

SOOP-CLM PLUS 加值服務功能模組

Service-Oriented Operation Portal - Centralized Log Management PLUS

產品定位

SOOP-CLM PLUS 是經由國內各大企業專案導入 SOOP-CLM 後，歸納出類似的需求，進而研發出多樣的加值服務功能模組，顧名思義是用以擴充 SOOP-CLM 的原生功能，希望能讓 SOOP-CLM 更加符合國內客戶的需求，同時提升客戶後續維運管理 SOOP-CLM 平台的便利性，如今已經有眾多客戶選擇以加值服務功能模組的方式導入各式新功能，在享受新功能帶來的便利性之外，也獲得產品化模組帶來的穩定與可靠度。

產品列表

SOOP-CLM PLUS Archive - 日誌備份加值服務功能模組

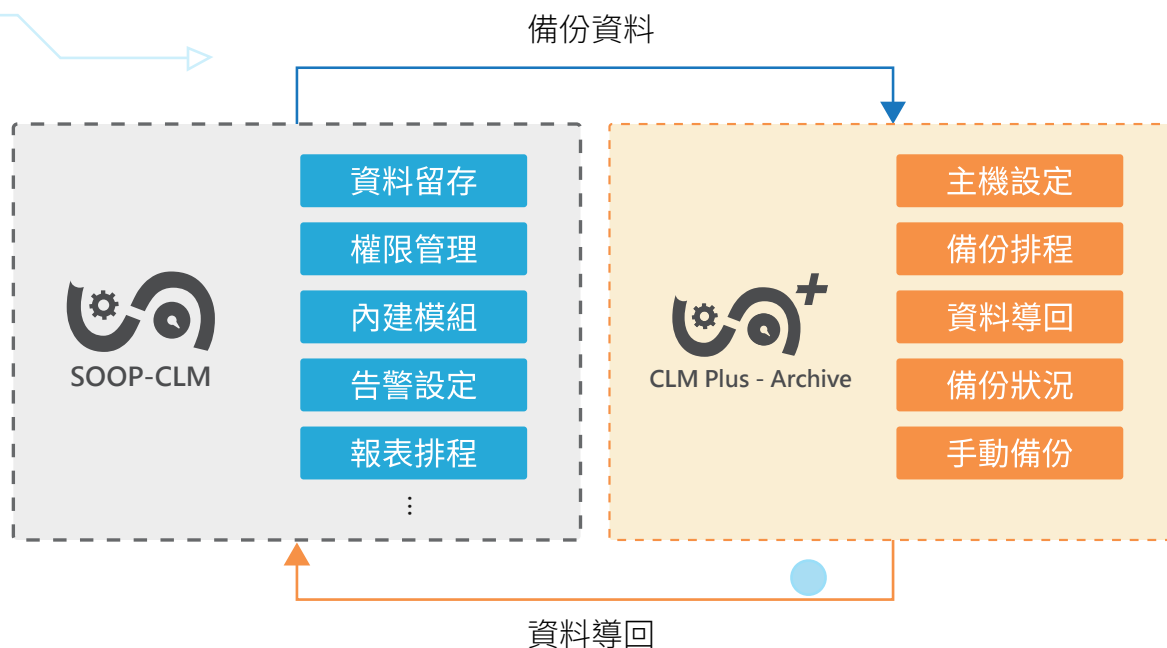
SOOP-CLM PLUS Data Queue - 日誌傳輸緩衝加值服務功能模組

產品介紹

針對 SOOP-CLM 集中式日誌管理平台，既要滿足長期日誌收集，也要維持效能穩定，區分冷熱資料是重要的。

熱資料配合日常維運所需，而冷資料，則是需留存長期資料，配合稽核、或未來有需要查找使用。若沒有區分，不僅會導致日誌管理平台占用過多硬體資源，還會降低查找日誌的速度，影響日常維運效率。

SOOP-CLM PLUS Archive - 日誌備份增值服務功能模組，就是為儲存冷資料而設計，讓客戶將 SOOP-CLM 收集到的日誌內容，另外導出到外部硬碟儲存設備，進而節省伺服器主機裡的儲存空間與成本，提升日常維運效率。



```
int x = 0;
Console.WriteLine("Enter X : ");
x = int.Parse(Console.ReadLine());
if (x > 10 && x < 100)
{
    Console.WriteLine("SUM = ");
}
Console.ReadLine();
```

產品功能

1. 提供資料壓縮功能，可將 SOOP-CLM 中的日誌資料進行無失真壓縮，資料壓縮比為 2 以上，最高可達 10:1 的壓縮比；即可節省一半以上的資料儲存空間，有效節省冷資料儲存空間。
2. 資料壓縮作業可自動排程進行壓縮儲存，減少人力操作維運成本。
3. 判斷每日新增之資料，進行差異及增量備份，縮短備份所需時間。
4. 可依客戶需求選擇要長期壓縮備份的資料及其備份保留時間。如：OS 資料備份儲存 5 年、網路設備的資料備份儲存 1 年，而 NetFlow 的資料不進行長期資料備份儲存。
5. 長期儲存日誌稽核資料，以符合資安法規要求。
6. 自動化進行備份任務，並自動監控備份任務執行情況，有效降低維運人員負擔。
7. 可進行磁碟、磁帶本地、遠端及雲端備份，隨時確保資料安全無虞。
8. 以快照避免人為疏失，並以複寫做為災難復原對策。
9. 備份資料方式具備密碼驗證機制，以確保資料保存安全性。
10. 可將日誌資料儲存其他伺服器中及外接儲存裝置中，如磁帶區等。
11. 針對已壓縮並儲存到外部儲存空間的資料，可將指定日期的資料導回 SOOP-CLM 進行查找。
12. 在導回已壓縮儲存的資料時，系統將進行資料正確性比對，若資料被竄改過，系統會立即示警，該資料將無法導回 SOOP-CLM，以確保資料正確性。
13. 具備自我監控功能，當監控項目有服務異常，可第一時間通知第一線維運人員。

產品介紹

由於日誌的產生與當時的交易數量或系統穩定度相關，特別是發生異常時，常常會產生超乎預期的日誌數量，加上受限於環境，容易造成傳輸失敗、掉資料的情形。

SOOP-CLM PLUS Data Queue - 日誌傳輸緩衝增值服務功能模組，類似滯洪池的概念，避免出乎預期的強降雨影響河道排水功能，讓突如其來的日誌都能完整的傳輸。

SOOP-CLM PLUS Data Queue 採三個節點互相備援，多點收集資料。遇到爆量資料的時候，依然可以分流穩定傳輸。若遇到網路斷線，SOOP-CLM PLUS Data Queue 上也有備份，避免資料遺失，加強 SOOP-CLM 日誌收集平台整架運作架構的完整性。



```
int x = 0;
Console.WriteLine("Enter X : ");
x = int.Parse(Console.ReadLine());
if (x > 10 && x < 100)
{
    Console.WriteLine("SUM = ");
}
Console.ReadLine();
```

產品功能

1. 可以支援獨立於 SOOP-CLM 的資料緩衝與傳輸機制，避免瞬間爆量的資料影響 SOOP-CLM 正常運作；在瞬間大流量的使用場景下，可以避免應用服務的崩潰，不會因為超過負載的流量請求而停止運作。
2. 具備分散式架構，可有效進行負載平衡；當日誌量增加時，亦支援垂直擴充方式，以維持正常的效能運作。
3. 具備 High availability 機制，內建 partition 和 replica 副本機制，在單一 Data Queue 節點失效時，資料不會遺失，亦可保證資料不丟失及確保資料的時間順序性。
4. 構建於 Broker 架構 Pull 的模式來處理日誌收集和傳輸，以達到高吞吐量 (throughput) 並維持低延遲；在有大量訊息累積的情況下，效能不會遞減。
5. 可以提供資料暫存功能，若無法與 SOOP-CLM 維持資料傳輸時，可以暫存資料避免資料丟失。
6. 若資料格式異動，或想要在舊資料中解析新的欄位，可拉取 Data Queue 的資料進行資料欄位重構。
7. 具備自我監控功能，當監控項目有服務異常，可第一時間通知第一線維運人員。



```
int x = 0;
Console.Write("Enter X : ")
x = int.Parse(Console.ReadLine());
if (x > 10 && x < 100)
{
    Console.Write("SUM = ")
}
Console.ReadLine();
```