



# VictoriaMetrics 去重機制



以繁體中文詳細說明:VictoriaMetrics 去重 (Deduplication)



## VictoriaMetrics 去重 (Deduplication) 詳細說明

VictoriaMetrics 是一款高效能、可水平擴展的時序數據庫 (TSDB)，廣泛用於監控與度量數據的存儲與查詢。當使用多個數據來源或有重複數據時，去重 (Deduplication) 機制能有效降低存儲空間佔用、提升查詢效率，並確保數據的一致性。

## 1. VictoriaMetrics 為何需要去重 (Deduplication) ？

在某些場景中，可能會出現重複的時序數據，例如：

- **多個 Prometheus 寫入相同數據**：如果 Prometheus HA（高可用）部署了多個實例並向 VictoriaMetrics 寫入相同的度量數據，則會產生重複數據。
- **集群模式下的數據合併**：在 VictoriaMetrics 集群模式中，不同節點可能會收到相同的數據，需要去重以避免重複存儲。
- **客戶端網路異常導致的重試**：某些客戶端可能因為網路問題重試寫入，從而產生重複數據。

VictoriaMetrics 透過內建的去重機制，確保最終存儲的數據唯一且有效。

## 2. VictoriaMetrics 的去重方式

VictoriaMetrics 提供了多種去重方法，主要包括：



## (1) 即時去重 (Ingestion Deduplication)

當數據寫入時，VictoriaMetrics 會透過**相同的標籤組 (label set)** 與**時間戳**來識別重複數據，並僅保留最新的一筆數據。

### 啟用方法

在 `vmstorage` (單機模式) 或 `vminsert` (集群模式) 中，啟用 `-dedup.minScrapeInterval=<duration>` 參數。例如：

```
vminsert -dedup.minScrapeInterval=1s
```

此設定的意思是：

- 在 `1s` 內，如果有來自同一個 `metric` (相同的 metric name + labels) 且時間戳相同的數據，則 VictoriaMetrics 只保留最新的數據，丟棄舊的。

**“注意：***VictoriaMetrics 預設的去重間隔為 `0` (即不啟用去重)。***”**

## (2) 查詢時去重 (Query-time Deduplication)

VictoriaMetrics 允許使用 `dedup()` 函數，在查詢時執行數據去重，而非在寫入時去重。

### 使用方法

當查詢數據時，可以使用 `dedup()` 函數，例如：

```
dedup(avg_over_time(node_cpu_seconds_total[5m]))
```

這表示：

- 先對 `node_cpu_seconds_total` 這個指標做 `5 分鐘` 滑動窗口計算平均值 (`avg_over_time`)
- 再執行去重處理，以確保不會計算重複數據

此外，也可直接在 `PromQL` 查詢中應用：





```
curl "http://<victoriametrics-server>:8428/api/v1/query?
query=dedup(rate(http_requests_total[5m]))"
```

這樣可確保查詢結果排除重複數據。

### (3) 物理存儲層面的去重

VictoriaMetrics 也會在後台執行壓縮與合併操作，進一步降低重複數據佔用的存儲空間，這主要透過：

- **數據壓縮 (Compaction)**：定期合併相鄰區間的數據，刪除重複值
- **分片與合併 (Merging of Chunks)**：優化數據存儲，使查詢時能更高效讀取

這些機制在 VictoriaMetrics 後台自動執行，無需額外手動干預。

---

## 3. 去重的影響與考量

### (1) 優勢

- **減少存儲空間**：重複數據被移除，可顯著降低存儲需求
- **提升查詢效率**：避免在計算時納入冗餘數據，減少計算成本
- **保證數據一致性**：確保相同時間戳的指標數據唯一，減少數據衝突問題

### (2) 可能的問題

- **過度去重可能丟失數據**：如果 `-dedup.minScrapeInterval` 設定過大，可能導致應該被保留的數據被刪除
- **影響 Prometheus HA 部署**：若在 Prometheus 高可用 (HA) 模式下啟用去重，可能導致某些節點的數據被丟棄，影響 HA 效果



## 4. 最佳實踐

- 如果使用 **Prometheus HA**，建議在查詢時去重（**Query-time Deduplication**）
    - 這樣可以確保寫入時保留所有數據，而在查詢時再進行去重，避免數據丟失
  - 如果數據來源固定，建議啟用寫入去重（**Ingestion Deduplication**）
    - 可透過 `-dedup.minScrapeInterval` 設定合理的時間間隔（如 `1s`），減少存儲壓力
  - 查詢時使用 `dedup()` 避免重複計算
    - 在 PromQL 查詢中適當加入 `dedup()` 來優化結果
- 

## 5. 結論

VictoriaMetrics 提供了靈活的去重機制，包括寫入時去重、查詢時去重與物理存儲層面的壓縮優化。合理配置去重功能可以有效降低存儲成本、提升查詢效率，並確保數據一致性。對於不同的使用場景，可以選擇適合的去重策略，來最佳化時序數據的存儲與計算效率。

如果你的數據來源會產生大量重複數據，建議適當啟用 `-dedup.minScrapeInterval` 參數，並在查詢時搭配 `dedup()` 函數，以獲得最佳的查詢與存儲效能。