.3	實體儲存.....	5-63
5.9.4	系統架構.....	5-67
5.9.5	關鍵演算法 / 流程 .....	5-69

5.10	HBase 的安裝和設定.....	5-74
5.11	HBase 使用實例.....	5-76

# 3 Part

## Key/Value NoSQL 系統

### Chapter 06 Dynamo：Amazon 的高可用鍵值對儲存

6.1	簡介.....	6-2
6.2	背景.....	6-3
6.2.1	系統假設和要求.....	6-4
6.2.2	服務水準協定 (SLA).....	6-5
6.2.3	設計考慮.....	6-7
6.3	系統架構.....	6-9
6.3.1	系統介面.....	6-9
6.3.2	劃分演算法.....	6-10
6.3.3	複製.....	6-11
6.3.4	版本的資料.....	6-11
6.3.5	執行 get() 和 put() 操作.....	6-14
6.3.6	故障處理：暗示移交 (Hinted Handoff).....	6-16
6.3.7	處理永久性故障：備份同步.....	6-16
6.3.8	會員和故障檢測.....	6-17
6.3.9	增加 / 刪除儲存節點.....	6-19
6.4	實現.....	6-19
6.5	Amazon 使用的經驗與教訓.....	6-21
6.5.1	平衡效能和耐久性.....	6-22
6.5.2	確保均勻的負載分佈.....	6-24
6.5.3	不同版本：何時以及有多少.....	6-28
6.5.4	用戶端驅動或伺服器驅動協調.....	6-29
6.5.5	權衡後台和前台工作.....	6-30
6.5.6	討論.....	6-31
6.6	結論.....	6-31

## Chapter 07 LevelDb——出自 Google 的 Key-Value 資料庫

7.1	LevelDb 簡介 .....	7-2
7.2	LevelDb 的靜態部分 .....	7-3
7.2.1	整體架構 .....	7-3
7.2.2	log 檔案 .....	7-6
7.2.3	SSTable 檔案 .....	7-7
7.2.4	MemTable 詳解 .....	7-11
7.3	LevelDb 的動態部分 .....	7-12
7.3.1	寫入與刪除記錄 .....	7-12
7.3.2	讀取記錄 .....	7-13
7.3.3	Compaction 操作 .....	7-16
7.3.4	LevelDb 中的 Cache .....	7-19
7.3.5	Version、VersionEdit 和 VersionSet .....	7-21

## Chapter 08 Redis 實戰

8.1	Redis 安裝與準備 .....	8-2
8.1.1	下載與安裝 .....	8-2
8.1.2	設定檔修改 .....	8-3
8.1.3	啟動 Redis .....	8-5
8.1.4	停止 Redis .....	8-6
8.2	Redis 所支援的資料結構 .....	8-7
8.2.1	String .....	8-8
8.2.2	List .....	8-12
8.2.3	Set .....	8-18
8.2.4	Hash/ 雜湊 / 雜湊 .....	8-22
8.2.5	有序集合 / Zset .....	8-26
8.3	Key 操作指令 .....	8-33
8.3.1	概述 .....	8-33
8.3.2	指令範例 .....	8-34
8.4	事物 .....	8-38
8.4.1	事物概述 .....	8-38
8.4.2	相關指令 .....	8-39

8.4.3	指令範例.....	8-40
8.4.4	WATCH 指令和以 CAS 為基礎的樂觀鎖.....	8-41
8.5	Redis 的主從複製.....	8-42
8.5.1	Redis 的 Replication.....	8-42
8.5.2	Replication 的工作原理.....	8-43
8.5.3	如何設定 Replication.....	8-43
8.5.4	應用範例.....	8-44
8.6	Redis 的持久化.....	8-45
8.6.1	持久化機制.....	8-45
8.6.2	RDB 機制的優勢和劣勢.....	8-45
8.6.3	AOF 機制的優勢和劣勢.....	8-46
8.6.4	其他.....	8-47
8.7	Redis 的虛擬記憶體.....	8-48
8.7.1	簡介.....	8-48
8.7.2	應用場景.....	8-49
8.7.3	設定.....	8-49
8.8	pipeline/ 管線.....	8-51
8.8.1	請求回應協定和 RTT.....	8-51
8.8.2	管線 (pipelining).....	8-51
8.8.3	Benchmark.....	8-52
8.9	實例.....	8-53

## 4 Part

## 文件型 NoSQL 系統

### Chapter 09 針對文件的資料庫 CouchDB

9.1	CouchDB 介紹.....	9-2
9.1.1	基本概念.....	9-2
9.1.2	擴充概念.....	9-2
9.2	CouchDB 安裝與設定.....	9-3
9.3	REST API.....	9-5

9.3.1	資料庫 REST API .....	9-5
9.3.2	文件 REST API .....	9-5
9.3.3	視圖 REST API .....	9-6
9.3.4	附件 REST API .....	9-6
9.4	為應用建模 .....	9-7
9.4.1	描述實體 .....	9-7
9.4.2	描述一對一和一對多關聯性 .....	9-8
9.4.3	描述多對多關係 .....	9-9
9.5	實戰開發 .....	9-11
9.5.1	開發 Web 應用 .....	9-11
9.5.2	使用 CouchDB jQuery 外掛程式 .....	9-13
9.5.3	範例應用建模 .....	9-14
9.5.4	管理文件 .....	9-15
9.5.5	視圖 .....	9-19
9.6	進階內容 .....	9-26
9.6.1	許可權控制與安全 .....	9-26
9.6.2	文件更新驗證 .....	9-27
9.6.3	分組 .....	9-27
9.6.4	鍵的排序 .....	9-28

---

## Chapter 10 MongoDB 實戰

10.1	為什麼要使用 MongoDB .....	10-2
10.1.1	不能確定的表結構資訊 .....	10-2
10.1.2	序列化可以解決一切問題嗎 .....	10-2
10.1.3	無需定義表結構的資料庫 .....	10-3
10.2	MongoDB 的優勢和不足 .....	10-3
10.2.1	無表結構 .....	10-3
10.2.2	容易擴充 .....	10-4
10.2.3	豐富的功能 .....	10-5
10.2.4	效能卓越 .....	10-5
10.2.5	簡便的管理 .....	10-5
10.2.6	MongoDB 的不足 .....	10-6
10.3	基本概念 .....	10-6

10.4	Linux 下 MongoDB 的安裝和設定、啟動和停止 .....	10-7
10.4.1	下載 .....	10-7
10.4.2	安裝 .....	10-7
10.4.3	啟動資料庫 .....	10-9
10.4.4	停止資料庫 .....	10-12
10.5	建立、更新及刪除文件 .....	10-13
10.5.1	連接資料庫 .....	10-13
10.5.2	插入記錄 .....	10-14
10.5.3	_id key .....	10-15
10.5.4	修改記錄 .....	10-16
10.5.5	刪除記錄 .....	10-16
10.6	查詢記錄 .....	10-17
10.6.1	普通查詢 .....	10-17
10.6.2	條件查詢 .....	10-18
10.6.3	findOne() 語法 .....	10-19
10.6.4	透過 limit 限制結果集數量 .....	10-19
10.7	高級查詢 .....	10-19
10.7.1	條件運算符號 .....	10-20
10.7.2	\$all 比對所有 .....	10-20
10.7.3	\$exists 判斷欄位是否存在 .....	10-20
10.7.4	Null 值處理 .....	10-21
10.7.5	\$mod 取模運算 .....	10-22
10.7.6	\$ne 不等於 .....	10-22
10.7.7	\$in 包含 .....	10-23
10.7.8	\$nin 不包含 .....	10-23
10.7.9	\$size 陣列元素個數 .....	10-24
10.7.10	正規表示法比對 .....	10-24
10.7.11	JavaScript 查詢和 \$where 查詢 .....	10-24
10.7.12	count 查詢記錄筆數 .....	10-25
10.7.13	skip 限制傳回記錄的起點 .....	10-25
10.7.14	sort 排序 .....	10-26
10.7.15	游標 .....	10-26



10.8	MapReduce .....	10-27
10.8.1	Map .....	10-28
10.8.2	Reduce .....	10-29
10.8.3	Result .....	10-29
10.8.4	Finalize .....	10-30
10.8.5	Options .....	10-31
10.9	索引 .....	10-31
10.9.1	基礎索引 .....	10-32
10.9.2	文件索引 .....	10-32
10.9.3	組合索引 .....	10-33
10.9.4	唯一索引 .....	10-33
10.9.5	強制使用索引 .....	10-33
10.9.6	刪除索引 .....	10-35
10.10	性能優化 .....	10-35
10.10.1	explain 執行計畫 .....	10-35
10.10.2	最佳化器 Profile .....	10-36
10.10.3	效能最佳化舉例 .....	10-38
10.11	性能監控 .....	10-40
10.11.1	mongosniff .....	10-40
10.11.2	Mongostat .....	10-41
10.11.3	db.serverStatus .....	10-42
10.11.4	db.stats .....	10-44
10.11.5	協力廠商工具 .....	10-44
10.12	Replica Sets 複製集 .....	10-45
10.12.1	部署 Replica Sets .....	10-45
10.12.2	主從操作記錄檔 oplog .....	10-49
10.12.3	主從設定資訊 .....	10-50
10.12.4	管理維護 Replica Sets .....	10-51
10.12.5	增減節點 .....	10-54
10.13	Sharding 分片 .....	10-59
10.13.1	建立 Sharding Cluster .....	10-60
10.13.2	管理維護 Sharding .....	10-63