一銀教育訓練-
SQL資料庫介紹

Debby Hsu

Agenda

1. SQL Server 架構

2. SQL Server 參數設定

3. 效能監控方式

4. 成效數據對比

◆提升Memory – 750g Ram

◆提升CPU core數 – 28 core

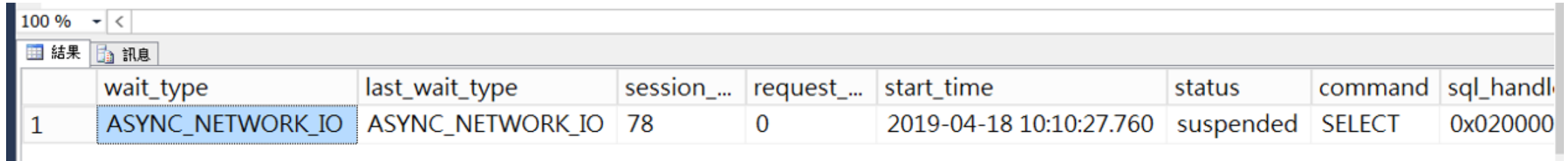＞＞＞升級 SQL enterprise Core Based version

```sql
SELECT TEXT,session_id,status,command,cpu_time,total_elapsed_time
FROM sys.dm_exec_requests a
CROSS APPLY sys.dm_exec_sql_text(sql_handle) AS b
ORDER BY total_elapsed_time desc
```

	TEXT	session_id	status	command	cpu_time	total_elapsed_time
1	SELECT SCHEMA_NAME(tbl.schema_id) AS [Schema], tbl.name AS [Name], tbl.object_id AS [ID] FRO...	219	suspended	SELECT	39018	572144
2	SELECT SCHEMA_NAME(tbl.schema_id) AS [Schema], tbl.name AS [Name], tbl.object_id AS [ID] FRO...	499	suspended	SELECT	41416	520168
3	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	384	runnable	SELECT	132549	304750
4	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	474	runnable	SELECT	30450	174810
5	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	358	runnable	SELECT	29220	168444
6	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	328	runnable	SELECT	31443	110958
7	CREATE PROCEDURE [dbo].[SAR_CaseAssign_GetPendingCases] @ForAutoAssign BIT = 0, @Assi...	309	suspended	SELECT	285489	72719
8	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	287	runnable	SELECT	11283	51113
9	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	331	runnable	SELECT	5759	50589
10	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	355	runnable	SELECT	5463	50549
11	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	479	runnable	SELECT	7397	49594
12	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	408	runnable	SELECT	8203	49075
13	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	315	runnable	SELECT	10151	47524
14	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	510	runnable	SELECT	11693	46996
15	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	532	runnable	SELECT	9230	46505
16	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	317	runnable	SELECT	10930	45971
17	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	523	runnable	SELECT	9953	45484
18	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	522	runnable	SELECT	9253	44956
19	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	537	runnable	SELECT	6633	44436
20	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	472	runnable	SELECT	4982	43404
21	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	425	runnable	SELECT	17968	42389
22	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	406	runnable	SELECT	4631	41314
23	SELECT X.ent_num, X.alt_num, X.sdn_name, X.sdn_type, ISNULL(X.sdn_grp_num, 0) AS sdn_grp_nu...	538	runnable	SELECT	8703	39774
24	CREATE PROC [dbo].[DBCHK_GetResultFilter] @GRIDTYPE CHAR(1), @ROLE INT, @HANDLE...	348	running	SELECT	4585	27885
25	CREATE PROC [dbo].[DBCHK_GetResultFilter] @GRIDTYPE CHAR(1), @ROLE INT, @HANDLE...	464	runnable	SELECT	5391	24728
26	CREATE PROCEDURE [dbo].[UserDashboard_GetQueueStatistic] @UserId NVARCHAR(40) AS BE...	360	runnable	SELECT	1690	24217

SELECT wait_type,last_wait_type,* FROM SYS.dm_exec_requests WHERE SESSION_ID =XX
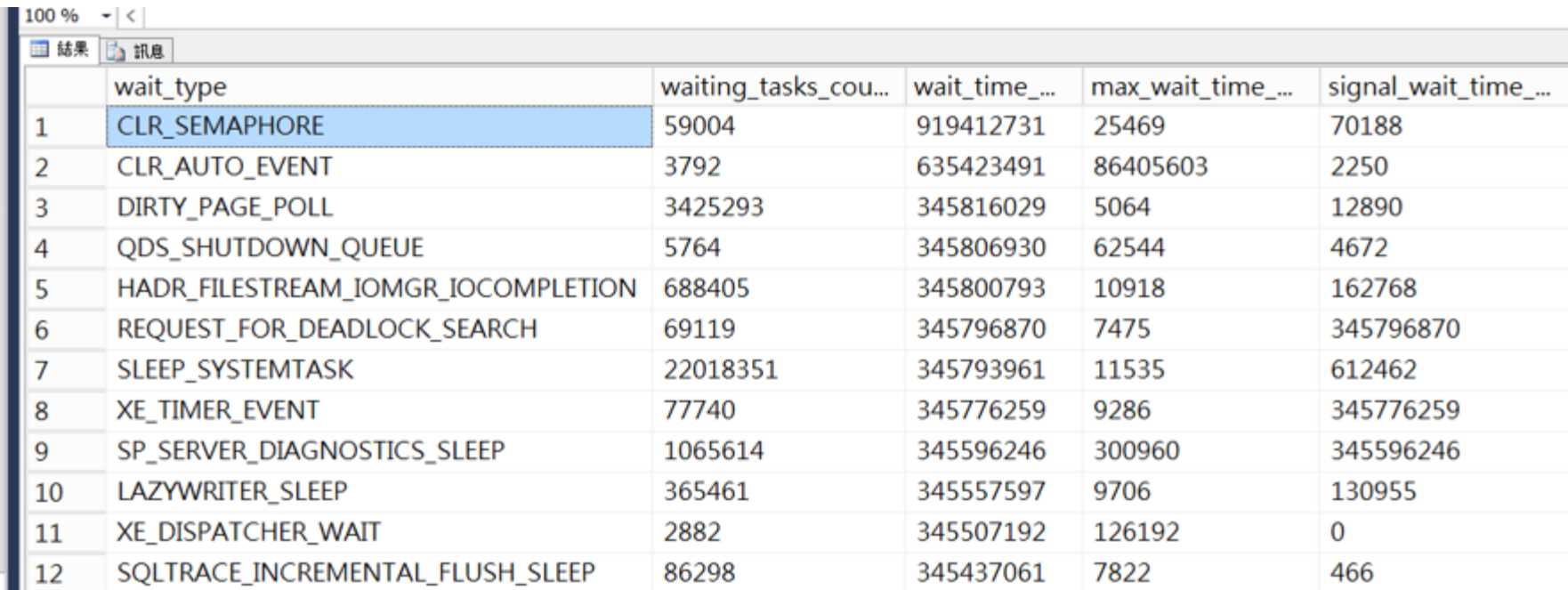
select * from sys.dm_os_wait_stats order by 3 desc

	wait_type	waiting_tasks_cou...	wait_time_...	max_wait_time_...	signal_wait_time_...
1	CLR_SEMAPHORE	59004	919412731	25469	70188
2	CLR_AUTO_EVENT	3792	635423491	86405603	2250
3	DIRTY_PAGE_POLL	3425293	345816029	5064	12890
4	QDS_SHUTDOWN_QUEUE	5764	345806930	62544	4672
5	HADR_FILESTREAM_IOMGR_IOCOMPLETION	688405	345800793	10918	162768
6	REQUEST_FOR_DEADLOCK_SEARCH	69119	345796870	7475	345796870
7	SLEEP_SYSTEMTASK	22018351	345793961	11535	612462
8	XE_TIMER_EVENT	77740	345776259	9286	345776259
9	SP_SERVER_DIAGNOSTICS_SLEEP	1065614	345596246	300960	345596246
10	LAZYWRITER_SLEEP	365461	345557597	9706	130955
11	XE_DISPATCHER_WAIT	2882	345507192	126192	0
12	SQLTRACE_INCREMENTAL_FLUSH_SLEEP	86298	345437061	7822	466

https://www.sqlshack.com/sql-server-wait-types/#l

```sql
SELECT TOP 100
  qs.total_elapsed_time / qs.execution_count / 1000000.0 AS average_seconds, qs.total_elapsed_time / 1000000.0
 AS total_seconds,qs.execution_count,SUBSTRING (qt.text,qs.statement_start_offset/2, (CASE WHEN
qs.statement_end_offset = -1 THEN LEN(CONVERT(NVARCHAR(MAX), qt.text)) * 2
ELSE qs.statement_end_offset END - qs.statement_start_offset)/2) AS individual_query,o.name AS object_name,
DB_NAME(qt.dbid) AS database_name FROM sys.dm_exec_query_stats qs CROSS APPLY
sys.dm_exec_sql_text(qs.sql_handle) as qt LEFT OUTER JOIN sys.objects o ON qt.objectid = o.object_id
where qt.dbid = DB_ID()  ORDER BY average_seconds DESC;
```

	average_seconds	total_seconds	execution_count	individual_query	object_name	database_name
1	23.949673000	143.698043000	6	SELECT DISTINCT B.REFNO, B.new_acct_name, B.user_brnch...	NMCHK_GETReconciliationRpt	BSADBTW
2	20.769993000	2617.019166000	126	;WITH TempFullData AS (--namecheck SELECT A.QueueID, ...	UserDashboard_GetQueueStatistic	BSADBTW
3	13.485914000	62790.416682000	4656	;WITH CTE_PersonalQueue AS (SELECT A.UID, A.CIF_KE...	DBCHK_GetResultFilter	BSADBTW
4	13.158851000	13.158851000	1	INSERT INTO dbo.Customer_RiskAssessmentAuditLog(Cif...	CIP_UpdateCIPRecord	BSADBTW
5	11.544976000	11.544976000	1	IF EXISTS(SELECT NULL FROM dbo.[CUST_ETL_CHG_LOG...	HRS_GetQuestionaireBusinessCustomerDetails	BSADBTW
6	9.832815000	26440.441516000	2689	INSERT INTO @Temp_FullResults([NameID], [SourceTable], [S...	SWIFT_Detection_CLR	BSADBTW
7	9.734640000	48.673200000	5	;WITH workgroupBranches AS (/*get queue with its branches*/	SAR_CaseAssign_GetWorkgroupStatistics	BSADBTW
8	8.793521000	8.793521000	1	IF NOT EXISTS (SELECT NULL FROM sys.[columns] WHER...	GM_GetAllBusinessEstablishYearDim	BSADBTW
9	8.714766000	43.573831000	5	INSERT INTO dbo.OFAC_DoNotScanList_Result_Groups ([main...	SWIFT_UpdateDoNotScanList_Result	BSADBTW
10	8.493904000	8.493904000	1	IF NOT EXISTS (SELECT NULL FROM sys.[columns] WHERE ...	GM_GetAllAccountStatusDim	BSADBTW
11	8.440992000	8.440992000	1	IF NOT EXISTS (SELECT NULL FROM sys.[columns] WHER...	GM_GetAllBankDealingPeriodDim	BSADBTW
12	8.409527000	8.409527000	1	INSERT INTO dbo.ACCT_XREF_LOG(ACCT_KEY,APPL,CIF...	CIP_UpdateCIPRecord	BSADBTW
13	8.301109000	24.903328000	3	;WITH workgroupBranches AS (/*get queue with its branches*/	SAR_CaseAssign_GetUserStatistics	BSADBTW
14	6.795187000	13.590374000	2	INSERT INTO dbo.CASEDETECTOR_TEMP_EMAIL(REF_KE...	SAR_CaseAssign_ProcessEmail	BSADBTW
15	6.791804000	14432.585397000	2125	INSERT INTO @Temp_New_Acct_Matches_Table([new_acct_ke...	NMCHK_Detection_Blacklist	BSADBTW
16	6.551435000	58.962920000	9	IF EXISTS(SELECT 1 FROM dbo.CUST_ETL_CHG_LOG WIT...	CIP_IsExistETL	BSADBTW

Page Life Expectancy:

衡量記憶體的指標，表示：平均資料可以停留在記憶體的時間(秒)

記憶體耗用量
在 IAN101 的 18/4/2019 10:11:42

Microsoft SQL Server 2014

此報表會提供執行個體中，元件記憶體耗用量的詳細資料，以及由預設的追蹤所記錄，執行個體記憶體使用量變更的歷程記錄資料。

| Memory Grants Outstanding | 0 |
| Memory Grants Pending | 0 |

| Page life expectancy | 107967 |

排名最前面的記憶體耗用元件

- MEMORYCLERK_SQLBUFFERPOOL (21471832 KB)
- OBJECTSTORE_LOCK_MANAGER (139664 KB)
- MEMORYCLERK_SQLCLR (131744 KB)
- CACHESTORE_OBJCP (82224 KB)
- MEMORYCLERK_SOSMEMMANAGER (67700 KB)
- 其他 (211808 KB)
- MEMORYCLERK_XTP (0 KB)

https://www.qa-knowhow.com/?p=939
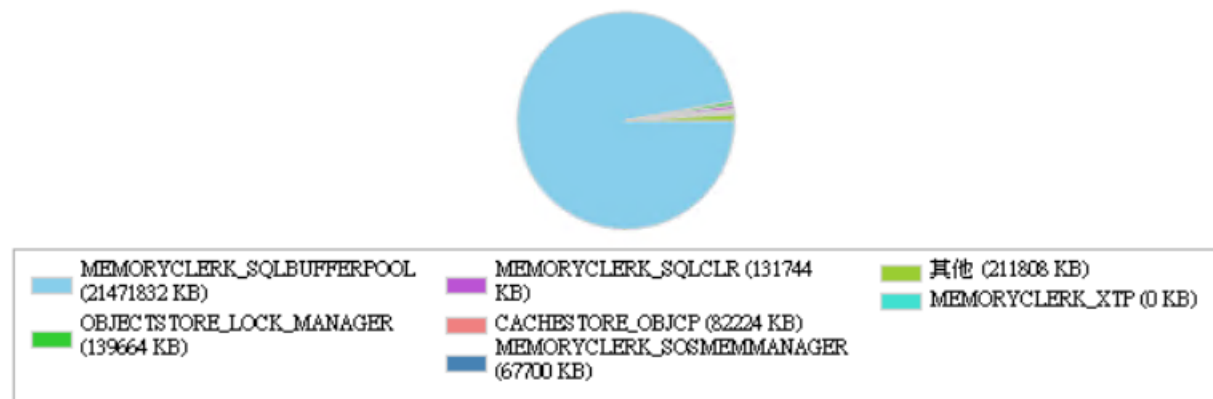
雜湊比對(hash join)：

當系統記憶體不足時，無法取得足夠記憶體把比對資料放入hash table，所以若大量電文近來，memory使用非常大。

平行處理原則的成本臨界值：50

代表cost超過多少就使用平行處理，也就是多顆CPU來執行語法，看執行計畫可以知道用了多少core跑該語法。

平行處理原則的最大程度：8

代表當使用平行處理的時候，最多用幾個core，0代表不限制，若是8代表最多只會有8core

SQL 參數設定

WWW.STI.Com.TW

執行計畫查詢CPU成本數:

估計子樹成本若高於cost threshold for parallelism，則會使用平行運算若cost threshold for parallelism 太低，代表query 都會使用平行運算，則CPU都有可能居高不下
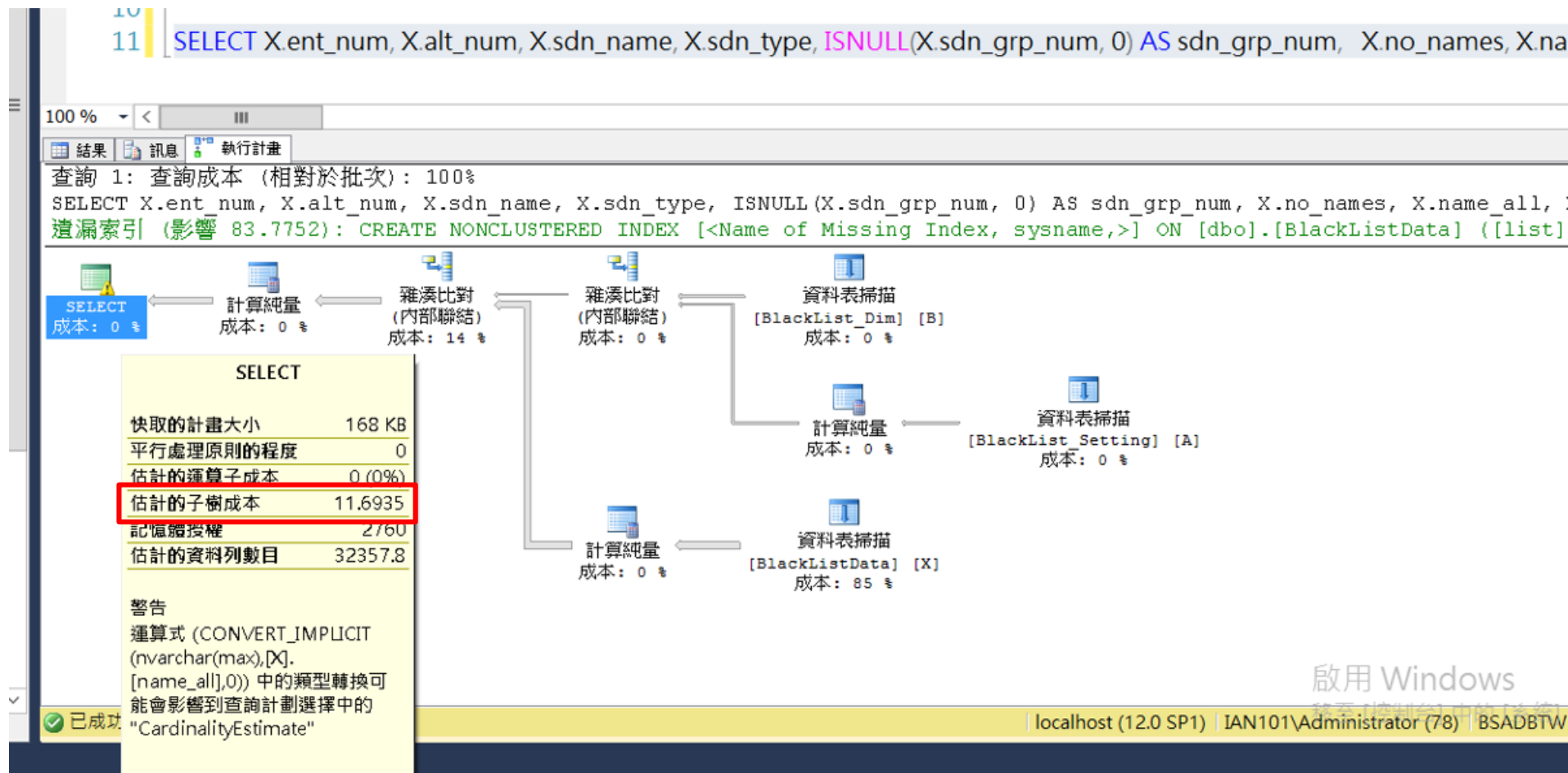
成效數據對比

1. IIS網頁處理速度變快

之前AML系統會在下午(15:00~16:00)會變的相當緩慢,但在資料庫升級後網頁速度明顯變快 (從500秒變為5~8秒)

(3/29)升級前的IIS網頁反應時間 (單位為毫秒) 在網框處可以看到大約在500秒上下
註 Log上時間和台灣時區不同,此為下午4點

BEFORE

AFTER

2. 資料庫記憶體效能明顯上升

我們使用PLE(Page Life Expectation)指標做觀察,它代表資料庫上的資料平均可以在記憶體存活多久的時間(單位為秒)
在升級前資料庫只能使用128GB,PLE明顯不足, 上班時間Cache大約2~3分鐘就被flush out

\\AMLDBPVM\SQLServer:Buffer Manager\Page life expectancy

在升級後我們給資料庫75GB,PLE明顯上升, 上班時間Cache大約可以在記憶體存活7~8個小時

\\AMLDBPVM\SQLServer:Buffer Manager\Page life expectancy

成效數據對比

3.SQL網路傳輸速度變快

https://support.microsoft.com/zh-tw/help/951037/information-about-the-tcp-chimney-offload-receive-side-scaling-and-net
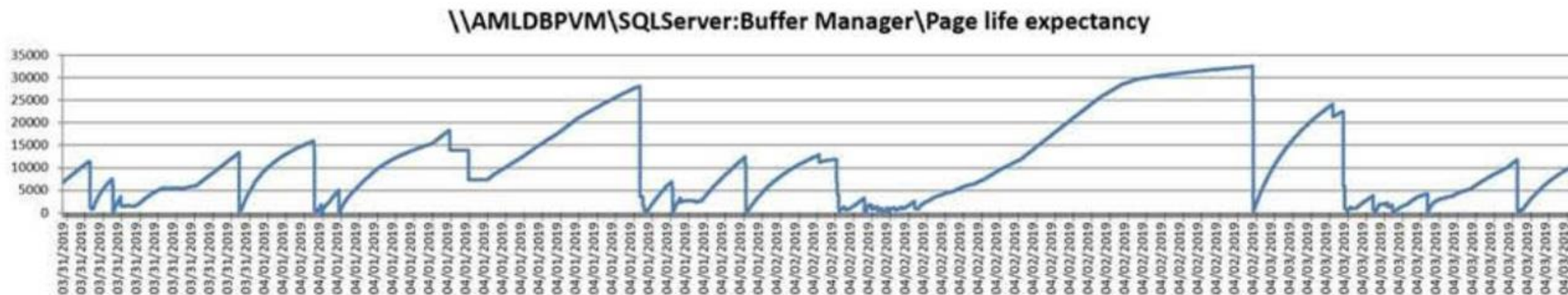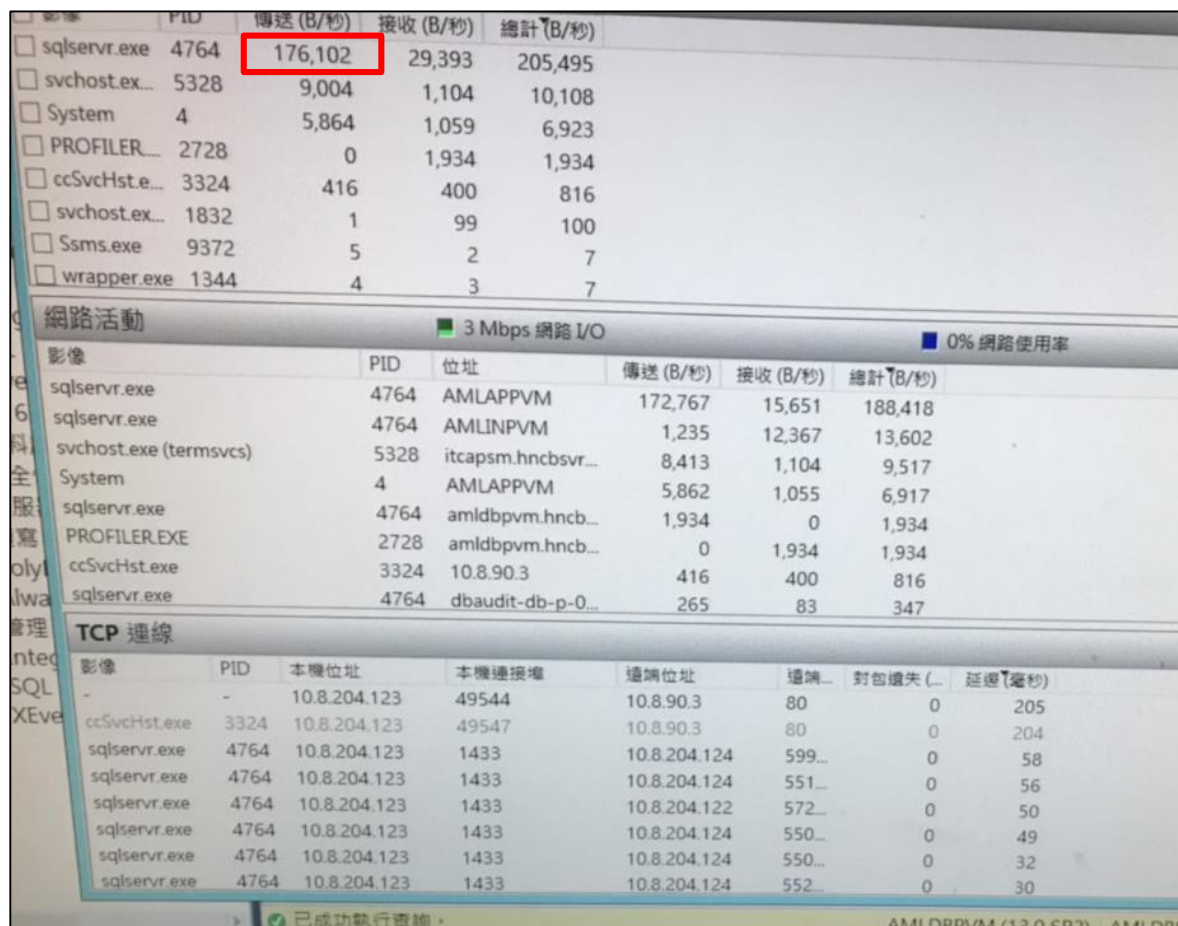
發現關閉TCP Chimney資料庫可以傳輸更多的位元(Byte)給IIS和Integration Service

BEFORE

AFTER