ProActive DBA Warehouse (PDW) 소개

2016. 12

㈜데이터웍스

목 차

- ✓ PDW 소개
- ✓ PDW 도입 시 장점
- ✓ PDW 시스템 아키텍처
- ✓ PDW 기능소개

PDW 소개

✓ ProActive DBA

ProActive DBA는 SAP ASE, IQ, Rep. Server 그리고 MSSQL 및 Oracle에 대한 수행되는 SQL 실행정보를 포함한 모든 DB 운영정보를 최소한의 부하로 실시간 수집하고 관리하여 DB 운영관련 문제 발생시 즉시 대응을 가능하게 하는 통합 DB 모니터링 솔루션입니다.

* 기타 상세한 내용은 별첨의 PDBA 브로슈어를 참고하시기 바랍니다.

✓ PDW 구축 필요성

PDW는 ProActive DBA의 확장옵션으로 복수의 각 서버에서 수집되는 모든 DB 운영정보를 수집하고 통합하여 장기적인 보관을 가능하게 함으로 DB 운영 환경에 대한 효율화 증대 및 DB 접근이력에 대한 감시 및 감사를 가능하게 하는 특수 목적용 DW 시스템입니다.

PDW 도입 시 장점

✓ 인프라 측면에서의 장점

- 복수의 PDBA 서버의 대량의 Output 파일을 통합 DW서버(PDW)로 관리
- PDBA의 Output Flat 파일이 자동으로 PDW로 적재되어 장기간 보관
- Sybase 뿐만 아니라 Oracle, MS-SQL까지 PDW로 통합관리
- PDW의 구축 및 적재, 활용을 위한 모든 기능을 제공

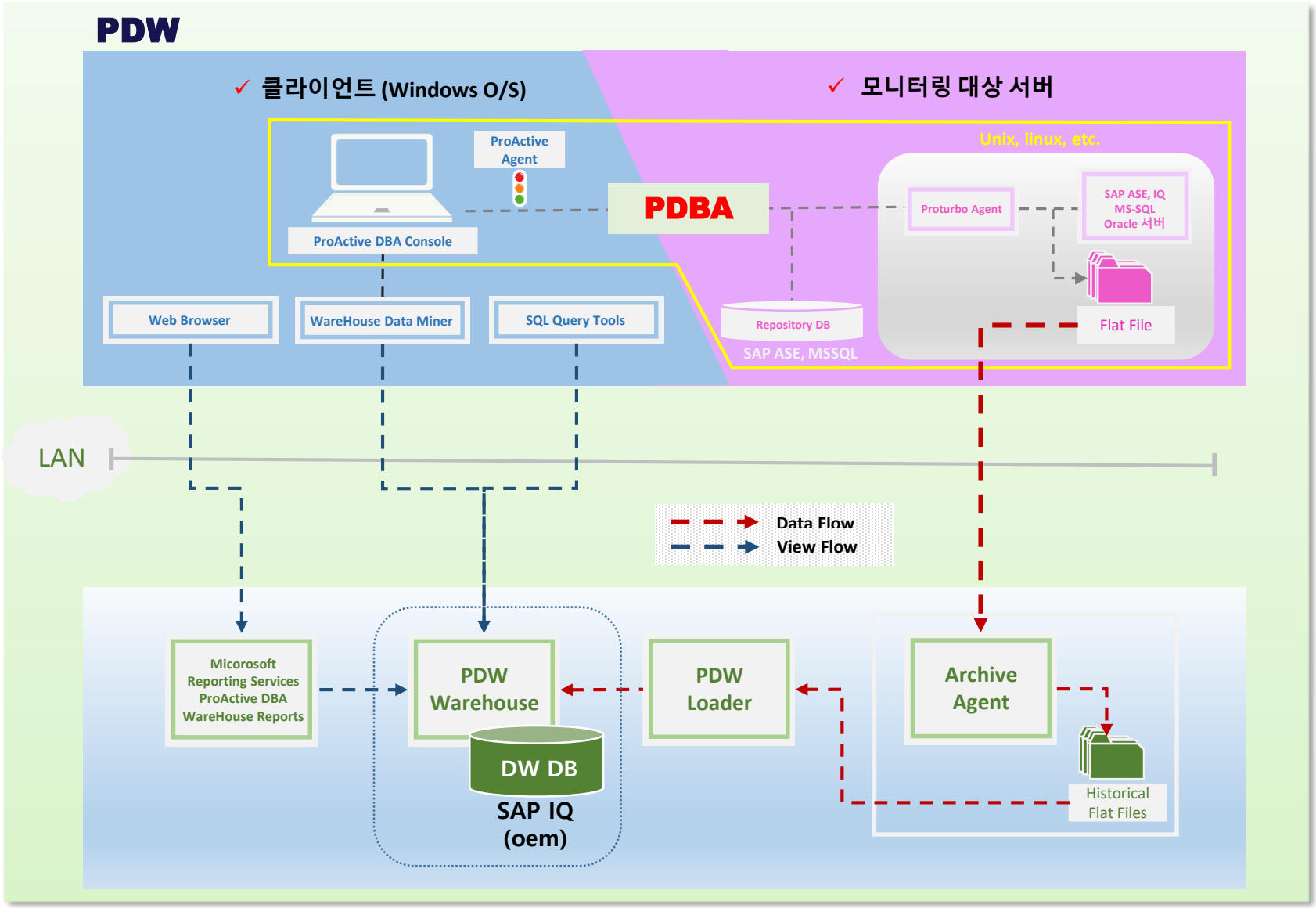
✓ 활용 측면에서의 장점

- DB서버 운용에 대한 추이분석 및 DB 접근이력에 대한 감사 목적
- 수행된 SQL text 원문 등, 상세 실행정보에 대한 장기적인 보유 가능
- 제공되는 Miner 툴 또는 표준 SQL을 사용하여 검색 및 분석 가능

✓ 비용적인 측면에서의 장점

- 기존 PDBA 사용고객인 경우 최소화된 추가비용으로 도입이 가능함
- PDW의 Data 저장을 위하여 저렴한 Sybase IQ(oem)사용이 가능함
- 별도의 구축작업 없이 간단한 설치 및 환경구성으로 작업이 완료됨
- 작업량 분산으로 PDBA Console 부하를 절감할 수 있음

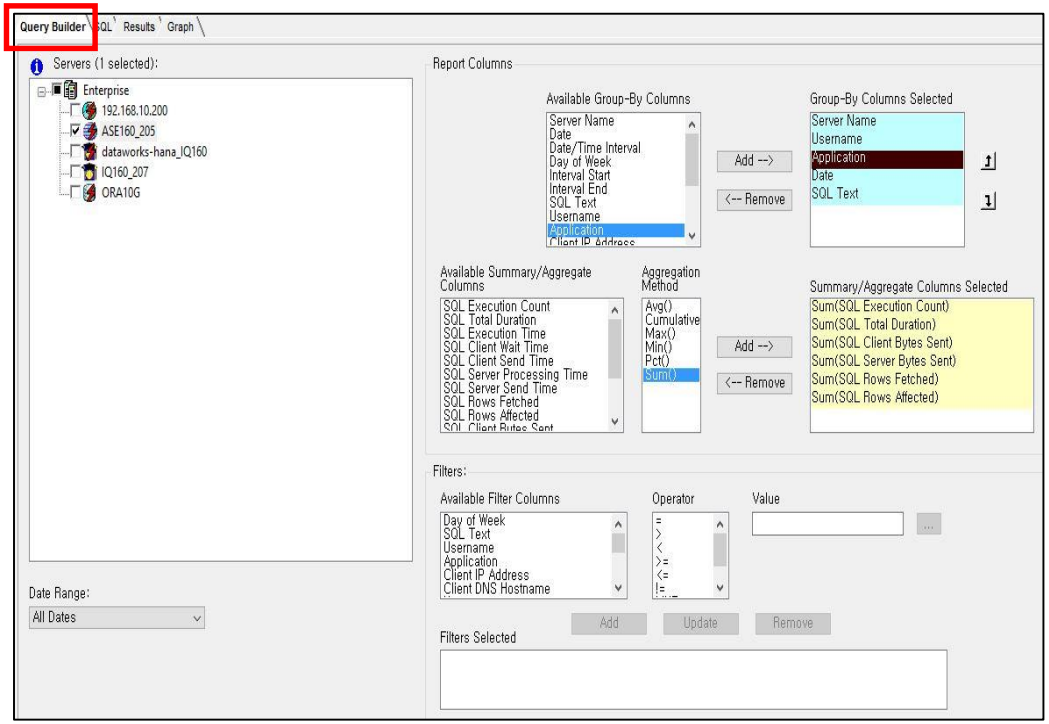
PDW 시스템 아키텍처

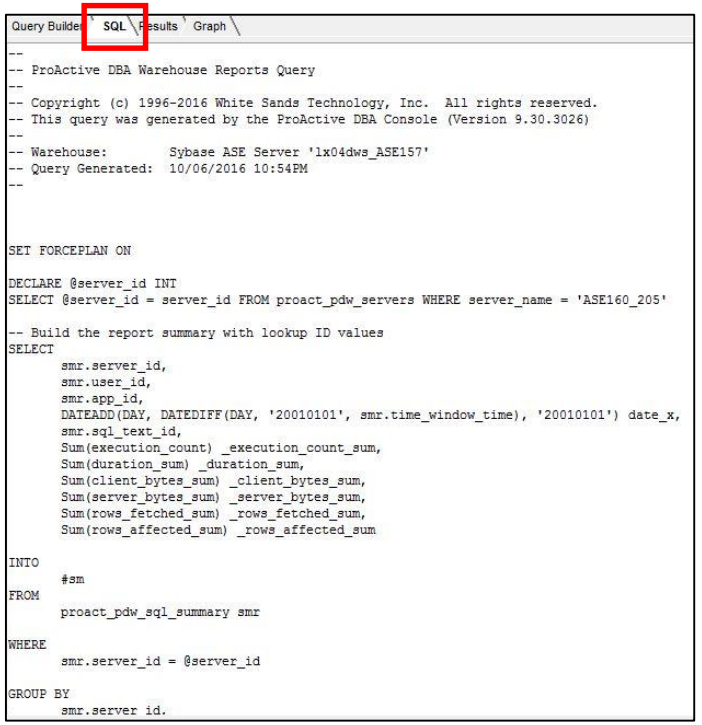

PDW
✓ 클라이언트 (Windows O/S)
✓ 모니터링 대상 서버
ProActive Agent
ProActive DBA Console
PDBA
Unix, linux, etc.
Proturbo Agent
SAP ASE, IQ
MS-SQL
Oracle 서버
Flat File
Repository DB
SAP ASE, MSSQL
Web Browser
WareHouse Data Miner
SQL Query Tools
LAN
Data Flow
View Flow
Micorosoft
Reporting Services
ProActive DBA
WareHouse Reports
PDW
Warehouse
DW DB
SAP IQ
(oem)
PDW
Loader
Archive
Agent
Historical
Flat Files

PDW 기능소개

WareHouse Data Miner 기능 (1/4)

- **PDW에서 제공하는 기본 SQL Builder**
- 정의되어 있는 Report를 선택한 후, 조건을 선택하여 주면 자동으로 SQL을 생성





PDW 기능소개

WareHouse Data Miner 기능 (2/4)

- 수행결과 예1)

- ASE 16.0 - SUMMARY SQL TEXT

Query Builder | SQL | **Results** | Graph

#	Server Name	Username	Application	Date	SQL Text
1	ASE160_205	<Unknown>	<Unknown>	2016-09-06	[Bulk Insert]
2					insert bulk testdb..htest2_bcp with nodescribe
3		sa	PDBA Console	2016-09-06	SELECT COUNT(*) FROM master.dbo.sysconfigures where parent = 0 and config = 0
4					SELECT COUNT(*) FROM master.dbo.sysusages
5					SELECT object_type, object, char_value, int_value FROM master..sysattributes WHERE class = 0 AND attribute = 0 AND object_type = ''
6					select @@kernelmode
7					select @@textsize
8					select @@version
9					select keys1 from sysindexes where id = object_id('') and ((status & 0) != 0 or (status2 & 0) != 0)
10					select low from master.dbo.spt_values where type = '' and number = 0
11					select low, high, name, phyname, status, vdevno, status2 from master.dbo.sysdevices order by vdevno, low
12					select name, dbid, suser_name(suid) owner, status, status2, status3, status4, status5, 0 size, 0 deallocated_pages into #db_list from master.dbo.sysdatabasesupdate
13					select name, value, value3, status from master.dbo.sysconfigures where parent = 0 and config = 0
14					select sum(size) from master.dbo.sysusages where dbid = 0 and ((segmap & 0) != 0)
15					select sum(size) from master.dbo.sysusages where dbid = 0 and ((segmap & 0) != 0) and (segmap != 0)
16					select u.dbid, u.segmap, u.lstart, u.size, u.vstart, curunreservedpgs(u.dbid, u.lstart, u.unreservedpgs), vdevno FROM master.dbo.sysusages u ORDER BY vstart
17					set fmtonly off
18					set showplan off
19					set statistics io off
20					set textsize 0
21				2016-09-08	select @@version
22					select keys1 from sysindexes where id = object_id('') and ((status & 0) != 0 or (status2 & 0) != 0)
23					select proc_role('')
24					select status from master.dbo.sysdatabases where dbid = db_id('')
25				2016-09-09	SELECT COUNT(*) FROM master.dbo.sysconfigures where parent = 0 and config = 0
26					SELECT COUNT(*) FROM master.dbo.sysusages
27					SELECT object_type, object, char_value, int_value FROM master..sysattributes WHERE class = 0 AND attribute = 0 AND object_type = ''
28					select @@kernelmode
29					select @@version
30					select keys1 from sysindexes where id = object_id('') and ((status & 0) != 0 or (status2 & 0) != 0)
31					select low from master.dbo.spt_values where type = '' and number = 0

PDW 기능소개

WareHouse Data Miner 기능 (3/4)

- 수행결과 예2)

- SAP IQ 16.0 – IQ MON

Query Builder | SQL | **Results** | Graph

#	Date/Time Interval	Avg(IQMon Threads Used)	Avg(IQMon Threads Reserved)	Avg(IQMon Threads Free)	Avg(IQMon CPU System (%))	Avg(IQMon CPU User (%))	Avg(IQMon CPU Total (%))
1	2016-09-08 12:30	220.033	54.267	222.700	0.45%	36.53%	36.98%
2	2016-09-08 13:30	220.567	54.133	222.300	0.44%	35.88%	36.33%
3	2016-09-08 14:30	219.840	54.160	223.000	0.45%	37.24%	37.69%
4	2016-09-08 15:30	219.183	54.167	223.650	0.44%	35.28%	35.72%
5	2016-09-08 16:30	217.733	54.433	224.833	0.43%	35.21%	35.65%
6	2016-09-08 17:30	218.017	54.183	224.800	0.46%	34.22%	34.68%
7	2016-09-08 18:30	217.783	54.217	225.000	0.46%	34.17%	34.63%
8	2016-09-08 19:30	218.050	54.150	224.800	0.46%	34.17%	34.62%
9	2016-09-08 20:30	218.200	53.833	224.967	0.47%	35.92%	36.39%
10	2016-09-08 21:30	220.700	51.067	225.233	0.69%	49.85%	50.54%
11	2016-09-08 22:30	218.683	52.117	226.200	0.57%	42.20%	42.76%
12	2016-09-08 23:30	216.767	54.083	226.150	0.41%	35.35%	35.76%
13	2016-09-09 00:30	217.567	53.733	225.700	0.45%	37.20%	37.65%
14	2016-09-09 01:30	217.233	53.800	225.967	0.45%	36.82%	37.27%
15	2016-09-09 02:30	218.733	52.633	225.633	0.55%	43.27%	43.82%
16	2016-09-09 03:30	218.933	52.483	225.583	0.57%	42.50%	43.07%
17	2016-09-09 04:30	219.183	52.550	225.267	0.57%	42.85%	43.42%
18	2016-09-09 05:30	218.017	53.450	225.533	0.49%	38.23%	38.71%
19	2016-09-09 06:30	218.117	52.483	226.400	0.56%	43.39%	43.94%
20	2016-09-09 07:30	223.967	48.133	224.900	0.68%	52.60%	53.28%
21	2016-09-09 08:30	219.233	51.633	226.133	0.55%	41.86%	42.41%
22	2016-09-09 09:30	216.433	54.033	226.533	0.45%	36.00%	36.45%
23	2016-09-09 10:30	217.350	53.567	226.083	0.47%	38.68%	39.15%
24	2016-09-09 11:30	216.633	53.900	226.467	0.41%	33.58%	33.99%

PDW 기능소개

WareHouse Data Miner 기능 (4/4)

- 수행결과 예3)

- Oracle 10g – Database Overview

Query Builder | SQL | Results | Graph

	Date/Time Interval	Interval Start	Interval End	Avg(Database Engine Utilization (%))	Avg(Database Disk I/Os/sec)	Avg(Database Network I/Os/sec)	Avg(Databa	Avg(Database Connections In Use)	Avg(Databas
1	2016-10-06 17:30	2016-10-06 17:00	2016-10-06 18:00	0.22%	0.215	322.596	0.000	7.813	
2	2016-10-06 18:30	2016-10-06 18:00	2016-10-06 19:00	0.16%	0.468	356.752	0.000	7.475	
3	2016-10-06 19:30	2016-10-06 19:00	2016-10-06 20:00	0.08%	0.114	244.736	0.000	8.017	
4	2016-10-06 20:30	2016-10-06 20:00	2016-10-06 21:00	0.20%	0.351	582.243	0.000	12.717	
5	2016-10-06 21:30	2016-10-06 21:00	2016-10-06 22:00	0.20%	0.341	592.256	0.000	12.867	
6	2016-10-06 22:30	2016-10-06 22:00	2016-10-06 23:00	0.82%	2.885	601.384	0.000	12.850	
7	2016-10-06 23:30	2016-10-06 23:00	2016-10-07 00:00	0.12%	0.245	292.239	0.000	12.767	

PDW 기능소개

SQL Tools 활용 (1/4)

- 사용자가 직접 PDW에 SQL Tools을 이용하여 접속하여 원하는 데이터를 검색
- **일별 가장 오랫동안 수행된 SQL 검색**
- 예) ASE160_205 서버에서 하루동안 수행된 쿼리 중 10s 이상 걸린 SQL

```
19
20  DECLARE @server_id int
21          , @server_name VARCHAR(50)
22          , @date VARCHAR(10)
23
24  SELECT @server_name = 'ASE160_205' --ASE160_205,   IQ160_207
25         , @date = '20160906'
26
27  SELECT @server_id = server_id
28  FROM proact_pdw_servers
29  WHERE server_name = @server_name
30
31  SELECT a.server_id
32         , a.source_file_id
33         , a.sql_detail_id
34         , a.login_id
35         , a.start_time
36         , a.end_time
37         , a.server_processing_time
38         , b.login_time
39         , b.host_id
40         , b.spid
41  INTO #test1
42  FROM proact_pdw_sql_detail a
43       , proact_pdw_login_detail b
44  WHERE a.server_id = @server_id
45  AND a.server_processing_time > 10000
46  AND a.start_time BETWEEN @date + ' 00:00:00' AND @date + ' 23:59:59'
47  AND a.server_id = b.server_id
48  AND a.source_file_id = b.source_file_id
49  AND a.login_id = b.login_id
50
51
52  CREATE NONCLUSTERED INDEX IX_test1_server_id ON #test1(server_id, source_file_id, sql_detail_id)
53
54
55  select @server_name AS server_name
56         , b.login_time
57         , b.start_time
58         , b.end_time
59         , b.spid
60         , convert(VARCHAR(255), a.raw_sql_text) AS long_query
61         , b.server_processing_time
62         , a.sql_text_id
63         , a.chunk_no
64         , a.b_continued
65  from proact_pdw_raw_sql_text a, #test1 b
66  WHERE b.server_id = a.server_id
67  AND b.source_file_id = a.source_file_id
68  AND b.sql_detail_id = a.sql_detail_id
```

server_name	login_time	start_time	end_time	spid	long_query	server_processing_
ASE160_205	Sep 6 2016 3:10PM	Sep 6 2016 3:10PM	Sep 6 2016 3:10PM	44	insert into htest3_backselect top 10000 * from htest3	10399
ASE160_205	Sep 6 2016 3:13PM	Sep 6 2016 3:13PM	Sep 6 2016 3:13PM	26	insert into htest3_backselect top 10000 * from htest3	10403
ASE160_205	Sep 6 2016 3:15PM	Sep 6 2016 3:15PM	Sep 6 2016 3:15PM	15	insert into htest3_backselect top 10000 * from htest3	12022
ASE160_205	Sep 6 2016 3:16PM	Sep 6 2016 3:16PM	Sep 6 2016 3:16PM	53	insert into htest3_backselect top 10000 * from htest3	10285
ASE160_205	Sep 6 2016 3:17PM	Sep 6 2016 3:17PM	Sep 6 2016 3:17PM	57	insert into htest3_backselect top 10000 * from htest3	10848
ASE160_205	Sep 6 2016 3:18PM	Sep 6 2016 3:18PM	Sep 6 2016 3:18PM	63	insert into htest3_backselect top 10000 * from htest3	11389
ASE160_205	Sep 6 2016 3:19PM	Sep 6 2016 3:19PM	Sep 6 2016 3:19PM	67	insert into htest3_backselect top 10000 * from htest3	11716
ASE160_205	Sep 6 2016 3:20PM	Sep 6 2016 3:20PM	Sep 6 2016 3:20PM	66	insert into htest3_backselect top 10000 * from htest3	11294
ASE160_205	Sep 6 2016 3:21PM	Sep 6 2016 3:21PM	Sep 6 2016 3:21PM	19	insert into htest3_backselect top 10000 * from htest3	10659
ASE160_205	Sep 6 2016 3:22PM	Sep 6 2016 3:22PM	Sep 6 2016 3:22PM	55	insert into htest3_backselect top 10000 * from htest3	10896
ASE160_205	Sep 6 2016 3:23PM	Sep 6 2016 3:23PM	Sep 6 2016 3:23PM	20	insert into htest3_backselect top 10000 * from htest3	12097
ASE160_205	Sep 6 2016 3:24PM	Sep 6 2016 3:24PM	Sep 6 2016 3:24PM	65	insert into htest3_backselect top 10000 * from htest3	11347
ASE160_205	Sep 6 2016 3:25PM	Sep 6 2016 3:25PM	Sep 6 2016 3:25PM	34	insert into htest3_backselect top 10000 * from htest3	12081
ASE160_205	Sep 6 2016 3:26PM	Sep 6 2016 3:26PM	Sep 6 2016 3:26PM	24	insert into htest3_backselect top 10000 * from htest3	11199
ASE160_205	Sep 6 2016 3:27PM	Sep 6 2016 3:27PM	Sep 6 2016 3:27PM	33	insert into htest3_backselect top 10000 * from htest3	11403
ASE160_205	Sep 6 2016 3:28PM	Sep 6 2016 3:28PM	Sep 6 2016 3:28PM	36	insert into htest3_backselect top 10000 * from htest3	11952
ASE160_205	Sep 6 2016 3:28PM	Sep 6 2016 3:29PM	Sep 6 2016 3:29PM	68	insert bulk testdb..htest1_bcp with nodescribe	12301
ASE160_205	Sep 6 2016 3:28PM	Sep 6 2016 3:29PM	Sep 6 2016 3:29PM	43	insert bulk testdb..htest3_bcp with nodescribe	12079
ASE160_205	Sep 6 2016 3:29PM	Sep 6 2016 3:29PM	Sep 6 2016 3:29PM	49	insert into htest3_backselect top 10000 * from htest3	12105
ASE160_205	Sep 6 2016 3:28PM	Sep 6 2016 3:29PM	Sep 6 2016 3:29PM	28	[Bulk Insert]	10163
ASE160_205	Sep 6 2016 3:30PM	Sep 6 2016 3:30PM	Sep 6 2016 3:30PM	35	insert into htest1_backselect top 10000 * from htest1	15675
ASE160_205	Sep 6 2016 3:30PM	Sep 6 2016 3:30PM	Sep 6 2016 3:30PM	19	insert into htest2_backselect top 10000 * from htest2	17251
ASE160_205	Sep 6 2016 3:28PM	Sep 6 2016 3:30PM	Sep 6 2016 3:30PM	28	insert bulk testdb..htest2_bcp with nodescribe	17741
ASE160_205	Sep 6 2016 3:30PM	Sep 6 2016 3:30PM	Sep 6 2016 3:30PM	22	insert into htest3_backselect top 10000 * from htest3	17647
ASE160_205	Sep 6 2016 3:31PM	Sep 6 2016 3:31PM	Sep 6 2016 3:31PM	15	insert into htest3_backselect top 10000 * from htest3	10306
ASE160_205	Sep 6 2016 3:31PM	Sep 6 2016 3:31PM	Sep 6 2016 3:31PM	65	insert into htest2_backselect top 10000 * from htest2	10178
ASE160_205	Sep 6 2016 3:32PM	Sep 6 2016 3:32PM	Sep 6 2016 3:32PM	17	insert into htest3_backselect top 10000 * from htest3	10448
ASE160_205	Sep 6 2016 3:33PM	Sep 6 2016 3:33PM	Sep 6 2016 3:33PM	67	insert into htest3_backselect top 10000 * from htest3	10478
ASE160_205	Sep 6 2016 3:34PM	Sep 6 2016 3:34PM	Sep 6 2016 3:34PM	28	insert into htest3_backselect top 10000 * from htest3	11221
ASE160_205	Sep 6 2016 3:35PM	Sep 6 2016 3:35PM	Sep 6 2016 3:35PM	25	insert into htest3_backselect top 10000 * from htest3	10427
ASE160_205	Sep 6 2016 3:37PM	Sep 6 2016 3:37PM	Sep 6 2016 3:37PM	28	insert into htest3_backselect top 10000 * from htest3	10191
ASE160_205	Sep 6 2016 3:38PM	Sep 6 2016 3:38PM	Sep 6 2016 3:38PM	12	insert into htest3_backselect top 10000 * from htest3	12411
ASE160_205	Sep 6 2016 3:38PM	Sep 6 2016 3:38PM	Sep 6 2016 3:38PM	26	insert into htest2_backselect top 10000 * from htest2	11803
ASE160_205	Sep 6 2016 3:39PM	Sep 6 2016 3:39PM	Sep 6 2016 3:39PM	57	insert into htest3_backselect top 10000 * from htest3	11021
ASE160_205	Sep 6 2016 3:39PM	Sep 6 2016 3:39PM	Sep 6 2016 3:39PM	56	insert into htest2_backselect top 10000 * from htest2	10507
ASE160_205	Sep 6 2016 3:40PM	Sep 6 2016 3:40PM	Sep 6 2016 3:40PM	63	insert into htest3_backselect top 10000 * from htest3	10023
ASE160_205	Sep 6 2016 3:42PM	Sep 6 2016 3:42PM	Sep 6 2016 3:42PM	24	insert into htest3_backselect top 10000 * from htest3	10846
ASE160_205	Sep 6 2016 3:43PM	Sep 6 2016 3:43PM	Sep 6 2016 3:43PM	67	insert into htest3_backselect top 10000 * from htest3	10949
ASE160_205	Sep 6 2016 3:45PM	Sep 6 2016 3:45PM	Sep 6 2016 3:45PM	20	insert into htest3_backselect top 10000 * from htest3	10694
ASE160_205	Sep 6 2016 3:46PM	Sep 6 2016 3:46PM	Sep 6 2016 3:46PM	41	insert into htest3_backselect top 10000 * from htest3	10352
ASE160_205	Sep 6 2016 3:48PM	Sep 6 2016 3:48PM	Sep 6 2016 3:48PM	30	insert into htest3_backselect top 10000 * from htest3	10919
ASE160_205	Sep 6 2016 3:49PM	Sep 6 2016 3:49PM	Sep 6 2016 3:49PM	43	insert into htest3_backselect top 10000 * from htest3	11252
ASE160_205	Sep 6 2016 3:50PM	Sep 6 2016 3:50PM	Sep 6 2016 3:50PM	1	insert into htest3_backselect top 10000 * from htest3	10625
ASE160_205	Sep 6 2016 3:52PM	Sep 6 2016 3:52PM	Sep 6 2016 3:52PM	68	insert into htest3_backselect top 10000 * from htest3	10363
ASE160_205	Sep 6 2016 3:54PM	Sep 6 2016 3:54PM	Sep 6 2016 3:54PM	21	insert into htest3_backselect top 10000 * from htest3	10085
ASE160_205	Sep 6 2016 3:55PM	Sep 6 2016 3:55PM	Sep 6 2016 3:55PM	37	insert into htest3_backselect top 10000 * from htest3	11211
ASE160_205	Sep 6 2016 3:56PM	Sep 6 2016 3:56PM	Sep 6 2016 3:56PM	26	insert into htest2_backselect top 10000 * from htest2	10655
ASE160_205	Sep 6 2016 3:56PM	Sep 6 2016 3:56PM	Sep 6 2016 3:56PM	41	insert into htest3_backselect top 10000 * from htest3	13236
ASE160_205	Sep 6 2016 3:57PM	Sep 6 2016 3:57PM	Sep 6 2016 3:57PM	35	insert into htest3_backselect top 10000 * from htest3	11537
ASE160_205	Sep 6 2016 3:58PM	Sep 6 2016 3:58PM	Sep 6 2016 3:58PM	35	insert into htest3_backselect top 10000 * from htest3	10876
ASE160_205	Sep 6 2016 3:59PM	Sep 6 2016 3:59PM	Sep 6 2016 3:59PM	45	insert into htest3_backselect top 10000 * from htest3	11607

Rows 41 of 860

PDW 기능소개

SQL Tools 활용 (2/4)

- **일별 특정 사용자가 접근한 테이블 목록**
- 예) User 명이 "pdba"인 사용자가 접근한 테이블 목록

Query | Result 1 | Result 2 | Result 3 | Result 4

```
68      th.table_id,
69      user_id
70
71
72  -- Build final report output
73  SELECT
74      CONVERT(VARCHAR(40), db.database_name) database_name,
75      CONVERT(VARCHAR(40), tbl.owner_name) owner_name,
76      CONVERT(VARCHAR(128), tbl.table_name) table_name,
77      CONVERT(VARCHAR(40), u.user_name) user_name,
78      Sum(selects_sum) selects_sum,
79      Sum(inserts_sum) inserts_sum,
80      Sum(bulk_inserts_sum) bulk_inserts_sum,
81      Sum(updates_sum) updates_sum,
82      Sum(deletes_sum) deletes_sum,
83      Sum(order_bys_sum) order_bys_sum,
84      Sum(group_bys_sum) group_bys_sum,
85      Sum(joins_sum) joins_sum,
86      Sum(havings_sum) havings_sum
87
88  FROM
89      #th th
90      INNER JOIN proact_pdw_table tbl ON  th.server_id = tbl.server_id AND
91          th.table_id = tbl.table_id
92
93      INNER JOIN proact_pdw_databases db ON   tbl.server_id = db.server_id AND
94          tbl.database_id = db.database_id
95
96      INNER JOIN proact_pdw_user u ON     th.server_id = u.server_id AND
97          th.user_id = u.user_id
98  where u.user_name="pdba"     --사용자 : pdba
99
100 GROUP BY
101     CONVERT(VARCHAR(40), db.database_name),
102     CONVERT(VARCHAR(40), tbl.owner_name),
103     CONVERT(VARCHAR(128), tbl.table_name),
104     CONVERT(VARCHAR(40), u.user_name)
105
106 ORDER BY
107     CONVERT(VARCHAR(40), db.database_name),
108     CONVERT(VARCHAR(40), tbl.owner_name),
109     CONVERT(VARCHAR(128), tbl.table_name),
110     CONVERT(VARCHAR(40), u.user_name)
111
112
113 go
114 DROP TABLE #sm
115 go
116 DROP TABLE #th
117 go
118
```

Query | Result 1 | Result 2 | Result 3 | Result 4

database_name	owner_nam	table_name	user_name	selects_sum	inserts_sum	bulk_inserts_sum	updates_sum	deletes_sum	order_l
master	dbo	sysdatabases	pdba	66	0	0	0	0	0
proact_pdw	dbo	proact_pdw_alert_detail	pdba	0	0	42	0	0	0
proact_pdw	dbo	proact_pdw_alert_name	pdba	22	0	4	0	0	0
proact_pdw	dbo	proact_pdw_alert_type	pdba	128	0	0	0	0	0
proact_pdw	dbo	proact_pdw_app	pdba	43	0	12	0	0	0
proact_pdw	dbo	proact_pdw_client_addr	pdba	43	0	10	0	0	0
proact_pdw	dbo	proact_pdw_client_appl_name	pdba	43	0	2	0	0	0
proact_pdw	dbo	proact_pdw_client_host_name	pdba	43	0	2	0	0	0
proact_pdw	dbo	proact_pdw_client_library	pdba	43	0	6	0	0	0
proact_pdw	dbo	proact_pdw_client_name	pdba	43	0	2	0	0	0
proact_pdw	dbo	proact_pdw_column	pdba	60	0	10	0	0	0
proact_pdw	dbo	proact_pdw_column_joins	pdba	20	0	8	0	0	1
proact_pdw	dbo	proact_pdw_columns_hit	pdba	39	0	10	0	0	1
proact_pdw	dbo	proact_pdw_database_users	pdba	20	0	2	0	0	0
proact_pdw	dbo	proact_pdw_databases	pdba	48	0	4	0	0	3
proact_pdw	dbo	proact_pdw_dbm_activity	pdba	42	0	9	0	0	1
proact_pdw	dbo	proact_pdw_dbm_activity_types	pdba	67	0	0	0	0	3
proact_pdw	dbo	proact_pdw_host	pdba	43	0	6	0	0	0
proact_pdw	dbo	proact_pdw_keys	pdba	3524	1762	0	1791	0	0
proact_pdw	dbo	proact_pdw_loader_options	pdba	132	0	0	0	0	0
proact_pdw	dbo	proact_pdw_lock_class	pdba	22	0	2	0	0	0
proact_pdw	dbo	proact_pdw_login_detail	pdba	0	0	41	0	0	0
proact_pdw	dbo	proact_pdw_login_summary	pdba	16	7	0	15	0	0
proact_pdw	dbo	proact_pdw_perf_entry_data	pdba	22	0	0	0	0	0
proact_pdw	dbo	proact_pdw_perf_entry_types	pdba	128	0	0	0	0	0
proact_pdw	dbo	proact_pdw_perf_instance_names	pdba	26	0	10	0	0	4
proact_pdw	dbo	proact_pdw_perf_metrics	pdba	52	8	54	32	0	0
proact_pdw	dbo	proact_pdw_process_cmd	pdba	22	0	12	0	0	0
proact_pdw	dbo	proact_pdw_process_misc_text	pdba	22	0	2	0	0	0
proact_pdw	dbo	proact_pdw_process_status	pdba	22	0	10	0	0	0
proact_pdw	dbo	proact_pdw_process_tran_name	pdba	22	0	10	0	0	0
proact_pdw	dbo	proact_pdw_raw_sql_text	pdba	0	0	66	0	0	0
proact_pdw	dbo	proact_pdw_role	pdba	21	0	2	0	0	0
proact_pdw	dbo	proact_pdw_role_user	pdba	21	0	2	0	0	0
proact_pdw	dbo	proact_pdw_servers	pdba	147	66	0	2	0	1
proact_pdw	dbo	proact_pdw_snapshot_locks	pdba	0	0	6	0	0	0
proact_pdw	dbo	proact_pdw_snapshot_processes	pdba	0	0	42	0	0	0
proact_pdw	dbo	proact_pdw_source	pdba	66	0	16	0	0	0
proact_pdw	dbo	proact_pdw_source_file	pdba	150	64	0	201	0	0
proact_pdw	dbo	proact_pdw_sql_detail	pdba	0	0	42	0	0	0
proact_pdw	dbo	proact_pdw_sql_summary	pdba	41	14	0	16	0	0
proact_pdw	dbo	proact_pdw_sql_text	pdba	62	0	36	0	0	0
proact_pdw	dbo	proact_pdw_sql_text_hash	pdba	49	18	0	0	0	3
proact_pdw	dbo	proact_pdw_table	pdba	47	0	6	9	0	3
proact_pdw	dbo	proact_pdw_table_defn_text	pdba	38	0	2	0	0	9
proact_pdw	dbo	proact_pdw_table_joins	pdba	20	0	10	0	0	1
proact_pdw	dbo	proact_pdw_tables_hit	pdba	42	0	10	0	0	1
proact_pdw	dbo	proact_pdw_time_window	pdba	70	0	16	0	0	0
proact_pdw	dbo	proact_pdw_user	pdba	90	0	4	0	0	6
tempdb	<Unknown>	<TempTable>	pdba	141	19	137	3	0	0

Rows 1 of 50

PDW 기능소개

SQL Tools 활용 (3/4)

- **일별 CPU를 가장 많이 사용한 SQL**
- 예) 하루동안 cpu time을 1초 이상 점유한 SQL

Query | table_usage.sql | Result 1 | Result 2 | Result 3 | Result 4 | Result 5 | Result 6 | Result 7 | Result 8 | Result 9 | Result 10

```sql
raw_sql_text varchar(255)
)
go

DECLARE @server_id INT
SELECT @server_id = server_id FROM proact_pdw_servers WHERE server_name = 'LX04DWS_ASE157'

select server_id, snapshot_time, spid, user_id, cpu_usage_cumul, cpu_usage_new, mem_usage
into #sn
from proact_pdw_snapshot_processes
where cpu_usage_cumul > 10 and snapshot_time between '20161010 00:00:00' and '20161010 23:00:00'
go

declare cur1 cursor for select snapshot_time, spid, user_id, cpu_usage_cumul, cpu_usage_new, mem_usage
from #sn
go

declare @snapshot_time datetime

declare @spid int
declare @user_id int
declare @user_name varchar(255)
declare @cpu_usage_cumul int
declare @cpu_usage_new int
declare @mem_usgae float
declare @start_time datetime
declare @end_time datetime
declare @chunk_no int
declare @raw_sql_text varchar(255)


DECLARE @server_id INT
SELECT @server_id = server_id FROM proact_pdw_servers WHERE server_name = 'LX04DWS_ASE157'




open cur1

fetch cur1 into @snapshot_time, @spid, @user_id, @cpu_usage_cumul, @cpu_usage_new, @mem_usgae

while @@sqlstatus = 0
begin

insert into #result
select
@server_id,
@spid,
c.user_name,
a.start_time,
a.end_time,
```

Result 7 | Result 8 | Result 9 | Result 10 | Result 11 | Result 12 | Result 13 | Result 14 | Result 15 | Result 16 | Result 17

server_id	spid	ser_name		end_time	cpu_usage_cumul	cpu_usage_new	mem_usage	chunk_no	
16	1	sa	2016-10-10 15:27:33.	Oct 10 2016 3:41PM	65	0	23	1	SELECT b.user_name ,l
16	1	sa	2016-10-10 15:27:33.	Oct 10 2016 3:41PM	65	0	23	2	er_id = h.server_idAND b.us
16	29	pdba	2016-10-10 15:11:15.	Oct 10 2016 3:11PM	43	0	25	1	BEGIN TRAN SAVE_SUMM
16	29	pdba	2016-10-10 15:11:15.	Oct 10 2016 3:11PM	43	0	25	2	= 241,last_saved_summary_
16	26	pdba	2016-10-10 15:21:39.	Oct 10 2016 3:21PM	80	0	41	1	BEGIN TRAN SAVE_SUMM
16	26	pdba	2016-10-10 15:21:39.	Oct 10 2016 3:21PM	80	0	41	2	dest.metric_sumsq + src.met
16	26	pdba	2016-10-10 15:21:39.	Oct 10 2016 3:21PM	80	0	41	3	0 then dest.metric_max else
16	26	pdba	2016-10-10 15:21:39.	Oct 10 2016 3:21PM	80	0	41	4	btypeAND dest.instance_nu
16	26	pdba	2016-10-10 15:21:39.	Oct 10 2016 3:21PM	80	0	41	5	2,last_saved_summary_time
16	19	pdba	2016-10-10 15:30:29.	Oct 10 2016 3:30PM	50	0	13	1	BEGIN TRAN SAVE_SUMM
16	19	pdba	2016-10-10 15:30:29.	Oct 10 2016 3:30PM	50	0	13	2	dest.metric_sumsq + src.met
16	19	pdba	2016-10-10 15:30:29.	Oct 10 2016 3:30PM	50	0	13	3	0 then dest.metric_max else
16	19	pdba	2016-10-10 15:30:29.	Oct 10 2016 3:30PM	50	0	13	4	btypeAND dest.instance_nu
16	19	pdba	2016-10-10 15:30:29.	Oct 10 2016 3:30PM	50	0	13	5	2,last_saved_summary_time
16	19	pdba	2016-10-10 15:30:41.	Oct 10 2016 3:30PM	50	0	45	1	BEGIN TRAN SAVE_SUMM
16	19	pdba	2016-10-10 15:30:41.	Oct 10 2016 3:30PM	50	0	45	2	dest.metric_sumsq + src.met
16	19	pdba	2016-10-10 15:30:41.	Oct 10 2016 3:30PM	50	0	45	3	0 then dest.metric_max else
16	19	pdba	2016-10-10 15:30:41.	Oct 10 2016 3:30PM	50	0	45	4	btypeAND dest.instance_nu
16	19	pdba	2016-10-10 15:30:41.	Oct 10 2016 3:30PM	50	0	45	5	= 2,last_saved_summary_tim
16	12	pdba	2016-10-10 15:30:54.	Oct 10 2016 3:31PM	65	0	45	1	BEGIN TRAN SAVE_SUMM
16	12	pdba	2016-10-10 15:30:54.	Oct 10 2016 3:31PM	65	0	45	2	dest.metric_sumsq + src.met
16	12	pdba	2016-10-10 15:30:54.	Oct 10 2016 3:31PM	65	0	45	3	0 then dest.metric_max else
16	12	pdba	2016-10-10 15:30:54.	Oct 10 2016 3:31PM	65	0	45	4	btypeAND dest.instance_nu
16	12	pdba	2016-10-10 15:30:54.	Oct 10 2016 3:31PM	65	0	45	5	2,last_saved_summary_time
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	25	1	SELECT b.user_name ,l
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	25	2	rver_idAND b.user_name = '
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	25	1	SELECT b.user_name ,l
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	25	2	rver_idAND b.user_name = '
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	25	1	SELECT b.user_name ,l
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	25	2	rver_idAND b.user_name = '
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	25	1	SELECT b.user_name ,l
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	25	2	rver_idAND b.user_name = '
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	45	1	SELECT b.user_name ,l
16	23	sa	2016-10-10 15:33:41.	Oct 10 2016 3:36PM	11	0	45	2	rver_idAND b.user_name = '
16	19	pdba	2016-10-10 15:40:53.	Oct 10 2016 3:41PM	64	0	45	1	BEGIN TRAN SAVE_SUMM
16	19	pdba	2016-10-10 15:40:53.	Oct 10 2016 3:41PM	64	0	45	2	dest.metric_sumsq + src.met
16	19	pdba	2016-10-10 15:40:53.	Oct 10 2016 3:41PM	64	0	45	3	0 then dest.metric_max else
16	19	pdba	2016-10-10 15:40:53.	Oct 10 2016 3:41PM	64	0	45	4	btypeAND dest.instance_nu
16	19	pdba	2016-10-10 15:40:53.	Oct 10 2016 3:41PM	64	0	45	5	2,last_saved_summary_time
16	21	pdba	2016-10-10 15:50:53.	Oct 10 2016 3:51PM	61	0	45	1	BEGIN TRAN SAVE_SUMM
16	21	pdba	2016-10-10 15:50:53.	Oct 10 2016 3:51PM	61	0	45	2	dest.metric_sumsq + src.met
16	21	pdba	2016-10-10 15:50:53.	Oct 10 2016 3:51PM	61	0	45	3	0 then dest.metric_max else
16	21	pdba	2016-10-10 15:50:53.	Oct 10 2016 3:51PM	61	0	45	4	btypeAND dest.instance_nu
16	21	pdba	2016-10-10 15:50:53.	Oct 10 2016 3:51PM	61	0	45	5	2,last_saved_summary_time
16	13	sa	2016-10-10 15:54:18.	Oct 10 2016 3:54PM	117	0	45	1	SELECT b.user_name ,l
16	13	sa	2016-10-10 15:54:18.	Oct 10 2016 3:54PM	117	0	45	2	rver_id = h.server_idAND b.u
16	19	pdba	2016-10-10 16:01:06.	Oct 10 2016 4:01PM	68	0	45	1	BEGIN TRAN SAVE_SUMM
16	19	pdba	2016-10-10 16:01:06.	Oct 10 2016 4:01PM	68	0	45	2	dest.metric_sumsq + src.met
16	19	pdba	2016-10-10 16:01:06.	Oct 10 2016 4:01PM	68	0	45	3	0 then dest.metric_max else
16	19	pdba	2016-10-10 16:01:06.	Oct 10 2016 4:01PM	68	0	45	4	btypeAND dest.instance_nu
16	19	pdba	2016-10-10 16:01:06.	Oct 10 2016 4:01PM	68	0	45	5	2,last_saved_summary_time

Rows 1 of 154

PDW 기능소개

SQL Tools 활용 (4/4)

- **시간대별 평균 Thread 사용률**
- 예) ASE160_205 서버의 시간대별 평균 Thread 사용률

Query | Result 1

```
46
47  DECLARE @server_id INT
48  SELECT @server_id = server_id FROM proact_pdw_servers WHERE server_name = 'ASE160_205'
49
50  SET FORCEPLAN ON
51
52  INSERT INTO #pm
53  SELECT
54      smr.server_id,
55      smr.time_window_time,
56      smr.instance_num,
57      smr.instance_name_id,
58      Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 1 THEN smr.metric_sum ELSE NULL END) / SUM
59      Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 2 THEN smr.metric_sum ELSE NULL END) / SUM
60      100.0 * ( ( Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 6 THEN smr.metric_sum ELSE NUL
61      100.0 * ( ( Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 4 THEN smr.metric_sum ELSE NUL
62      100.0 * ( ( Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 5 THEN smr.metric_sum ELSE NUL
63      100.0 * ( Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 7 THEN smr.metric_sum ELSE NULL
64      100.0 * ( Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 8 THEN smr.metric_sum ELSE NULL
65      Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 9 THEN smr.metric_sum ELSE NULL END) / SUM
66      Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 10 THEN smr.metric_sum ELSE NULL END) / SU
67      Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 11 THEN smr.metric_sum ELSE NULL END) / SU
68      Sum(CASE WHEN smr.entry_type = 560 AND smr.entry_subtype = 12 THEN smr.metric_sum ELSE NULL END) / SU
69      100.0 * ( 0.010000 * (Sum(CASE WHEN smr.entry_type = 570 AND smr.entry_subtype = 3 THEN smr.metric_su
70      100.0 * ( 0.010000 * (Sum(CASE WHEN smr.entry_type = 570 AND smr.entry_subtype = 4 THEN smr.metric_su
71      100.0 * ( 0.010000 * (Sum(CASE WHEN smr.entry_type = 570 AND smr.entry_subtype = 9 THEN smr.metric_su
72      100.0 * ( 0.010000 * (Sum(CASE WHEN smr.entry_type = 570 AND smr.entry_subtype = 5 THEN smr.metric_su
73      Sum(CASE WHEN smr.entry_type = 570 AND smr.entry_subtype = 6 THEN smr.metric_sum ELSE NULL END) / SUM
74      Sum(CASE WHEN smr.entry_type = 570 AND smr.entry_subtype = 7 THEN smr.metric_sum ELSE NULL END) / SUM
75
76  FROM
77      proact_pdw_perf_metrics smr
78      INNER JOIN proact_pdw_time_window w ON  smr.server_id = w.server_id AND
79          smr.time_window_id = w.time_window_id
80
81
82  WHERE
83      smr.server_id = @server_id AND
84      (
85          smr.time_window_time >= '20161010 00:00:00.000' AND
86          smr.time_window_time <= '20161011 00:00:00.000'
87      ) AND
88      smr.entry_type IN (560, 570)
89
90  GROUP BY
91      smr.server_id,
92      smr.time_window_time,
93      smr.instance_num,
94      smr.instance_name_id
95
96
```

Query | Result 1

time_window_time	instance_name	tpool_num_threads_avg	tpool_tasks_per_sec_avg	tpool_busy_percent_avg	tpool_idle_percent_avg	tpool_sleep_pe
Oct 10 2016 12:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 12:30AM	syb_default_pool	4	4.9591666666666665	0.14861111111111111	99.851388888888891	0
Oct 10 2016 12:30AM	syb_system_pool	3	0	0.0074074074074074077	99.992592592592587	0
Oct 10 2016 1:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 1:30AM	syb_default_pool	4	4.858888888888889	0.14791666666666667	99.852083333333326	0
Oct 10 2016 1:30AM	syb_system_pool	3	0	0.0055555555555555558	99.994444444444454	0
Oct 10 2016 2:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 2:30AM	syb_default_pool	4	4.8688888888888888	0.14722222222222223	99.852777777777774	0
Oct 10 2016 2:30AM	syb_system_pool	3	0	0.0074074074074074077	99.992592592592587	0
Oct 10 2016 3:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 3:30AM	syb_default_pool	4	4.8594444444444447	0.14166666666666666	99.85833333333332	0
Oct 10 2016 3:30AM	syb_system_pool	3	0	0.0074074074074074077	99.992592592592587	0
Oct 10 2016 4:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 4:30AM	syb_default_pool	4	4.9091666666666667	0.14861111111111111	99.851388888888891	0
Oct 10 2016 4:30AM	syb_system_pool	3	0	0.0074074074074074077	99.992592592592587	0
Oct 10 2016 5:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 5:30AM	syb_default_pool	4	5.0049999999999999	0.14583333333333334	99.854166666666671	0
Oct 10 2016 5:30AM	syb_system_pool	3	0	0.0064814814814814813	99.993518518518528	0
Oct 10 2016 6:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 6:30AM	syb_default_pool	4	4.9836111111111112	0.15069444444444444	99.849305555555546	0
Oct 10 2016 6:30AM	syb_system_pool	3	0	0.0064814814814814813	99.993518518518528	0
Oct 10 2016 7:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 7:30AM	syb_default_pool	4	4.9572222222222226	0.14166666666666666	99.85833333333332	0
Oct 10 2016 7:30AM	syb_system_pool	3	0	0.0074074074074074077	99.992592592592587	0
Oct 10 2016 8:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 8:30AM	syb_default_pool	4	4.9061111111111115	0.15138888888888888	99.848611111111126	0
Oct 10 2016 8:30AM	syb_system_pool	3	0	0.0074074074074074077	99.992592592592587	0
Oct 10 2016 9:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 9:30AM	syb_default_pool	4	4.9227777777777781	0.13194444444444445	99.868055555555557	0
Oct 10 2016 9:30AM	syb_system_pool	3	0	0.0074074074074074077	99.992592592592587	0
Oct 10 2016 10:30AM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 10:30AM	syb_default_pool	4	5.0186111111111114	0.17430555555555557	99.825694444444451	0
Oct 10 2016 10:30AM	syb_system_pool	3	0	0.0083333333333333332	99.991666666666666	0
Oct 10 2016 11:30AM	syb_blocking_pool	4	0	0.00071839080459770114	99.999281609195407	0
Oct 10 2016 11:30AM	syb_default_pool	4	2.7058333333333335	0.29525862068965514	99.704741379310349	0
Oct 10 2016 11:30AM	syb_system_pool	3	0	0.0095785440613026683	99.990421455938687	0
Oct 10 2016 12:30PM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 12:30PM	syb_default_pool	4	5.1516666666666664	0.12777777777777778	99.872222222222234	0
Oct 10 2016 12:30PM	syb_system_pool	3	0	0.0074074074074074077	99.992592592592587	0
Oct 10 2016 1:30PM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 1:30PM	syb_default_pool	4	36.679861111111109	0.28160919540229884	99.718390804597689	0
Oct 10 2016 1:30PM	syb_system_pool	3	0	0.047892720306513405	99.952107279693493	0
Oct 10 2016 2:30PM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 2:30PM	syb_default_pool	4	11.863055555555556	5.7494969818913475	94.25050301810866	0
Oct 10 2016 2:30PM	syb_system_pool	3	0	0.017246335153779824	99.982753664846214	0
Oct 10 2016 3:30PM	syb_blocking_pool	4	0	0	100	0
Oct 10 2016 3:30PM	syb_default_pool	4	27.572777777777777	9.2652777777777775	90.734722222222217	0
Oct 10 2016 3:30PM	syb_system_pool	3	0	0.010185185185185184	99.989814814814821	0

감 사 합 니 다.

데이터웍스 Dataworks