

# Redis集群服務器-高可用調研隨筆 - Snowolf的意境空間！ - ITeye技術網站

 snowolf.iteye.com/blog/1657488

## 一、Redis與MySQL對比

相同點：

- **Master-Slave**架構，集群架構下無法很好的完成數據拷貝，確保數據一致性。
- 支持數據文件持久化存儲，但數據文件過大時，宕機重啟可能存在安全隱患。

不同點：

- **Redis**時效性能遠比**MySQL**要高得多，支持複雜的數據類型，基本上都是內存操作，效率遠勝於**MySQL**。
- **Redis**是**NoSQL**型數據庫，或者說是**Store-Cache**型數據庫，而**MySQL**屬於**RDBMS**，關係型數據庫，雖然自身做了查詢緩存，但效果一般。
- **Redis**支持以數據橫向切分，便於根據業務需求擴展，鍵值構建類似於數據庫索引，靈活高效，不必忌諱數據之間的關聯。**Redis**依賴其數據類型，完成交集、並集、補集計算，更勝一籌。
- **MySQL**在數據量和連接數量上都有上限：單表數據量**500**萬條記錄，並發連接數**3000**/秒。
- **Redis**定時、定量將數據保存至本地，命中數據大部分存活在內存中，降低了數據文件讀取消耗；數據更新，以內存修改為主，不存磁盤在IO消耗。

結論：

- 兩者在高並發環境下，依靠自身的**Master-Slave**架構，完成橫向擴容都存在難度。要控制每個實例的數據文件大小，留有足夠的磁盤，內存空間。確保宕機後，服務可恢復。
- **Redis**更適合作為頻繁查詢為主，對數據進行交集、補集、並集操作，類似於SNS用戶社區關聯關係展現等，有著良好的數據類型支持，以及高效性。

## 二、Redis與Memcached，以及EhCache/OSCache

**EhCache/OSCache**、**Memcached**可謂是緩存架構裡的一朵朵奇葩。

- **EhCache**、**OSCache**在幾年前，都是小應用最喜歡使用緩存實現。尤其是當應用之間不需要考慮數據一致性問題時，幾乎無所不能。但到了分佈式緩存時代，雖然兩者也提供了相應的架構實現，但實現成本較高，且存在一定風險。例如**EhCache**，提供了**EhCache Server**架構，主要通過各個**EhCache**集群網絡多播等方式同步數據。但高並發下，網絡多播易演變成網絡風暴。增加了系統安全隱患。

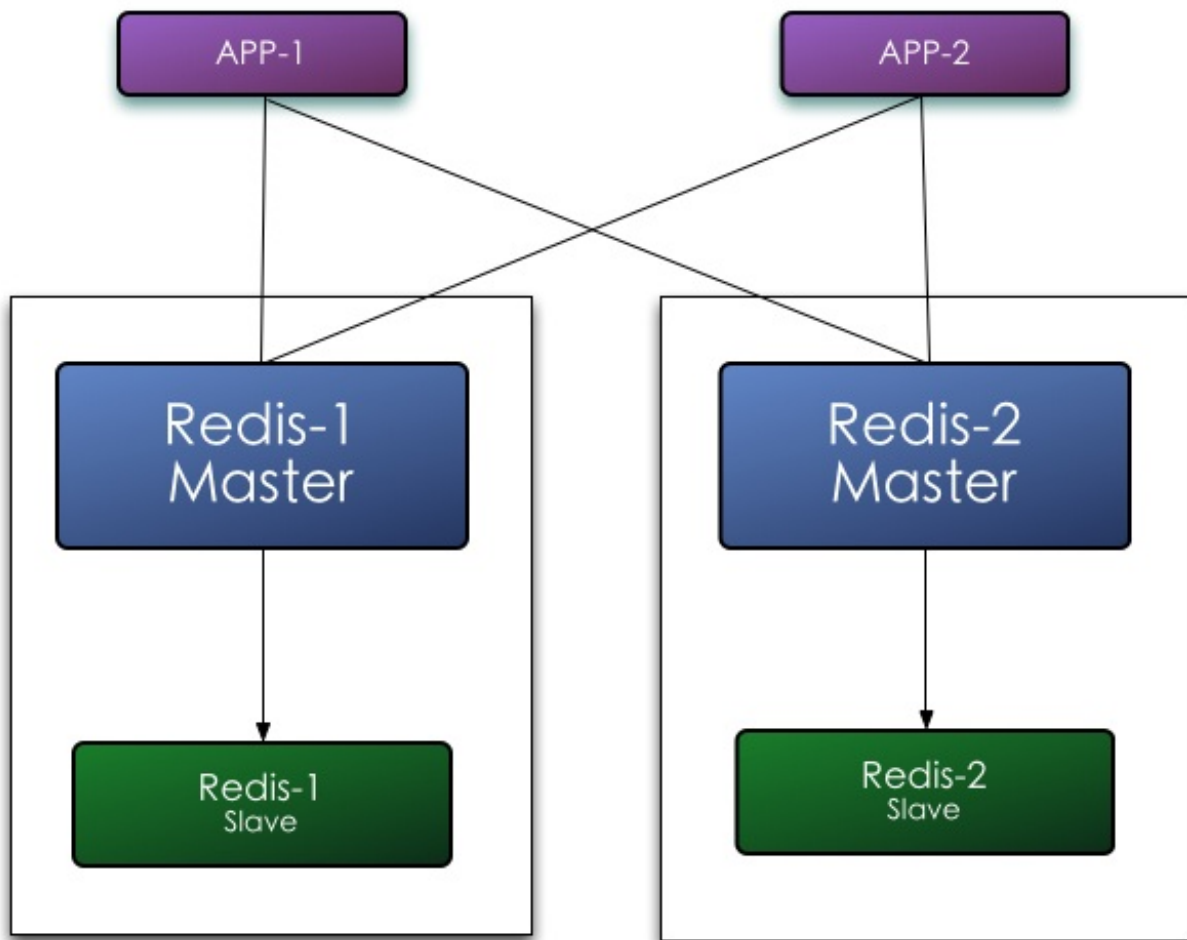
- **Memcached**走了另一條路，通過一致性哈希根據Key與Server的Hash對應關係，或者餘數算法等，將數據散落在不同的**Server**上，確保每個**Server**上都能平均**Cache**數據。也基緣於此，**Memcached**適合進行快速地橫向擴展。不必考慮磁盤存儲，只需要提供一個內存足夠大的**Server**主機即可。
- **Memcached**也有瓶頸，單個**ObjectSize**不得大於1MB，**KeySize**不得大於250個字符，**Write**要比**Read**耗時長，對大對象做**Write Cache**時尤為明顯。因此，**Memcached**適合小數據量對象的**Cache**。且當服務器宕機時，瘋漲的數據庫操作IO，很可能將數據庫服務器拖垮。

**Redis**可以簡單理解為**Store-Cache**，用作**Cache**：**ObjectSize**支持1GB，**KeySize**支持512Bytes，並支持複雜數據類型，可在內存中直接排序等。但**Redis**不能像**Memcached**那樣實現**Sharding**，直接進行橫向擴展。且自身作為**Database**時，也可能存在單點故障風險。

### 三、基於Redis高可用服務器架構簡單設想

1. Redis以Master-Slave為單元，公用虛擬IP，通過Keepalive實現自動切換，完成主從互備。
2. 通過Redis Client，如Jedis，在Client端完成Sharding，訪問多個Redis Server。

### 3. 讀寫分離，Write-Master，Read-Slave。



未盡之處，若橫向擴容時，Client一致性哈希，是否會由原先的A Server指向，改為新進的C Server？單純拷貝數據文件可解決單點到雙點的實現。但多點服務器擴容，尚未做一致性哈希嘗試，有一定的風險。

完全是個人頭腦風暴，歡迎拍磚。 😊

關於調優 <http://www.oschina.net/translate/redis-latency-problems-troubleshooting?from=20130317>