

ENPHAROS TP Guide

ENPHAROS v5.1



Copyright © 2014 Dabomsoft Co.,Ltd. All rights Reserved.

Copyright Notice

Copyright © 2014 Dabomsoft Co., Ltd. All Rights Reserved.

대한민국 서울시 구로구 디지털로 306, 714호(구로동, 대륭포스트타워2차)

Restricted Rights Legend

All Dabomsoft software(ENPHAROS TP®) and documents are protected by copyright laws and the Protection Act of Computer Programs, and international convention. Dabomsoft software and documents are made available under the terms of the Dabomsoft License Agreement and may only be used or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of Dabomsoft Co., Ltd.

이 소프트웨어(ENPHAROS TP®) 사용설명서의 내용과 프로그램은 저작권법, 컴퓨터프로그램보호법 및 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 Dabomsoft Co., Ltd. 와의 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 Dabomsoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

Trademarks

ENPHAROS JAVA®, ENPHAROS TP®, ENPHAROS LINK® and ENPHAROS TRACE® are registered trademark of Dabomsoft Co., Ltd. Other products, titles or 서비스s may be registered trademarks of their respective companies.

ENPHAROS JAVA®, ENPHAROS TP®, ENPHAROS LINK® 와 ENPHAROS TRACE®는 Dabomsoft Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

안내서 정보

안내서 제목: ENPHAROS TP Guide

발행일: 2015-12-14

소프트웨어 버전: ENPHAROS TP v5.1

안내서 버전: v5.1

목 차

1. ENPHAROS TP 소개.....	14
2. Real-Time Monitor.....	17
2.1. Response Time.....	18
2.2. TPS.....	18
2.3. CPU	19
2.4. Queue	19
2.5. Logical Monitored Systems	20
2.5.1. TPSVC Active Requests.....	20
2.5.2. TPSQL Active Requests.....	21
2.5.3. Total Active Request Information	21
2.6. Monitored Systems.....	21
2.6.1. 에이전트 별 상태 및 실시간 정보.....	22
2.6.2. 에이전트 별 큐 정보	23
2.6.3. Queue Analyzer 연계	23
2.7. Throughput	25
2.7.1. Statistics Analyzer 연계.....	25
2.8. Average Response Time	26
2.8.1. TOP-N Analyzer 연계	26
2.9. Active Request	26
2.10. Request Performance.....	28
2.10.1. Request Analyzer 연계.....	29
3. Request Analyzer.....	32
3.1. Occurrences 영역	32
3.2. Call-Tree 영역	34
3.2.1. 서비스 타입.....	35
3.2.2. Method 타입.....	36
3.2.3. SQL타입	37
3.2.4. Link타입	38
3.3. CallTree summary pop-up.....	40
4. Statistics Analyzer.....	42
4.1. TPSVC 통계분석	42
4.2. TPSQL 통계분석	46
5. Top-N Analyzer.....	50
5.1. Avg.Response Time.....	51
5.2. Sum Response Time.....	52
5.3. Execution Count.....	53
5.4. Issue Count	54

6. Issue Analyzer	57
6.1. TP Issue 종류	57
6.1.1. TP AP Error	58
6.1.2. TP Service Error	58
6.1.3. TP Sql Error	59
6.1.4. TP Server Abnormally Down Error	60
6.1.5. General Error	61
7. Event Analyzer	64
7.1. Event Info	65
7.1.1. Event List	65
7.1.2. Request List & Request Performance	66
7.2. Event Analyzer를 통한 상세분석	66
8. Compare Analyzer	70
8.1. Compare 리스트	70
8.2. Trend	71
9. Caller Callee Analyzer	73
10. Queue Analyzer	75
10.1. Tuxedo Queue List	76
10.2. Tmax Queue List	77
11. Resource Monitor	79
12. Resource Statistics Analyzer	81
13. System Log Analyzer	83
13.1. Tuxedo ULOG	83
13.2. Tmax slog	84

그림 목차

[그림] 2-1 Real-Time Monitor 화면구성	17
[그림] 2-2 에이전트별 Response Time 차트.....	18
[그림] 2-3 에이전트별 TPS 차트.....	19
[그림] 2-4 에이전트별 CPU 차트.....	19
[그림] 2-5 에이전트별 Queue 차트.....	20
[그림] 2-6 logical Monitored Systems.....	20
[그림] 2-7 TPSVC Active Requests	21
[그림] 2-8 TPSQL Active Requests	21
[그림] 2-9 Total Active Request Information.....	21
[그림] 2-10 Non-Grouping 형식으로 에이전트 표현 화면	22
[그림] 2-11 에이전트 정상상태와 이벤트 발생상태	22
[그림] 2-12 Monitored Systems 차트의 TP Monitor Queue 개수 표현	23
[그림] 2-13 Monitored Systems 차트와 Queue Analyzer 화면과의 연계	24
[그림] 2-14 Throughput	25
[그림] 2-15 Throughput 과 Statistics Analyzer 연계	25
[그림] 2-16 Average Response Time	26
[그림] 2-17 Average Response Time	26
[그림] 2-18 Active Request 차트	27
[그림] 2-19 Active Requests 분석 화면	27
[그림] 2-20 Active Requests 과 Request Analyzer 연계	29
[그림] 3-1 Request Analyzer 화면 구성	32
[그림] 3-2 Occurrences에서 Misc 정보	33
[그림] 3-3 Request 성능 분포 차트	34
[그림] 3-4 Call-Tree 분석화면	34
[그림] 3-5 Call-Tree 서비스 데이터 및 메타정보	36
[그림] 3-6 Call-Tree Method 데이터	36
[그림] 3-7 Call-Tree SQL 데이터 및 메타정보	37
[그림] 3-8 Instant Query 정보	38
[그림] 3-9 SQL PLAN 정보	38
[그림] 3-10 Link 정보	39
[그림] 3-11 전체 Call-Tree 정보	40
[그림] 3-12 CallTree Summary pop-up 화면	40
[그림] 4-1 Statistics Analyzer 화면 구성	42
[그림] 4-2 Statistics Analyzer의 TPSVC 차트	43
[그림] 4-3 XY축 막대-값 표시	44

[그림] 4-4 TPSVC의 Elapsed Time에 대한 Request Top-N 차트	44
[그림] 4-5 TPSVC의 Elapsed Time Request Top-N 차트와 Request Analyzer 연계	45
[그림] 4-6 Statistics Analyzer의 TPSQL 차트.....	46
[그림] 4-7 TPSQL의 Elapsed Time Request Top-N 차트와 Request Analyzer 연계	47
[그림] 5-1 Top-N Analyzer Chart-View 화면 구성	50
[그림] 5-2 Top-N Analyzer List-View 화면 구성	51
[그림] 5-3 Avg.Response Time 과 Request Analyzer 연계에 의한 Top-N 분석.....	52
[그림] 5-4 Sum Response Time 과 Request Analyzer 연계에 의한 Top-N 분석.....	53
[그림] 5-5 Execution Count 과 Request Analyzer 연계에 의한 Top-N 분석	54
[그림] 5-6 Issue Count 과 Request Analyzer 연계에 의한 Top-N 분석	55
[그림] 6-1 Issue Analyzer 화면 구성	57
[그림] 6-2 TP AP Error	58
[그림] 6-3 TP Service Error.....	59
[그림] 6-4 TP Sql Error	59
[그림] 6-5 TP Server Abnormal Down Error.....	60
[그림] 6-6 Core Dump 발생과 Request Analyzer 연계.....	60
[그림] 6-7 General Error	61
[그림] 7-1 Event Analyzer 화면 구성	64
[그림] 7-2 이벤트 리스트	65
[그림] 7-3 Request List & Request Performance	66
[그림] 7-4 Request Analyzer 화면 연계	67
[그림] 8-1 Compare Analyzer 화면 구성.....	70
[그림] 8-2 Compare 리스트	71
[그림] 8-3 Compare 트랜드 차트	71
[그림] 9-1 Caller Callee Analyzer 화면 구성	73
[그림] 10-1 TP Queue Analyzer 화면 구성	75
[그림] 10-2 TP Queue Analyzer Trend	76
[그림] 10-3 Tuxedo Queue List.....	76
[그림] 10-4 Tmax Queue List.....	77
[그림] 11-1 Resource Monitor 화면구성	79
[그림] 12-1 Tmax Queue List	81
[그림] 13-1 System Log Analyzer	83
[그림] 13-2 Tuxedo ULOG	84
[그림] 13-3 Tmax slog.....	84

표 목차

[표] 2-1 서비스 처리 시간에 따른 경고색.....	20
[표] 2-2 Total Active Request Information 항목 정보.....	21
[표] 2-3 수행시간에 따른 경고색.....	22
[표] 2-4 에이전트 정보.....	23
[표] 2-5 Instant Queue 분석 화면 항목 정보.....	24
[표] 2-6 Request Performance 색상.....	28
[표] 6-1 Issue 종류.....	58
[표] 7-1 이벤트 유형별 상세 정보.....	65
[표] 7-2 이벤트 리스트 항목 정보.....	65

안내서에 대하여

안내서의 대상

본 안내서는 ENPHAROS TP[®]를 사용하여 TP Monitor 어플리케이션 성능을 모니터링 하고자 하는 사용자를 위한 안내서이다. ENPHAROS TP[®]를 이용해서 어플리케이션 성능 모니터링을 위한 기본 개념과 모니터링을 위해 필요한 설정 정보에 대해서 설명한다. 또한 ENPHAROS TP[®]를 사용해서 어플리케이션 성능 모니터링의 다양한 구축 환경에 대한 설명과 해당 환경에서 적용을 위한 방법을 제시한다.

안내서의 전제 조건

본 안내서는 ENPHAROS TP[®] 시스템에 대한 전반적인 이해와 ENPHAROS TP[®] 시스템이 제공하는 각종 기능 및 특성에 대한 습득을 위한 기본서이다.

본 안내서를 원활하게 이해하기 위해서는 다음과 같은 사항을 미리 알고 있어야 한다.

- TP Monitor에 대한 기본적인 이해
- TP 서비스 프로그래밍의 이해

관련 안내서

본 안내서는 ENPHAROS TP의 기본적인 운영 방법을 설명하며, ENPHAROS TP 설치 및 사용자 안내서는 다루지 않는다.

- "ENPHAROS TP Installation Guide"
ENPHAROS TP의 설치 및 삭제방법에 관하여 기술한다.
- "ENPHAROS TP Administration Guide"
ENPHAROS TP의 에이전트에 대한 환경설정 파일과 시스템 관리방법에 대해 기술한다.
- TP Monitor에 대한 안내서는 TP Monitor 사에서 제공하는 안내서를 참고한다.

안내서 구성

ENPHAROS TP®(이하 ENPHAROS TP라고 함) User Guide는 총 13개의 장으로 구성되었다. UI에 대한 설명보다는 주로 ENPHAROS TP 이용에 관한 내용에 초점을 맞추어 작성이 되었다. UI에 대한 설명 및 설정에 대한 내용은 별도의 메뉴얼인 <*ENPharos 5.0 TP 화면 안내서*>를 참고하기를 바란다.

각 장의 주요 내용은 다음과 같다.

- 제1장: ENPHAROS TP 소개

ENPHAROS TP에 대한 기본적인 개요에 대하여 설명한다.

- 제2장: Real-Time Monitor

ENPHAROS TP의 메인 화면인 실시간 모니터링에 대하여 기술한다.

- 제3장: Request Analyzer

Request에 대한 상세 Call-Tree 정보에 대한 내용을 기술한다..

- 제4장: Statistics Analyzer

통계 정보를 분석하는 방법과 내용에 대하여 기술한다.

- 제5장: Top-N Analyzer

Top-N을 통한 분석 방법과 내용에 대하여 기술한다.

- 제6장: Issue Analyzer

Issue 통계 정보를 통한 분석 방법과 내용에 대하여 기술한다.

- 제7장: Event Analyzer

Event 통계 정보를 통한 분석 방법과 내용에 대하여 기술한다.

- 제8장: Compare Analyzer

과거 특정일 간의 업무처리 데이터를 비교 분석하는 내용에 대하여 기술한다.

- 제9장: Caller Callee Analyzer

서비스 호출 관계에서 호출하는 서비스와 호출 받는 서비스의 관계를 분석하는 방법과 내용에 대하여 기술한다.

- 제10장: Queue Analyzer

최근 1시간 동안 TP Monitor의 Queue 화면에 대하여 기술한다.

- 제11장: Resource Monitor

TP Monitor 계정으로 기동된 모든 프로세스에 대한 시스템자원 사용률에 대한 분석 방법 및 내용을 기술한다.

- 제12장: Resource Statistics

최근 1시간 동안 TP Monitor 계정으로 기동된 모든 프로세스에 대하여 시스템자원 사용률에 대한 통계차트 분석 방법 및 내용을 기술한다.

- 제13장: System Log Analyzer

TP Monitor에서 제공되는 시스템로그에 대한 분석 방법 및 내용을 기술한다.

안내서 규약

표 기	의미
<AaBbCc123>	프로그램 소스 코드의 파일명
<Ctrl>+C	Ctrl과 C를 동시에 누름
[Button]	GUI의 버튼 또는 메뉴 이름
진하게	강조
" "(따옴표)	다른 관련 안내서 또는 안내서 내의 다른 장 및 절 언급
'입력항목'	화면 UI에서 입력 항목에 대한 설명
하이퍼링크	메일계정, 웹 사이트
>	메뉴의 진행 순서
+----	하위 디렉터리 또는 파일 있음
----	하위 디렉터리 또는 파일 없음
참고	참고 또는 주의사항
[그림] 1-1	그림 이름
[표] 1-1	표 이름
AaBbCc123	명령어, 명령어 수행 후 화면에 출력된 결과물, 예제코드
[]	옵션 인수 값
	선택 인수 값
「」	디렉터리 경로 또는 해당 파일 경로

시스템 환경

종류	요구사항	
Software	Web Browser	Internet Explorer 7.0 이상, Mozilla Firefox 4.0 이상 Google Chrome, Safari 5.0 이상, Opera 11 이상
	Viewer	Adobe Flash Player 11.9 이상
Hardware	NIC	100Mbps
	RAM	1024MB 이상 메모리 공간
	Display	최소 1024X768 해상도, 24bit 색상 이상

01

ENPHAROS TP 소개

1. ENPHAROS TP 소개

ENPHAROS TP®(이하 ENPHAROS TP라고 함)는 TP-monitor middleware(이하 TP Monitor라고 함) 기반에서 실행되는 서비스¹가 호출되는 시점부터 종료될 때까지 실시간으로 프로세싱 플로우(Processing flow)에 대한 정보와 TP Monitor의 상태정보, 시스템 자원 사용에 대한 정보, 거래 부하율 등 업무를 처리함에 있어 시스템, TP Monitor, 어플리케이션간 발생할 수 있는 각종 성능저하 이슈들을 쉽게 찾고 분석하여 능동적으로 해결 할 수 있도록 정보를 제공하는 시스템이다.

ENPHAROS TP 의 주요 모니터링 영역은 다음과 같다.

- 분산된 각 시스템 리소스(CPU, Memory)에 대한 모니터링
- 분산된 각 TP Monitor 엔진에 대한 모니터링
- 분산된 각 TP Monitor 로그파일에 대한 모니터링
- 분산된 각 TP Monitor 큐 정보에 대한 모니터링
- 분산된 각 TP Monitor 서비스(어플리케이션)에 대한 모니터링
- 전체 거래 부하율 및 처리율에 대한 모니터링

각 영역에 대한 실시간 모니터링을 통하여 여러 시스템에 걸쳐 분산 처리되고 있는 각 시스템에 대한 상태, 각 TP Monitor에 대한 상태, 개별 TP서비스의 상태, 각 거래에 대한 업무처리 상태 등을 가독성이 높은 여러 실시간 차트들을 제공하여 하나의 화면에서 종합적으로 관리/운영을 할 수 있도록 하고 있다.

실시간 모니터링 중 한 영역에서 문제가 발생 시 이를 감지하여 사용자에게 문제가 발생하였음을 이벤트 알림을 통하여 즉각 알려준다. 이러한 이벤트는 사용자 설정에 의하여 정의되며 업무적인 조건을 포함하여 설정할 수 있다. 어떠한 시스템의 어떠한 영역에서 어떠한 내용에 의해 발생하였는지를 즉각적으로 알려주기 때문에 장애발생에 대한 빨 빠른 대처가 가능하다.

ENPHAROS TP는 실시간 모니터링 기능뿐만 아니라 현재까지 처리 되었던 거래들에 대하여 통계화 된 데이터로 산출하여 시간대 별, 업무 별, 처리결과 별, 에러코드 별, 채널코드 별 비교 분석이 가능하도록 차트화 하여 제공해 준다. 이들 통계 데이터는 1분, 10분, 1시간, 하루 단위의 통계 데이터로 구성되어 있으며, 시스템, TP Monitor, TP서비스 등의 튜닝 및 장애예방을 위한 사후분석을 위해 사용이 된다.

ENPHAROS TP는 TP서비스 프로그램의 원본 소스수정 없이도 프로그램의 가시성(visibility)을 부여하여 어플리케이션에 대한 디버깅 작업 또는 원본소스 확인 없이도 수행된 함수 및 SQL과 그에 대한 처리시간 (Elapsed time)과 처리결과 확인이 가능하다. 이를 통해서 이슈가 발생하거나, 발생 할 소지가 큰 함수와 SQL 문이 무엇인지 파악할 수 있고, 특히 SQL에 대해서는 바인드 변수에 대한 값과 패치(fetch)건수, 소스라인, 수행결과 코드에 대한 SQL의 종합적인 수행정보들을 제공하고 있다.

¹ 서비스: TP-Monitor 기반의 어플리케이션

ENPHAROS TP에서 하나의 거래는 속성을 나타내는 메타정보들과 오류정보, 그리고 Call-Tree 정보로 구성되어 있다. 메타정보는 주로 업무적인 내용의 코드들의 집합으로써 거래코드, 업무코드, GUID, 세션정보, 프로그램정보, 업무 파라미터 정보 등으로 구성되어 있고 거래의 유일성을 나타내는 정보이다. 하나의 거래가 프로그램 내에서 처리되는 과정은 수 많은 함수들과 SQL 문들의 수행으로 이루어진다. 이러한 함수들의 수행정보들과 SQL 문에 대한 수행정보들을 호출된 순서대로 트리(tree)구조로 구성하여 거래의 처리 과정(processing flow)을 가시성(visibility) 있게 부여하여 나타내는 데이터를 Call-Tree 데이터라고 한다.

ENPHAROS TP가 제공하는 주요 분석화면들 중에 Real-Time Monitoring 화면, Request Analyzer 화면, Statistics Analyzer 화면, Top-N Analyzer 화면, Issue Analyzer 화면, Event Analyzer 화면 등은 모두 과거의 특정 시점에 대한 Call-Tree 데이터를 가지고 있으며, Request Analyzer 화면을 제외한 이들 화면 모두는 Request Analyzer 화면과의 연계가 가능하다. Request Analyzer 화면 연계를 통하여 특정 시점에 수행 되었던 거래에 대한 Call-Tree 데이터와 관련 메타 데이터와 오류 정보들 등 이들 여러 정보들을 이용하여 소스레벨까지의 상세 분석이 가능하다.

02

Real-Time Monitor

- 2.1 Response Time
 - 2.2 TPS
 - 2.3 CPU
 - 2.4 Queue
 - 2.5 Logical Monitored Systems
 - 2.6 Monitored Systems
 - 2.7 Throughput
 - 2.8 Average Response Time
 - 2.10 Active Requests
 - 2.11 Request Performance
-

2. Real-Time Monitor

Real-Time Monitor(이하 실시간 모니터링 이라 함)는 대시보드 형태로 ENPHAROS TP가 제공하는 여러 기능들 중에 실시간성 모니터링을 위한 가시성 있는 차트들만을 뽑아 사용자관점에서 모든 대상 영역에 대하여 종합적으로 모니터링이 가능하도록 만든 메인 화면이다.

주요 모니터링 대상으로는 다음과 같다.

- CPU, Memory 등 시스템 리소스 사용률
- 각종 이벤트
- 거래 부하율(초당, 10분, 24H)
- 거래 처리율 (TP서비스관점/SQL 관점)
- TP Monitor Queue 적재 상황
- 평균 응답시간
- Active Request
- Request Performance



[그림] 2-1 Real-Time Monitor 화면구성

- ① Response Time : Agent별 최근 5분간 평균응답시간 분포를 나타내는 차트
- ② TPS : Agent별 최근 5분간 초당 처리건수 분포를 나타내는 차트
- ③ CPU : Agent별 최근 5분간 CPU 사용률의 분포를 나타내는 차트
- ④ Queue : Agent별 최근 5분간 TP-monitor의 큐(Queue)의 상황을 나타내는 차트
- ⑤ Logical Monitored Systems : 논리적인 부하량 및 처리량을 표현

- ⑥ Monitored Systems : 모니터링 대상 시스템의 주요 상황을 표현
- ⑦ Throughput : 하루 동안의 발생된 거래량을 10분 단위로 추이차트로 표현된 추이차트
- ⑧ Average Response Time : 하루 동안 처리된 거래에 대한 응답시간을 10분 단위로 추이차트로 표현된 추이차트
- ⑨ Active Requests : 현재 처리되고 있는 서비스의 상황을 Grid 로 표현
- ⑩ Request Performance : 최근 5분간 처리 완료된 거래에 대하여 수행시간과 수행결과에 따라 표현되는 점차트

Logical Monitored Systems, Monitored Systems, Active Requests는 실시간으로 데이터 발생시 표현이 되고, Throughput 차트, Average Response Time 차트는 10분 간격으로 표현을 하고 있다. 나머지 Request Time 차트, TPS 차트, CPU 차트, Queue 차트, Request Performance등의 차트는 5초 간격으로 표현하고 데이터 갱신이 이루어진다.

2.1. Response Time

Response Time 차트는 처리 서비스에 대해서 5초 간격으로 최근 5분간의 에이전트 별 응답시간을 그래프로 표현한다. 에이전트 별로 그래프 색상은 다르게 표현되며, 마우스 포인터를 그래프의 선 상위에 위치하면 에이전트 명, 시간, 해당 시간 응답시간, 응답률, 총 응답시간을 확인할 수 있다.

Response Time 차트는 Layer별로 구분하여 볼 수 있다. ENPHAROS TP UI의 Layer에는 서비스의 처리를 나타내는 TPSVC와 서비스 내에서 처리하는 SQL인 TPSQL이 있다.



[그림] 2-2 에이전트별 Response Time 차트

2.2. TPS

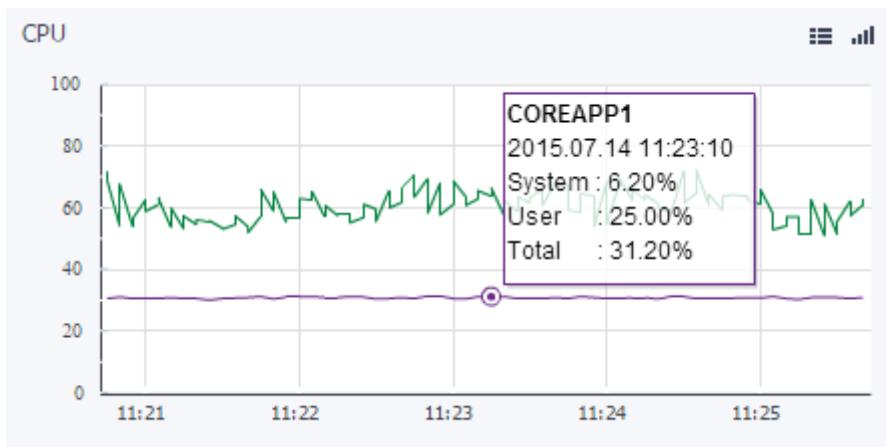
TPS (Throughput Per Second) 차트는 에이전트별로 5초 간격으로 최근 5분간의 초당 처리 건수를 그래프로 표현한다. TPSVC/TPSQL Layer별로 구분하여 볼 수 있으며, 에이전트 별로 그래프의 색상이 다르게 표현되며, 차트확대가 가능하다. 마우스 포인터를 그래프의 선 상위에 위치하면 에이전트명, 시간, 해당 시간 처리율, 처리 건수를 확인할 수 있다.



[그림] 2-3 에이전트별 TPS 차트

2.3. CPU

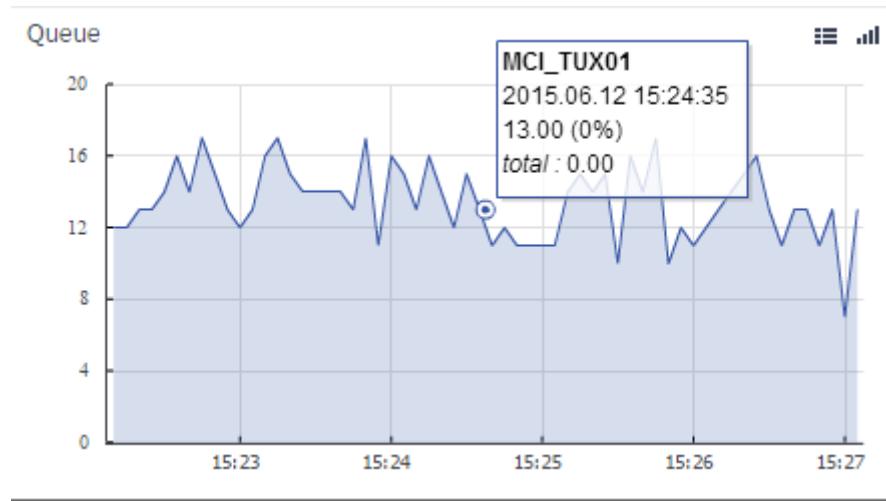
CPU 차트는 에이전트 별로 5초 간격으로 최근 5분간 CPU 사용률을 그래프로 표현한다. 에이전트 별로 그래프의 색상이 다르게 표현되며, 마우스 포인터를 그래프의 선 상위에 위치하면 에이전트 명, 시간, CPU사용률을 확인할 수 있다.



[그림] 2-4 에이전트별 CPU 차트

2.4. Queue

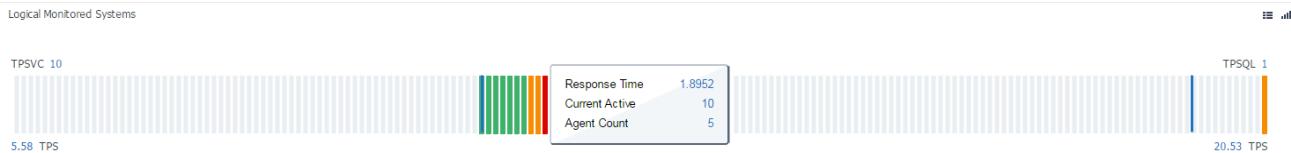
Queue 차트는 TP Monitor의 서버 프로세스에서 처리하지 못하고 대기하고 있는 건수를 표현하는 차트로, 5초 간격으로 최근 5분간의 모니터링 대상 TP 에이전트의 큐 건수를 그래프로 표현한다. 그래프를 통하여 최근 5분간 큐에 얼마나 쌓여있었는지를 확인할 수 있으며, 이를 바탕으로 TP Monitor 서버 프로세스를 추가적으로 실행해야 하는지 알 수 있다. Response Time 그래프와 마찬가지로 에이전트별 색상 범례 확인과 차트 선택, 차트 확대가 가능하다. 마우스 포인터를 그래프의 선 위에 위치하면 에이전트 종류, 시간, 큐 건수 확인이 가능하다.



[그림] 2-5 에이전트별 Queue 차트

2.5. Logical Monitored Systems

Logical Monitored Systems 차트는 TP Monitor 시스템 전체에 대한 처리 응답시간과 요청 건수, 큐 건수를 종합적으로 나타내어 시스템의 현황을 직관적으로 표현하는 차트로 Layer별 요청 건수와 TPS(초당 처리율)로 나타낸다. TPS의 수치가 증가 할수록 스피드메타의 애니메이션 속도도 증가하게 되므로 스피드메타의 애니메이션이 매우 빠르게 동작한다면 이는 많은 부하를 받아 처리가 되고 있음을 나타낸다.



[그림] 2-6 logical Monitored Systems

현재 실행중인 서비스의 수행시간에 따라 초록색, 주황색, 빨간색으로 표현이 된다.

각 색상의 의미는 [표] 2-1과 같다.

경고색	설명
초록색 (Normal)	요청수행시간 <= 5초
주황색 (Warning)	5초 < 요청수행시간 <= 10초
빨강색 (Fatal)	10초 < 요청수행시간

[표] 2-1 서비스 처리 시간에 따른 경고색

2.5.1. TPSVC Active Requests

TPSVC Active Requests는 Logical Monitored Systems 차트의 왼쪽 부분으로 Active 상태의 TP서비스 개수와 TPS(초당 처리 건수) 정보를 표현한다.

아래 [그림] 2-7에서 현재 처리중인 Active 상태의 TP서비스 20개가 수행 중에 있고, 초당 11.68건의

Request를 처리하고 있다.

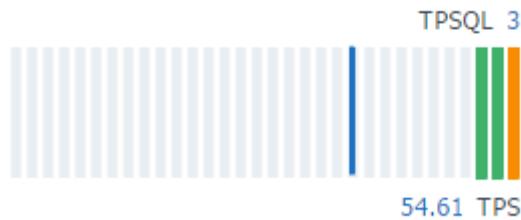


[그림] 2-7 TPSVC Active Requests

2.5.2. TPSQL Active Requests

TPSQL Active Requests는 Logical Monitored Systems 차트의 오른쪽 부분으로 Active 상태의 TP서비스가 처리하고 있는 SQL 건수와 TPS(초당 처리 건수) 정보를 표현한다.

아래 [그림] 2-8에서 현재 처리중인 SQL 건수는 3건이고, 초당 54.61건의 SQL을 처리하고 있다.



[그림] 2-8 TPSQL Active Requests

2.5.3. Total Active Request Information

Total Active Request Information은 평균 응답시간, 현재 처리중인 Request 건수, 에이전트 수를 나타낸다.

Response Time	1.9224
Current Active	19
Agent Count	3

[그림] 2-9 Total Active Request Information

목록	설명
Response Time	현재 처리중인 Active Request에 대한 평균 응답시간
Current Active	현재 처리중인 Active Request 건수
Agent Count	TP 에이전트의 수

[표] 2-2 Total Active Request Information 항목 정보

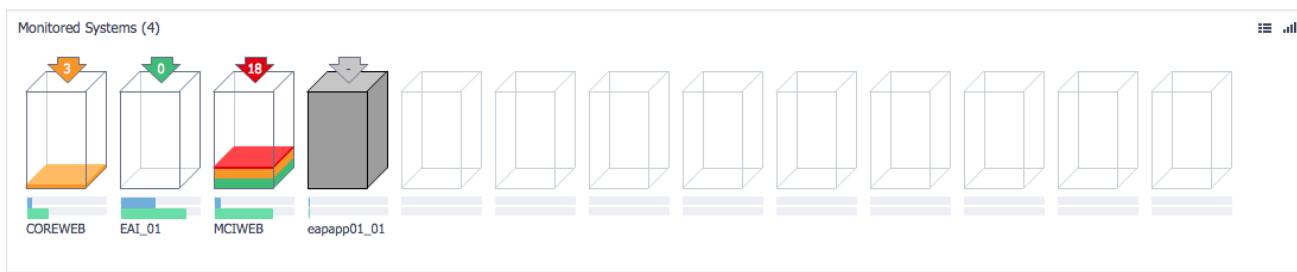
2.6. Monitored Systems

Monitored Systems은 등록되어 있는 각 에이전트들의 상태를 육면체 형태의 그래픽으로 표현하고, 에이

전트 별로 발생되는 각각의 거래(Request)들을 처리하는 시간의 경과에 따라 색상을 부여하여 육면체의 밑면을 채우는 방식으로 시스템 별 에이전트들의 현재 상황을 사용자에게 가시성을 높여 제공하고 있다. 이는 하나의 뷰로 각 시스템 자원에 대한 상황, 거래(Request) 처리 상황, 각종 이벤트 상황 등 전체 시스템에 대한 모니터링이 가능하도록 제공된 강력한 기능들 중에 하나이다.

다음은 육면체의 에이전트 모듈이 표현하고 있는 정보들이다.

- 에이전트의 up/down
- Active Request 개수
- Active Request에 대한 처리시간 경과
- TP Monitor의 Queue 대기 개수
- 시스템의 CPU/MEM 사용률



[그림] 2-10 Non-Grouping 형식으로 에이전트 표현 화면

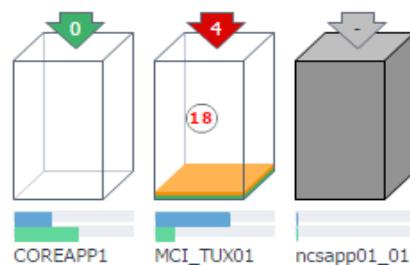
차트에서 표현하는 색상의 의미는 [표] 2-3과 같다.

경고색	내용
초록색 (Normal)	요청수행시간 <= 5초
주황색 (Warning)	5초 < 요청수행시간 <= 10초
빨강색 (Fatal)	10초 < 요청수행시간

[표] 2-3 수행시간에 따른 경고색

2.6.1. 에이전트 별 상태 및 실시간 정보

Monitored Systems에서 표현되는 각각의 에이전트 모듈들에 대해 발생한 이벤트의 종류, 현재 처리중인 Active Request 건수, Queue 상태정보, CPU 정보(%), 메모리 정보(%), 에이전트 명 등을 알 수 있다.



[그림] 2-11 에이전트 정상상태와 이벤트 발생상태

[그림] 2-11에서 다음과 같은 정보들을 확인 할 수 있다.

에이전트명	Active Request 처리 경과시간	Active Request 건수	Queue 상태정보	에이전트 상태정보
COREAPP1	5초 이하 (Normal:초록색)	0	0	Active
MCI_TUX01	10초 이상 (Fetal:빨강색)	4	18	Active
ncsapp01_01	-	-	-	down

[표] 2-4 에이전트 정보

참고

에이전트가 다운되면 담당자는 즉시 에이전트 상태나 시스템 상태를 확인해야 한다. 에이전트가 비정상 종료된 경우에는 기술지원 담당자에게 전달할 수 있도록 Core 파일이나 로그 파일을 복사하고 에이전트를 재 기동하고, 네트워크나 시스템 장애인 경우에는 해당 장애 상황에 따라서 조치를 취해야 한다

2.6.2. 에이전트 별 큐 정보

TP Monitor 기반의 TP서비스가 많은 부하 또는 성능저하로 인해 요청 받은 데이터를 처리하지 못하게 되면 해당 입력 데이터는 TP Monitor Queue에 차례로 저장이 되고 해당 TP서비스는 이를 순차적으로 처리를 하게 된다. 이때 TP Monitor Queue에 쌓인 데이터의 개수를 실시간으로 파악하여 에이전트 모듈의 중앙에 빨간 글씨로 표현한다.

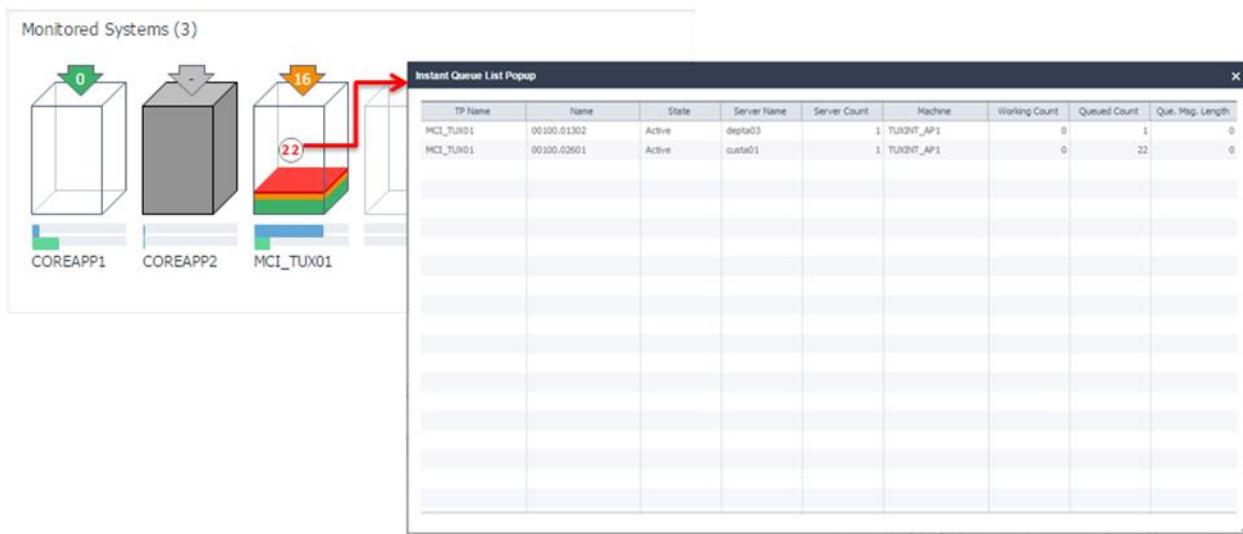
[그림] 2-12의 내용으로 “MCI_TUX01 에이전트의 TP Monitor에서 현재 총 12개의 처리되지 못한 데이터가 있다.”라고 파악할 수 있다.



[그림] 2-12 Monitored Systems 차트의 TP Monitor Queue 개수 표현

2.6.3. Queue Analyzer 연계

에이전트 모듈 안의 큐 건수를 클릭하면 ‘Queue Analyzer’ 화면과 연계되어 현재 어떤 시스템의 어떤 서비스가 관계된 큐에서 데이터가 몇 건씩 적재가 되어 있는지 상세한 정보를 확인할 수 있다.



[그림] 2-13 Monitored Systems 차트와 Queue Analyzer 화면과의 연계

항목	내용
TP Name	해당 시점의 TP Alias 명
Name	해당 시점의 TP 서버 프로세스 명 (Tuxedo인 경우 큐명)
State	해당 시점의 TP 서버 상태 (Tuxedo인 경우 큐 상태) RDY: Ready / NRDY: Non-Ready / RUN: 실행 중 / Unregistered: 등록되지 않은 상태 BLK: 해당 서비스의 모든 프로세스가 Suspend 됨 / PBLK: 해당 서비스의 일부 프로세스가 Suspend 됨 / UNADV: 해당 서비스가 속한 모든 서버 프로세스가 unadvertise 된 상태 PUNADV: 해당 서비스가 속한 일부 서버 프로세스가 unadvertise 된 상태
Server Name	해당 시점의 Server 명
Server Count	해당 시점의 Server 건수
Machine	TP 애이전트가 설치된 시스템 명
Working Count	해당 시점의 Working 건수
Queued Count	해당 시점의 Queue 건수
Que. Msg Length	해당 시점의 Queue 메시지 길이

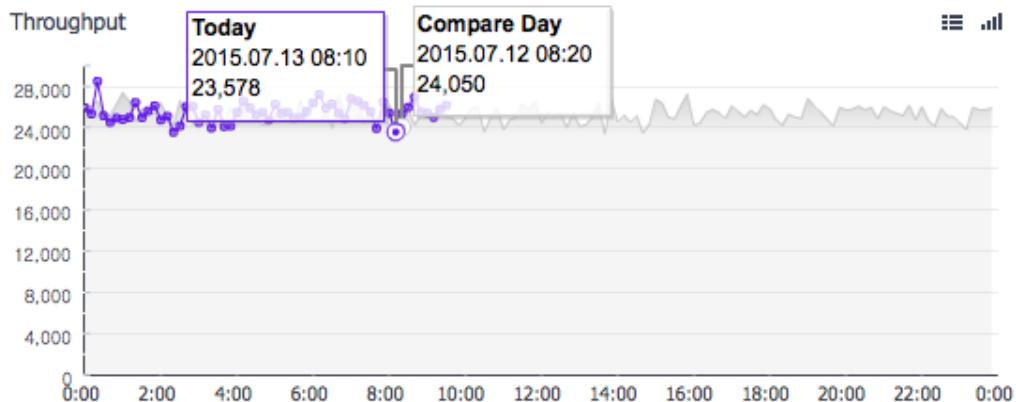
[표] 2-5 Instant Queue 분석 화면 항목 정보

이를 통해 실시간으로 TP Monitor에 대한 Queue 상황 및 내용을 쉽게 파악 할 수 있어 TP서비스 구성에 대한 튜닝이 가능하며 나아가서는 TP서비스 개수 문제로 인한 성능저하나 장애를 사전에 예방 할 수 있는 근거를 제공 받을 수 있다.

TP Monitor Queue에 관련 된 자세한 내용은 <본 문서 "10.Queue Analyzer"> 를 참고한다.

2.7. Throughput

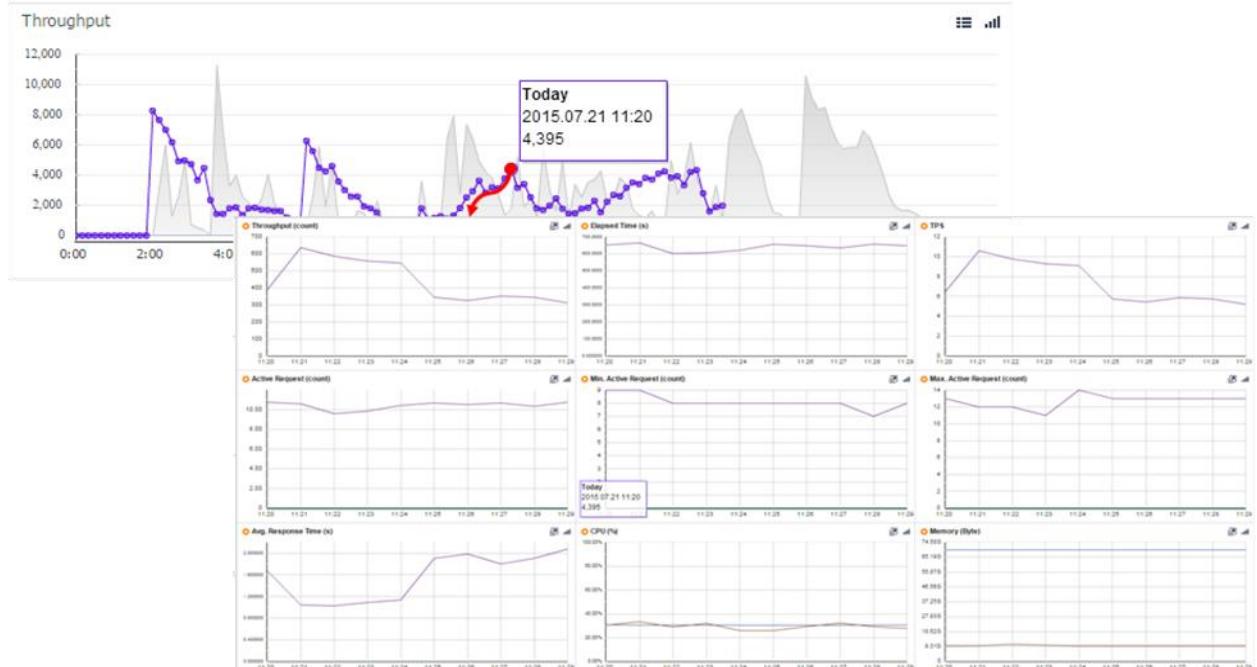
Throughput은 서비스에 대한 거래(Request) 처리량을 표현하는 차트이다. 10분 간격으로 기록되며, 당일과 Compare Day의 거래량을 10분 단위로 비교하여 보여준다. 마우스 포인터를 그래프의 선 위에 위치하면 데이터 포인트가 나타나며 오늘 해당 시간대에 요청 처리한 총 처리량과 24시간 전의 총 처리량을 데이터 팁으로 표현한다.



[그림] 2-14 Throughput

2.7.1. Statistics Analyzer 연계

Throughput에 표시되고 있는 차트의 특정 시점을 클릭하면 Statistics Analyzer 화면으로 전환되어 클릭한 지점에 해당하는 시간 범위 동안의 데이터 추이를 확인할 수 있다.

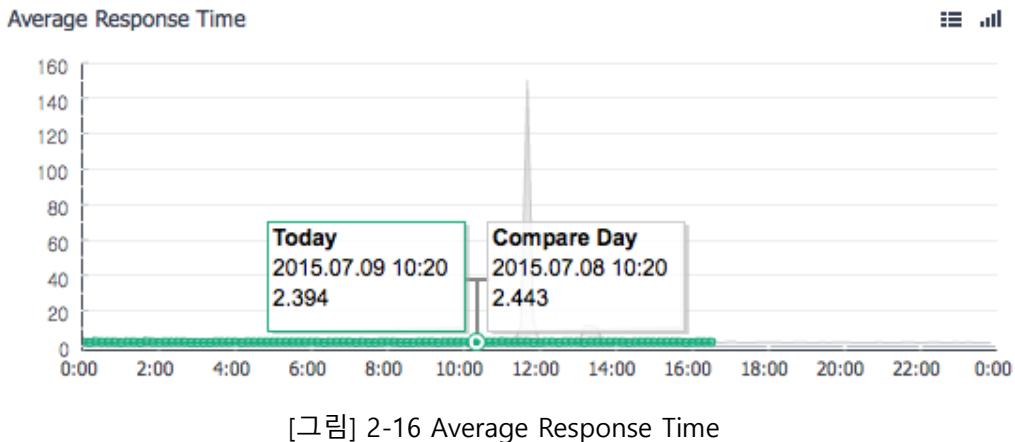


[그림] 2-15 Throughput 과 Statistics Analyzer 연계

Statistics Analyzer에 관련된 자세한 내용은 <본 문서 "4.Statistics Analyzer">를 참고한다.

2.8. Average Response Time

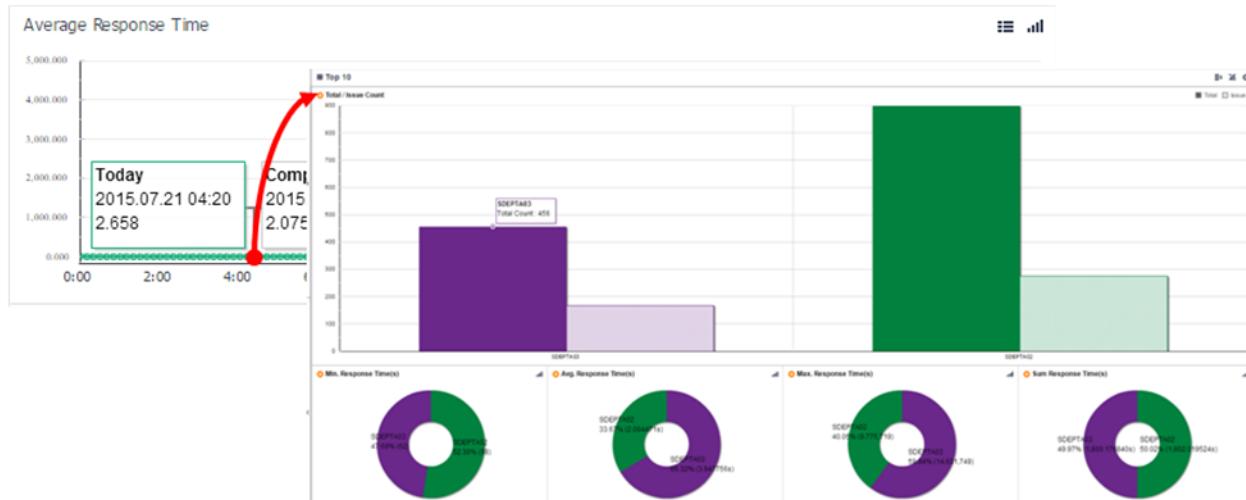
Average Response Time은 TPSVC Layer의 Throughput(처리량)에 대한 평균 응답 시간을 제공한다. 10분 간격으로 기록되며, 오늘 발생한 평균 응답시간을 10분 단위로 종합하여 보여주기 때문에, 각 시간대의 평균 응답시간을 비교할 수 있다. 마우스 포인터를 그래프의 선 위에 위치하면 해당 시간 평균 응답시간과 24시간 전의 평균 응답시간을 확인할 수 있다.



[그림] 2-16 Average Response Time

2.8.1. TOP-N Analyzer 연계

Average Response Time에 표현되고 있는 차트를 클릭하면 TOP-N Analyzer 화면으로 전환되어 클릭한 지점에 해당하는 시간 범위 동안 처리된 거래를 정렬하여 확인할 수 있다



[그림] 2-17 Average Response Time

Top-N에 관련 된 자세한 내용은 <본 문서 “5.Top-N Analyzer”> 를 참고한다.

2.9. Active Request

Active Request 차트는 각 시스템에서 현재 처리중인 거래(Request)에 대한 정보를 제공한다. 현재 처리중

인 거래(Request)를 5초단위로 표현하며, TPSVC/TPSQL로 구분하여 볼 수 있다.
Elapsed(s) 항목에 표현하는 텍스트 색상은 Logical Monitored Systems 차트에서 선택한 색상이 적용되며 초록색은 5초 미만이고, 주황색은 5초 이상 10초 미만, 빨간색은 10초 이상을 표현한다. 기본적으로 TPSVC/TPSQL Layer 별 확인이 가능하다.

Active Requests (4)		
Request	Agent	Elapsed(s)
SCUSTA03	MCI_TUX01	4.648035
SCUSTA01	MCI_TUX01	7.351759
SDEPTA03	MCI_TUX01	3.197080
SDEPTA03	MCI_TUX01	3.468668

Active Requests - TPSVC

[그림] 2-18 Active Request 차트

대상 Request를 더블클릭 시 [그림] 2-18 과 같이 관련 된 Active Request에 대한 상세정보가 pop-up 되어 출력된다.

Active Requests (4)		
Request	Agent	Elapsed(s)
SCUSTA03	MCI_TUX01	4.648035
SCUSTA01	MCI_TUX01	7.351759
SDEPTA03		
SDEPTA03		

Active Requests

Request	Request Time	Agent	Elapsed(s)
SCUSTA03	2015.06.18 18:07:41.253	MCI_TUX01	5.246671
SCUSTA03	2015.06.18 18:07:41.399	MCI_TUX01	5.099804

Auto Refresh

Signature

Info	No.	Signature	Layer	Type	Processing Time(s)	Elapsed(s)	Gap
65	SQL	0.11% : fetch	TPSQL	TPSQL	0.006024	0.006024	0.000000
66	SQL	0.00% : closecursor	TPSQL	TPSQL	0.000002	0.000002	0.000000
67	SQL	0.00% : biz_account_check	TPFUNC	FUNC	0.000011	0.012199	0.000000
68	(Method)	0.05% : check_input_account	TPFUNC	FUNC	0.002697	0.002697	0.000000
69	SQL	0.06% : declare : select A.ACCOUNT_NO from ACCOUNT_INFO	TPSQL	TPSQL	0.003442	0.003442	0.000000
70	SQL	0.11% : fetch	TPSQL	TPSQL	0.005990	0.005990	0.000000
71	SQL	0.00% : closecursor	TPSQL	TPSQL	0.000002	0.000002	0.000000
72	SQL	0.00% : biz_service_check	TPFUNC	FUNC	0.000055	0.000057	0.000000
73	(Method)	0.00% : check_input_data	TPFUNC	FUNC	0.000002	0.000002	0.000000
74	SQL	0.00% : insert : insert into PFM_APLOGTX (GLOBAL_ID,REQST_RESP...	TPSQL	TPSQL	0.000450	0.000450	2.000000

[그림] 2-19 Active Requests 분석 화면

pop-up 창으로 활성화 된 Active Request 상세정보 화면은 두 개의 영역으로 구분이 되어하는데, 상단부분은 현재 처리중인 Active Request의 목록이 출력이 되고, 하단부분에는 상단에서 선택된 Active request

에 해당하는 거래의 call-tree 구조가 출력이 된다. call-tree 데이터의 가장 아래 부분(마지막 부분)이 현재 서비스 내에서 처리가 진행 중인 부분이다.

이를 통해 현재 처리되고 있는 거래(Request)들 중 예상외로 처리시간이 오래 걸리는 거래(Request)를 대상으로 Active Request 분석화면을 통하여 상세 처리과정을 확인하면 어느 부분에서 처리가 늦어지고 있고, 그 이유가 무엇인지 알 수가 있다. 만약 SQL 처리 부분에서 늦어지고 있다면 SQL의 메타정보를 통해 SQL Full-Text 와 Bind Value 를 통해 직접 SQL을 날려서 확인 해 볼 수도 있고, ENPHAROS TP 에서 제공하는 SQL PLAN 기능을 사용하여 문제가 되는 SQL에 대한 실행계획을 확인 할 수 있다.

Active Request 상세정보 call-tree 영역에 대한 설명은 <본 문서 “3.2 Call-Tree 영역”> 의 내용을 참조한다

2.10. Request Performance

Request Performance 차트는 최근 5분 동안 발생한 거래(Request)들에 대하여 오류종류에 따라 각기 다른 색상의 점으로 표시한 처리결과를 수행시간대의 좌표에 위치하게 하여 만든 분포도로써 사용자가 보다 쉽게 TP서비스들에 대한 성능을 판단할 수 있도록 가시성을 극대화한 실시간 차트이다.

Request Performance 차트의 구성은 현재 시간의 흐름을 표현하는 X축과 거래가 처리된 시간을 표현하는 Y축의 좌표로 구성되어 있고, 실시간으로 처리완료 된 거래(Request)를 수신하는 대로 바로 표현해 주고 있다 현재부터 과거의 5분 동안의 처리결과에 대한 내용만을 표시해 주기 때문에 이전에 이루어졌던 거래(Request)에 대한 내용은 Request Analyzer 화면을 통해서 특정 시간대의 조회를 통해 관련 데이터를 확인 할 수 있다.

참고

Request Performance 차트는 여러 메뉴에서 같은 이름의 같은 형식으로 사용되고 있지만, 실시간 모니터링 메뉴에서만이 실시간성으로 처리되고 있고, 나머지 메뉴에서는 과거 데이터에 대한 이미지 표현을 목적으로 사용되고 있다.

Request Performance에 표현되는 점의 색상은 오류종류를 나타낸다. [표] 2-7은 기본설정 된 오류종류에 따른 색상표이다. 색상에 대한 변경은 <ENPharos 5.0 TP 화면 안내서 의 “3.10 Request Performance”>을 참고한다.

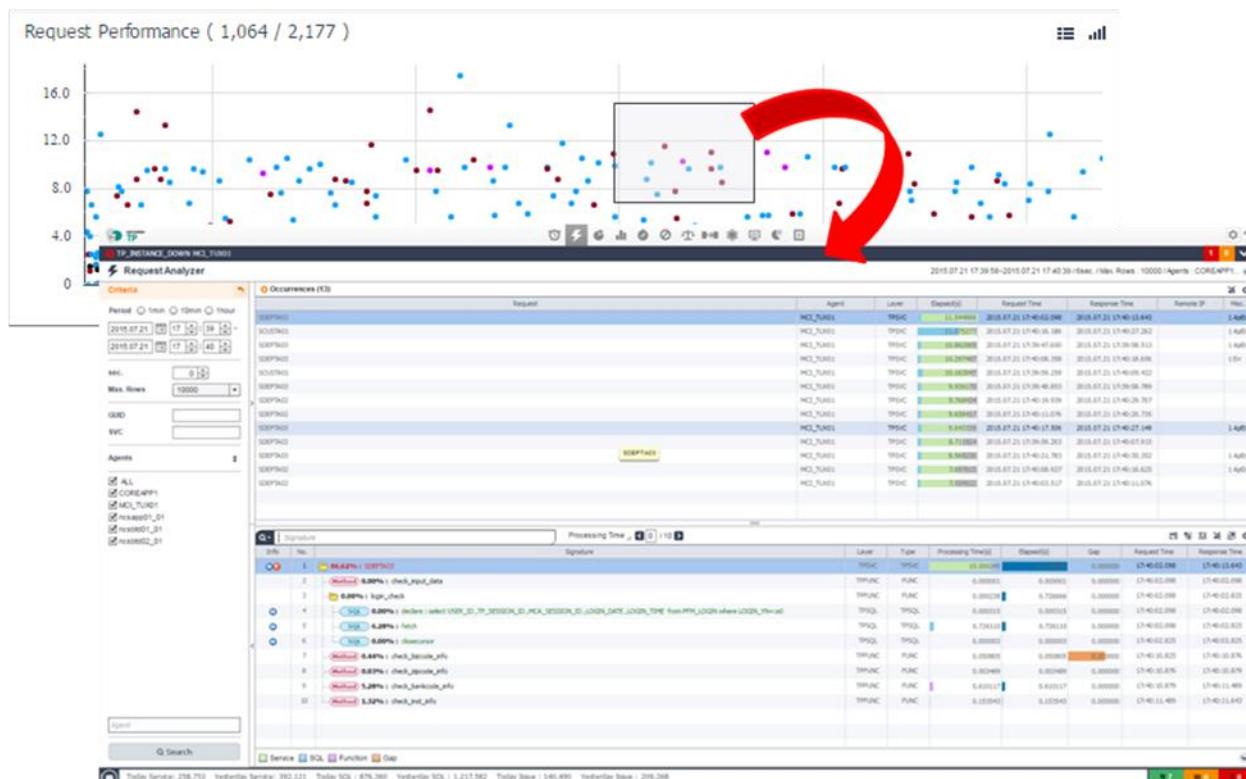
구분색	설명
파란색	정상 수행
검정색	수행 시, DB SQL Error 발생
주황색	수행 시, Memory Leak 발생
형광핑크색	수행 시, TP 서비스 Error 발생
보라색	수행 시, 업무 에러 발생
회색	수행 시, DB Uncommitted 발생 (TP에서는 발생하지 않음)

[표] 2-6 Request Performance 색상

2.10.1. Request Analyzer 연계

Request Performance에 표현되는 Request들을 드래그하여 특정 범위를 선택하면 Request Analyzer 화면으로 전환되어 해당 범위에 포함된 Request들의 상세 정보인 Call-Tree를 확인 할 수 있다.

Request Analyzer에 대한 설명은 <본 문서 "3.Request Analyzer">의 내용을 참조한다



[그림] 2-20 Active Requests 과 Request Analyzer 연계

참고

Request Performance 차트에서 드래그를 통하여 Request Analyzer 화면으로 이동 시 선택된 점 보다 더 많은 Request가 출력되는 경우가 있다. 이는 각 점들이 동일 시간대의 같은 위치에 중복되는 경우가 발생하여 마치 하나의 점으로만 보여지기 때문이다.

03

Request Analyzer

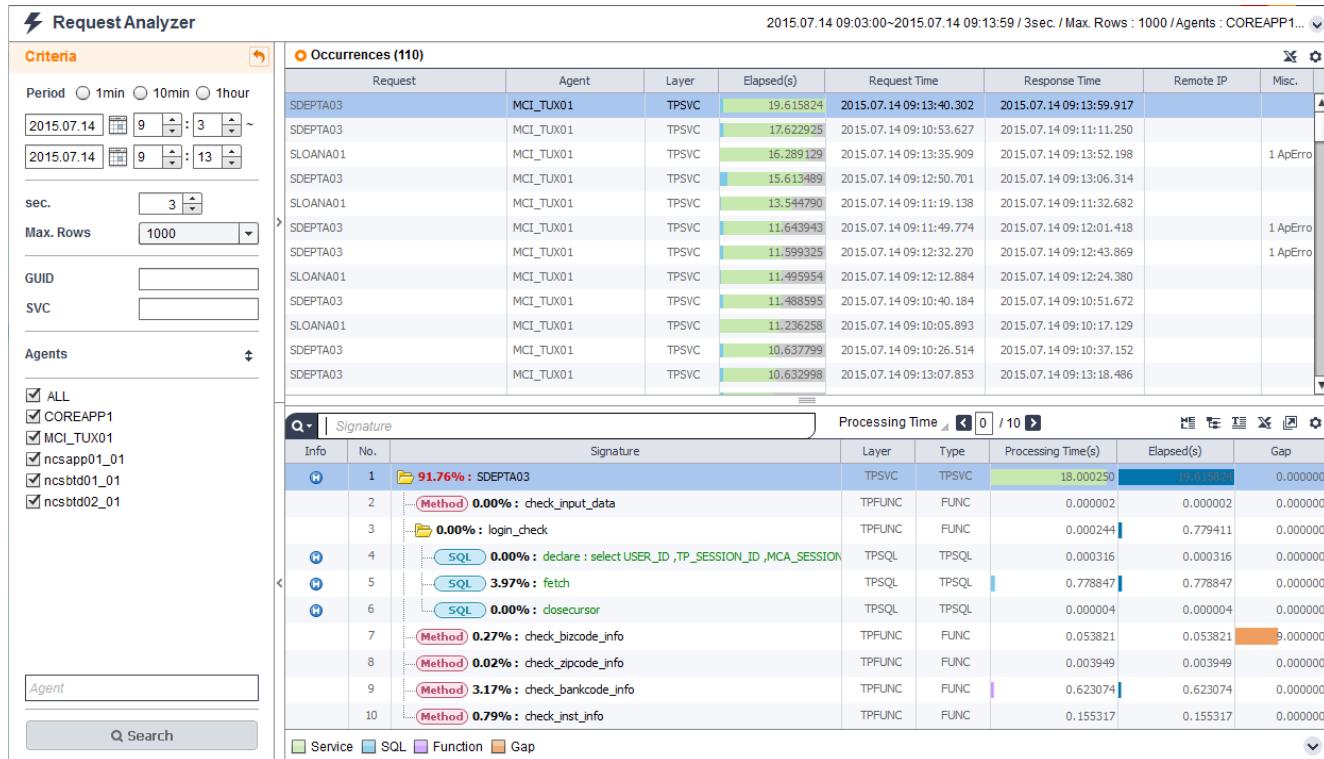
- 3.1 Occurrences 영역
 - 3.2 Call-Tree 영역
 - 3.3 CallTree Summary pop-up
-

3. Request Analyzer

Request Analyzer는 TP서비스내의 처리 플로우가 어떻게 이루어지고 있고, 각각의 함수와 SQL은 무엇이고 그 처리시간이 전체 거래 처리 중 차지하는 비율이 어떠한지, 또 처리결과 값은 무엇인지, 이러한 플로우를 갖는 거래의 속성 값들은 무엇인지 등 거래에 대한 전반적인 내용과 당시의 처리 플로우를 명확히 알려주어 소스레벨까지의 분석이 가능하도록 하는 가장 기본적인 분석화면이다.

Request Analyzer는 거래 시간과 GUID 및 TP서비스명으로 조회를 하여 단독으로도 사용될 수 있지만, 실시간 모니터링, Statistics Analyzer, Top N Analyzer, Issue Analyzer, Event Analyzer 등각 분석 메뉴의 Request Performance 영역과 연계되어 사용자가 선택한 최종 거래 데이터들의 Call-Tree를 분석 할 수 있다는 것이 가장 큰 장점이다.

Request Analyzer에 대한 화면 구성은 아래 [그림] 3-1과 같다.



[그림] 3-1 Request Analyzer 화면 구성

3.1. Occurrences 영역

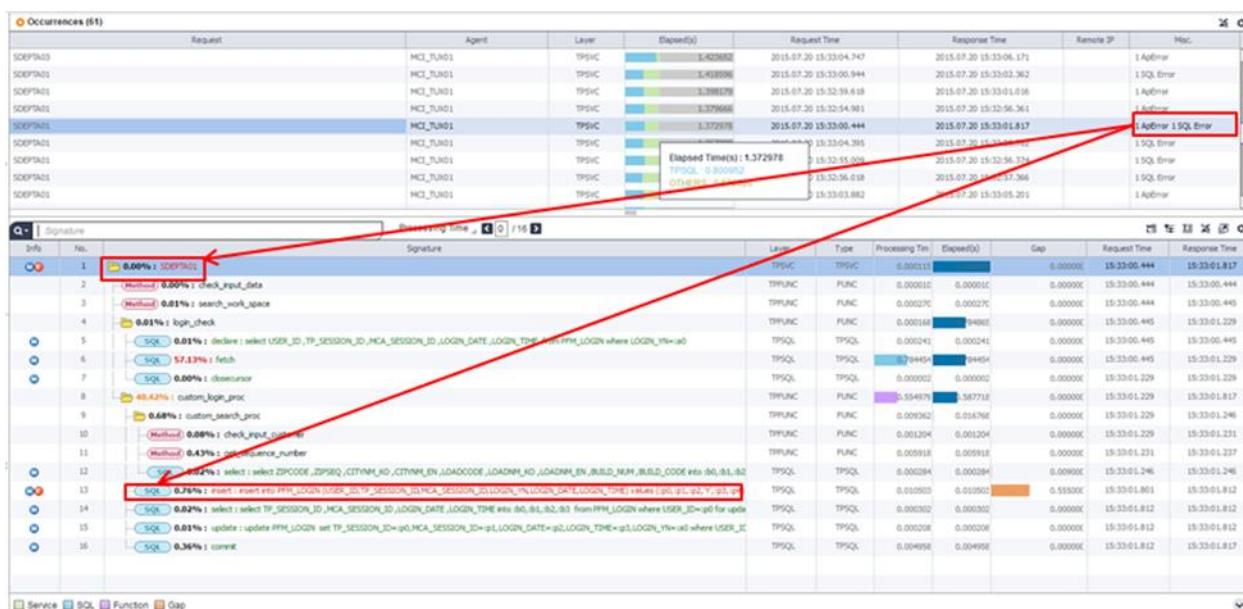
Occurrences 영역은 선택된 거래들에 대한 리스트와 차트, 주요 성능정보들과 에러정보를 포함한다. 각각의 컬럼에 대한 내용은 다음과 같다.

- Request : TP서비스명 또는 거래코드에 의한 거래명을 나타낸다.
- Agent : 각 시스템 별 구성된 TP Monitor 에 대한 ENPHAROS TP Agent 를 나타낸다.
- Layer : ENPHAROS TP 에서는 2가지의 Layer 가 존재한다. TPSVC 와 TPSQL 이다.
- Elapsed(s) : 해당 거래가 TP서비스에서 처리된 총 시간을 나타낸다. 총 2가지의 색상으로써 처리 영역을 구분하는데, 초록색은 TP서비스가 수행한 총 시간을, 파란색은 SQL 이 수행된 총 시간을

나타낸다. 이는 전체 TP서비스 수행시간 대비 SQL 수행시간을 직관적으로 비교하기 위하여 제공하는 기능이다. 커서를 해당 거래의 Elapsed 그래프에 놓으면 관련 데이터 값이 표현된다. 만약 수행되었던 거래들 중에 SQL 수행 시간이 상대적으로 많은 거래를 찾는다면 Elapsed(s)에서 초록색 대비 파란색이 긴 거래를 찾으면 된다.

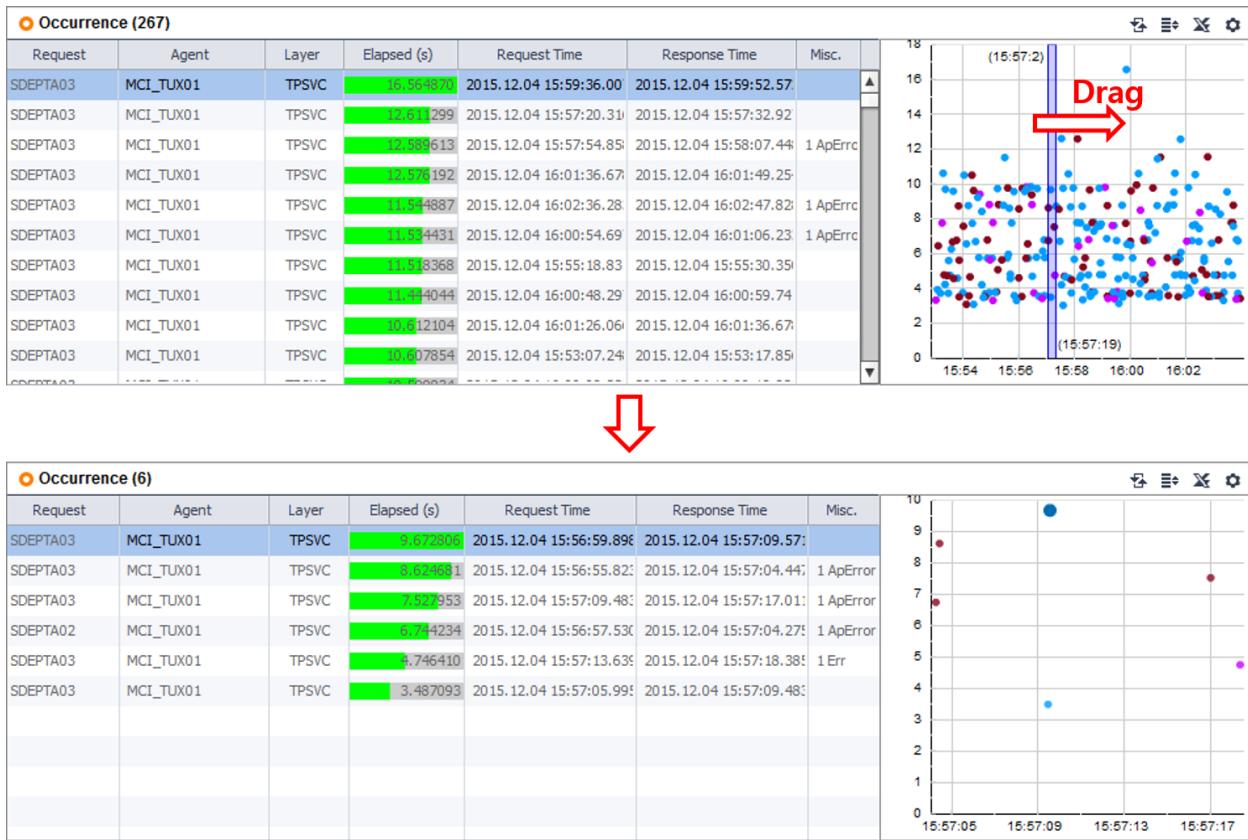
- Misc : 주로 해당 거래에 대한 이슈 사항과 이슈 발생 건수를 확인 할 때 사용된다. Ap Error, SQL Error, Error 등 3가지가 존재하며 발생 시마다 종복과 누적되어 에러정보가 보여진다.
 - ✓ AP Error : pharostp.cfg 파일의 [app_errinfo]에 정의된 에러정보에 의해 발생한 업무에러
 - ✓ SQL Error : SQL 수행 후 결과 값으로 받은 sqlcode 값에 의한 에러
 - ✓ Error : TP서비스의 자체 에러(예: TPFAIL)

[그림] 3-2는 Misc 내용에 대한 Call-Tree 데이터에서 해당 되는 부분을 나타낸 그림으로 Misc 내용에는 “1 ApError 1 SQL Error”으로 표기 되어 있다. Call-Tree 상으로는 TP서비스 에러와 TP서비스 내의 SQL 에러가 발생한 것을 확인 할 수 있다.



[그림] 3-2 Occurrences에서 Misc 정보

Occurrence 영역에 List로 표현된 정보를 우측에 위치한 성능 분포 차트로도 동시에 확인할 수 있다. 차트를 이용하여 성능 분포를 한눈에 확인할 수 있고, 사용자가 원하는 시점을 Zoom in 하여 더 상세하게 확인할 수도 있다. 차트뷰는 우측 상단의 뷰 전환 버튼을 이용하여 표시/숨김할 수 있다.

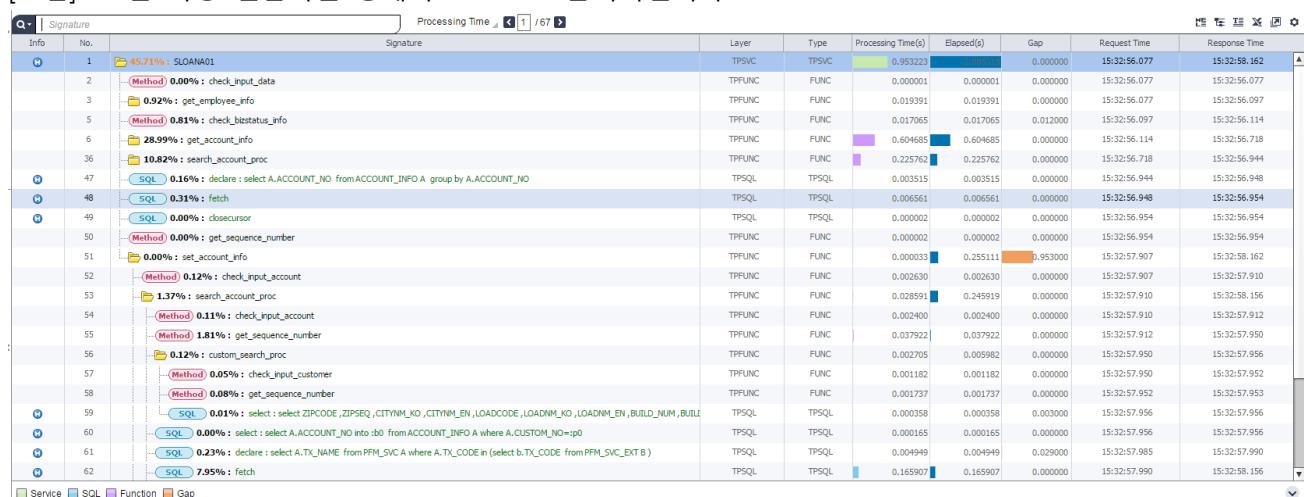


[그림] 3-3 Request 성능 분포 차트

3.2. Call-Tree 영역

Call-Tree 영역은 Occurrences 영역의 대상 거래들 중 선택된 하나의 거래에 대한 각각의 수행된 함수들과 SQL들에 대한 성능정보들과 에러정보, 거래의 속성을 나타내는 메타정보를 포함하는 거래 단위의 상세 성능정보 영역이다. Call-Tree 데이터의 기능은 TP서비스 내의 처리 플로우를 사용자가 쉽게 알 수 있도록 가시성을 부여하여 제공하고, 주요 소스 레벨까지 성능정보를 확인 할 수 있어 장애 발생에 대한 원인 분석 및 TP서비스에 대한 튜닝 시 사용되는 가장 기본적인 분석 화면이다.

[그림] 3-3은 가장 일반적인 형태의 Call-Tree 분석화면이다.



[그림] 3-4 Call-Tree 분석화면

Call-Tree 데이터를 통해서 필히 확인해야 할 부분은 각 함수 및 SQL에 대한 Signature 와 Process time, Elapsed time, Gap time 등의 처리시간 정보들이다. 이를 정보들을 통해서 성능저하가 발생하는 구간이나 문제가 될 소지가 있는 부분을 찾을 수 있기 때문이다.

Call-Tree 영역에서 제공하는 각각의 처리시간 정보들은 총 4가지의 색상으로써 처리영역을 구분하는데, 초록색은 TP서비스가 수행한 총 시간을, 파란색은 SQL 이 수행된 총 시간을, 보라색은 일반 함수가 수행된 총 시간을 나타내고, 주황색은 이전 호출 응답처리 시간과 다음 호출 요청 시간과의 차이인 Gap Time 을 나타낸다. 전체 걸린 시간과 이를 구성하는 색상만으로 대략 어느 영역에서 많은 시간이 걸렸는지를 직관적으로 알 수 있다.

- Elapsed Time(s) : 호출 받은 서비스 또는 함수 또는 SQL이 요청을 받아 이를 처리하고 결과 값을 리턴 하는 데까지 걸린 시간이다.
- Processing Time(s) : 호출 받은 서비스 또는 함수 또는 SQL이 요청을 받아 이를 처리하고 결과 값을 리턴 하는 데까지 걸린 시간에 하위 레벨의 수행되었던 각 함수와 SQL 의 처리시간을 뺀 순수한 해당 서비스 또는 함수 또는 SQL 만의 처리시간
- Gap Time(s): 이전 호출 응답처리 시간과 다음 호출 요청 시간과의 차이를 Gap Time 이라 하는데, 이러한 시간 차가 발생하는 경우는 이전 호출과 다음 호출 간에 어떠한 pending이 발생할 수 있는 로직 또는 hooking 하지 않는 사용자 함수에 의해 발생 된 시간 차로 이해하면 된다. 만약 ATMI 함수를 이용한 TP서비스 호출인 경우에는 Gap Time 을 통해 TP Monitor의 Queue에 적재되어 Pending 된 시간을 산출할 수 있다.

3.2.1. 서비스 탑

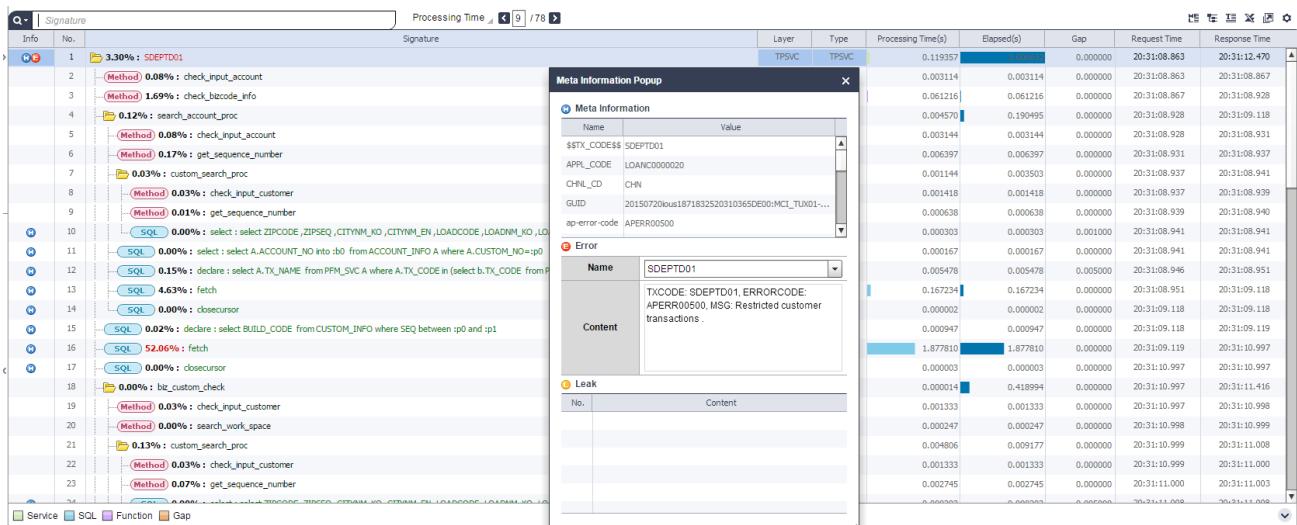
Call-Tree 데이터의 가장 윗 상단 첫 부분은 Occurrences 의 Request와 동일한 데이터로써 서비스명 또는 거래명(거래코드)으로써 거래를 나타내는 데이터로 'TPSVC' 데이터 탑으로 정의 된다. 이 부분에 대한 중요 데이터 항목으로는 Processing Time 과 Elapsed Time 이 있고, 거래의 속성을 나타내는 메타정보와 처리결과를 나타내는 오류정보가 있다.

거래의 속성을 나타내는 메타정보에는 TX_CODE(거래코드), APPL_CODE(업무코드), CHNL_CD(채널코드), User data 부, GUID 등의 업무 종속적인 코드 값들과 DB 세션정보, SQL 커서 정보 등을 포함하고 있다.

처리결과를 나타내는 오류정보에는 업무서비스에서 정의한 오류코드와 그에 따른 오류 메시지를 포함한다.

이들에 대한 설정은 <*ENPharosTP_AdministrationGuide* 의 "3.2.8 포지션정보, 3.2.9 에러위치 정보">를 참고한다.

[그림] 3-4는 메타정보와 오류정보가 함께 설정 된 거래에 대한 Call-Tree 데이터의 메타정보이다. 메타정보로는 TX_CODE(거래코드), APPL_CODE(업무코드), CHNL_CD(채널코드), GUID에 대한 설정 된 값들이 있고, 오류정보로는 오류코드와 오류 메시지가 설정되어 있는 것을 확인 할 수 있다.



[그림] 3-5 Call-Tree 서비스 데이터 및 메타정보

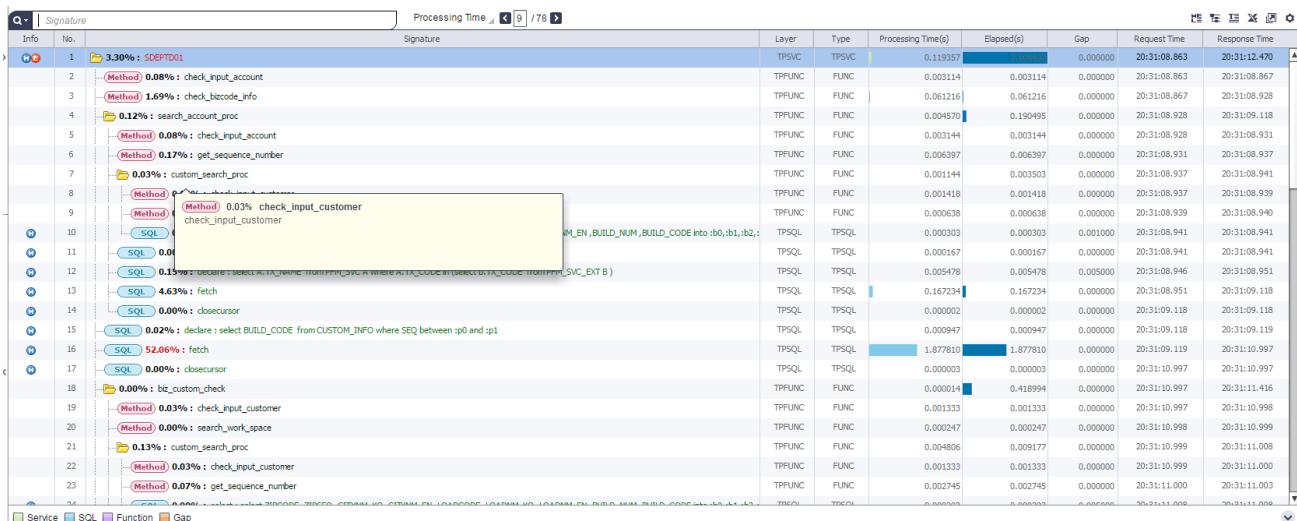
3.2.2. Method 탑입

Call-Tree 데이터 중 가장 많은 부분을 차지하는 데이터 탑입 중의 하나로써 주로 사용자함수가 대상이 되며 'FUNC'의 데이터 탑입으로 정의된다. 사용자함수는 관련된 메타정보가 없고, 단지 수행시간과 처리 결과 값에 대해서만 관리를 한다.

일반적인 사용자함수인 경우 표준화가 되어 있지 않기 때문에 이를 위한 특별한 설정과 작업이 필요하다.

이에 대한 내용은 <*ENPharosTP_AdministrationGuide* 의 "4. 사용자 함수 성능정보 추출 방법"> 을 참고한다.

[그림] 3-5는 일반적인 형태의 사용자함수에 대한 Call-Tree 데이터로 커서를 가까이 가져가면 관련된 내용의 툴팁이 활성화 된다.

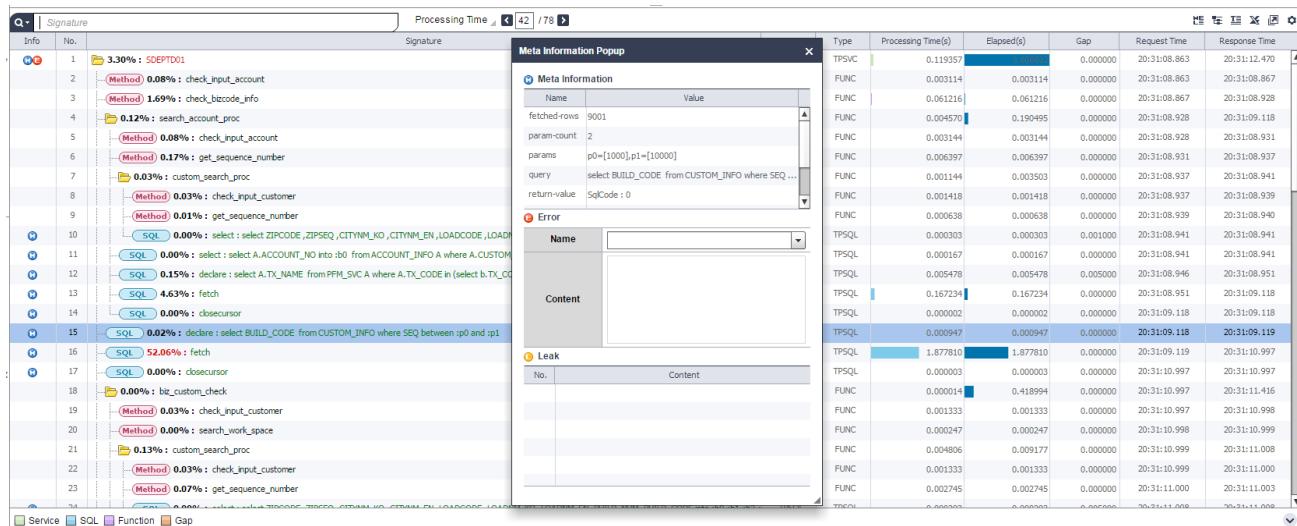


[그림] 3-6 Call-Tree Method 데이터

3.2.3. SQL 탑

Call-Tree 데이터 중에서 사용자함수와 더불어 가장 많은 부분을 차지하고, 성능 이슈가 발생 하였을 때 가장 먼저 확인해 봄야 할 부분이 바로 SQL 부분이다. 데이터 탑으로는 'TPSQL' 으로 정의된다. SQL은 일반 사용자함수와는 달리 SQL 수행결과정보를 위한 메타정보가 존재한다. 이 메타정보에는 수행된 SQL에 대한 전반적인 내용으로 구성되어 있는데 SQL Full-Text 를 포함하여 Bind Variable 개수, Bind Value, Fetch 건수, Update 건수, SQL Code 값, 소스파일에 대한 정보 등으로 구성되어 있다.

[그림] 3-6은 SQL 탑의 Call-Tree 데이터에 대한 메타정보를 나타낸 그림이다.



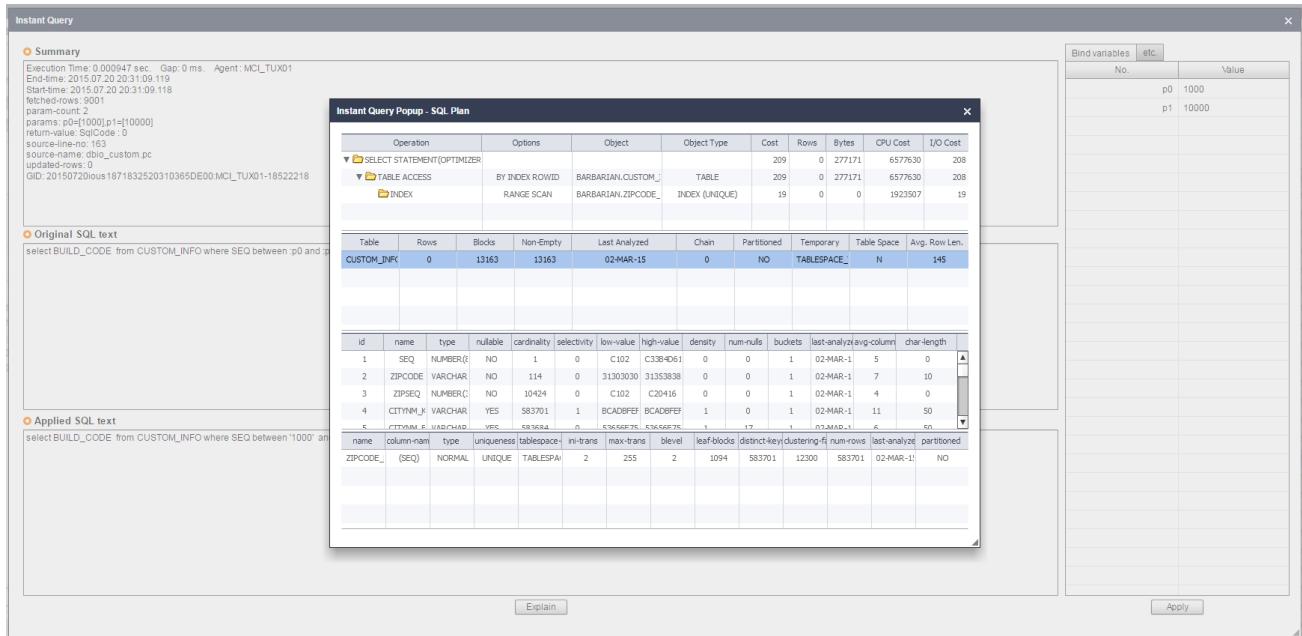
[그림] 3-7 Call-Tree SQL 데이터 및 메타정보

메타정보의 항목들 중 'query' 부분의 값을 클릭하면 [그림] 4-7 의 SQL 수행정보가 pop-up 으로 뜬다. 상단의 summary 부분은 위에서 설명한 SQL 메타정보가 출력되고 우측 부분에는 Bind Variable 과 Bind Value 에 대한 내용이 출력되고, 중간 Original SQL Text 부분에는 SQL Full-Text 가 출력된다.



[그림] 3-8 Instant Query 정보

Instant Query pop-up 창의 아래 'Apply' 버튼을 클릭을 하면 'Explain' 버튼이 활성화가 되면서 아랫부분의 'Applied SQL Text' 부분에 우측 Bind Value로 설정된 SQL Full-Text 가 만들어진다. 이때 'Explain' 버튼을 클릭하게 되면 [그림] 3-8의 SQL PLAN 정보가 출력된다.



[그림] 3-9 SQL PLAN 정보

이를 통하여 성능저하가 발생 하거나 문제가 될 소지가 있는 SQL에 대하여 어렵지 않게 DBA 수준의 SQL 튜닝이 가능하다.

단, SQL PLAN 기능을 사용하기 위해서는 설정이 필요하다.

설정과 관련해서는 <*ENPharosTP_AdministrationGuide_V5* 의 F.SQL PLAN 기능 설정 및 확인>을 참고하도록 한다.

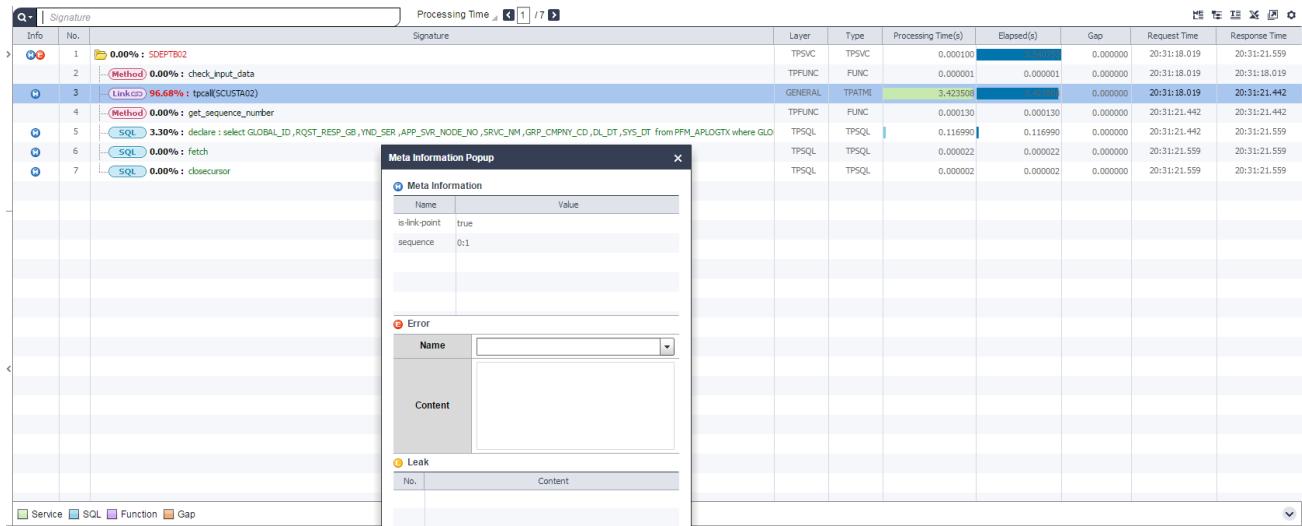
3.2.4. Link 타입

Call-Tree 데이터에서 Link 라 함은 TP서비스간 연계를 의미한다. TP Monitor에서는 TP서비스 호출을 통하여 프로세스 분산처리를 지원한다. 이때 사용되는 표준 API 가 ATMI(Application Transaction Management Interface)이고, ENPHAROS TP 에서 지원하는 ATMI 로는 tpcall, tpacall, tpforward, tpcancle, tpreturn 등이 있다. 이들 API 를 사용하여 TP서비스 호출이 발생하는 경우 Call-Tree 데이터에서는 'ATMI 함수(TP서비스명)'으로 Signature 를 표기하고, Signature 타입을 Link 로 표기한다. 또한 Layer 는 'GENERAL', Type 은 'TPATMI' 로 표기한다.

[그림] 3-9은 TP서비스 호출 시에 Signature 에 표기되는 내용과 그에 따르는 메타정보를 나타내는 그림이다. 메타정보에 표기되는 중요정보로는 'sequence' 가 있으며, 용도는 Multi 서비스에 대한 전체 Call-Tree 를 출력할 때 연계되는 순서 시컨스로 사용된다. sequence는 처음 거래를 처리하는 TP서비스에서는

항상 그 값이 '0' 이다. sequence에서 콜론(:)의 개수는 TP서비스 호출 depth를 가르키며 숫자의 증가는 해당 TP서비스 내에서 다른 TP서비스를 호출한 횟수를 나타낸다.

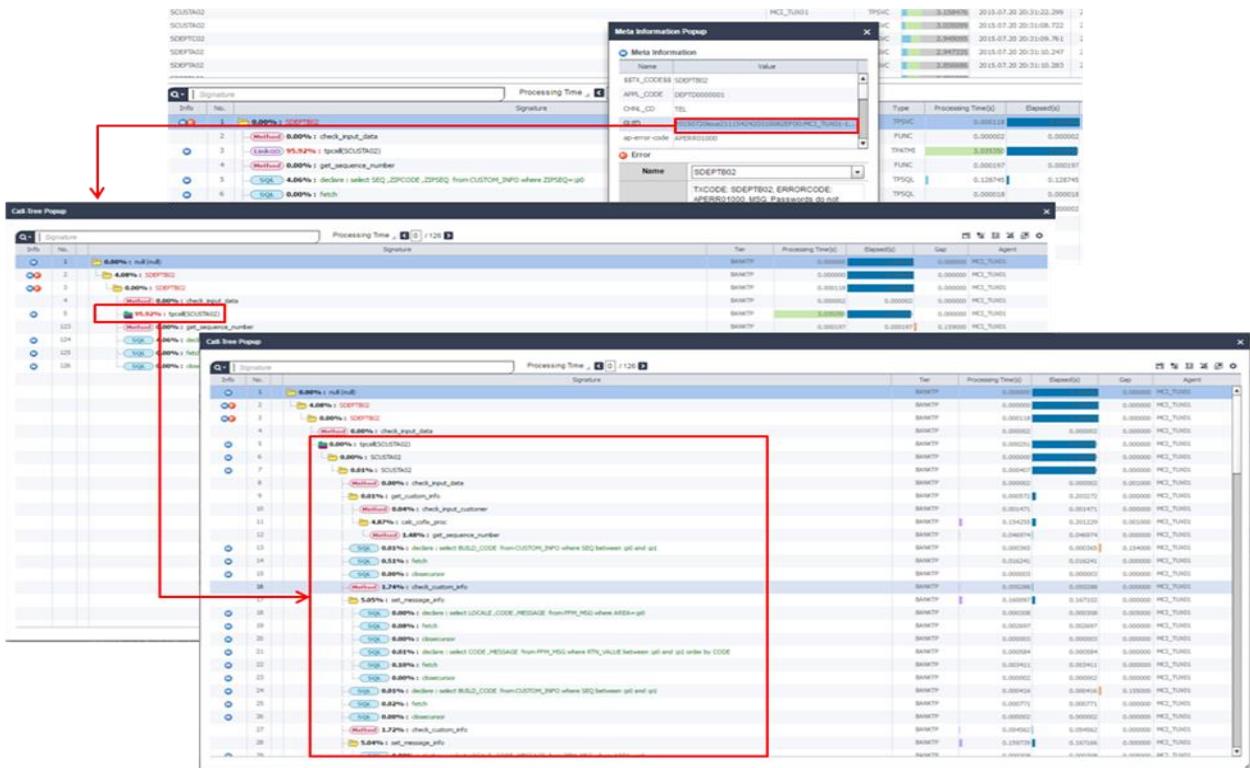
예를 들어 '0:1:1:2:1' 의 시퀀스를 갖고 있는 TP서비스는 하나의 거래를 처리하기 위해 5번째로 호출 된 TP서비스이고, 4번째 TP서비스 내에서 두 번째로 호출된 TP서비스이다.



[그림] 3-10 Link 정보

Call-Tree 데이터의 서비스 탑재에 대한 메타정보들 중에 GUID 부분의 값을 클릭하게 되면 해당 거래에 대한 TP서비스 연계정보를 포함한 전체 Call-Tree 가 출력된다. Multi 서비스를 이용한 연계거래에 있어 매우 유용한 기능으로 거래관점에서 전체 플로우 중 어느 서비스에서 어느 구간이 성능저하가 발생 했다는 것을 직관적으로 파악할 수 있다.

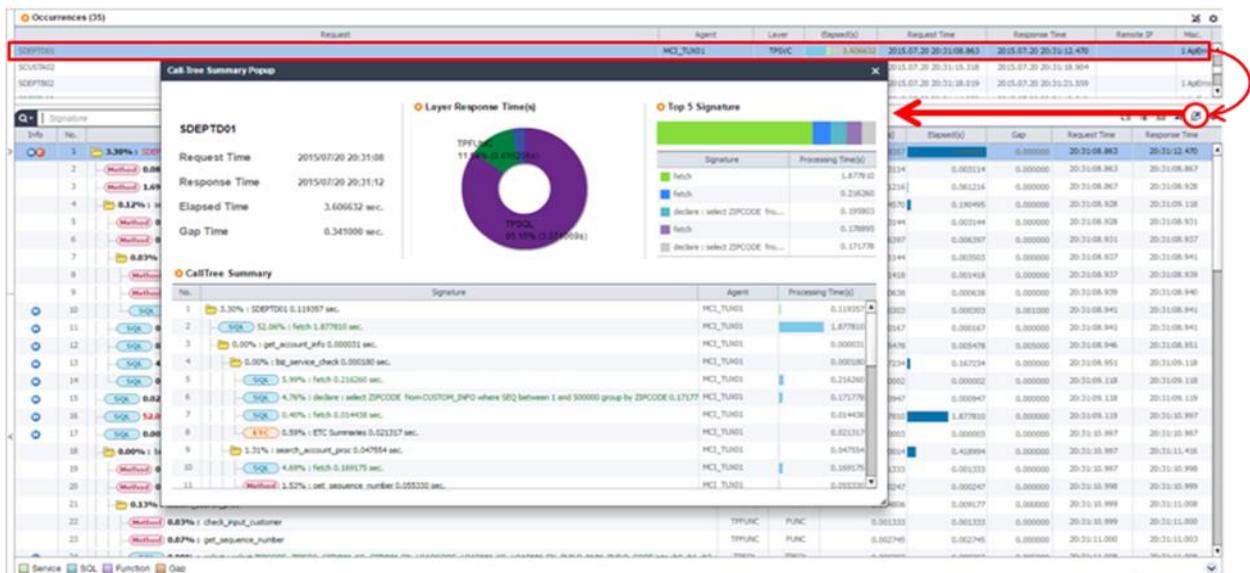
[그림] 3-10은 이러한 과정을 보여주는 그림으로, 두 개의 TP서비스 처리과정을 마치 하나의 TP서비스가 처리한 것처럼 Call-Tree 데이터를 제공해 주고 있다.



[그림] 3-11 전체 Call-Tree 정보

3.3. CallTree summary pop-up

Calltree 영역 우측 상단의 Calltree Summary PopUp 버튼을 클릭하면 팝업으로 Calltree Summary를 출력한다. 이는 한 거래에 대하여 한 눈에 모든 상황을 가시성을 높여 직관적으로 확인 할 수 있는 기능을 제공한다.



[그림] 3-12 CallTree Summary pop-up 화면

04

Statistics Analyzer

- 4.1 TPSVC 통계분석
 - 4.2 TPSQL 통계분석
-

4. Statistics Analyzer

Statistics Analyzer는 특정 시간 동안의 총 거래량, 평균 처리 시간, 초당 처리 건수, 시간 대별 처리현황, 평균 응답시간, CPU 및 Memory에 대한 시스템 자원 사용률 등의 통계 정보를 토대로 여러 방면으로 비교/분석할 수 있는 화면이다. 특징으로는 차트의 내용을 Data Grid 형태로 출력하여 분석할 수 있으며, 특정 시점의 Request Top-N, Request Performance를 제공하여 업무 각각에 대해 상세 분석이 가능하고, 각 차트는 일/시/분 단위로 드릴다운 기능을 제공하여 단계적인 분석이 가능하다는 장점이 있다.

ENPHAROS TP의 Statistics Analyzer 화면은 2개의 Layer를 제공한다. Layer 선택에 따라 'TPSVC'와 'TPSQL'을 기준으로 각 영역에 대한 통계 차트 및 데이터를 검색할 수 있다. 'TPSVC' 선택이면 TP서비스에 대한 각종 통계차트 및 데이터를, 'TPSQL' 선택이면 SQL 수행에 대한 각종 통계차트 및 데이터를 검색할 수 있다. 즉, 거래 및 TP서비스에 대한 통계분석을 위해서는 'TPSVC' Layer를 선택하고, 순수 SQL에 대한 통계분석을 위해서는 'TPSQL' Layer를 선택하여 분석한다.

Statistics Analyzer에 대한 화면 구성은 [그림] 4-1과 같다.



[그림] 4-1 Statistics Analyzer 화면 구성

화면 구성에 대한 자세한 설정 및 사용법은 <*ENPharosTP_화면안내서* 의 "5.Statistics Analyzer"> 부분을 참고한다.

4.1. TPSVC 통계분석

일반적인 업무거래 또는 TP서비스에 대한 통계분석을 위해서는 Layer를 'TPSVC'를 선택하고, 원하는 시간 대를 설정하여 조회하고 각 차트들의 그래프 내용을 비교/분석한다.

대체로 Throughput 차트, Elapsed Time 차트, TPS 차트, Active Request 차트 및 Min.Active Request와 Max.Active Request 차트는 유사한 형태의 그래프 유형을 보인다. 그러나 Avg.Response Time 차트는 이

와는 반대의 증감을 나타낸다. 즉, 거래 부하량과 처리량이 높아 질수록 Avg.Response Time 차트의 그래프는 낮아지고, 거래 부하량과 처리량이 낮아 질수록 Avg.Response Time 차트의 그래프는 높아진다. 이러한 관계를 잘 파악하여 이들 차트들 중 어느 한 차트의 그래프가 이상징후를 보였다면 이는 TP서비스에 문제가 있으므로 반드시 확인하여 그러한 문제가 추후에 발생하지 않도록 해야 한다.

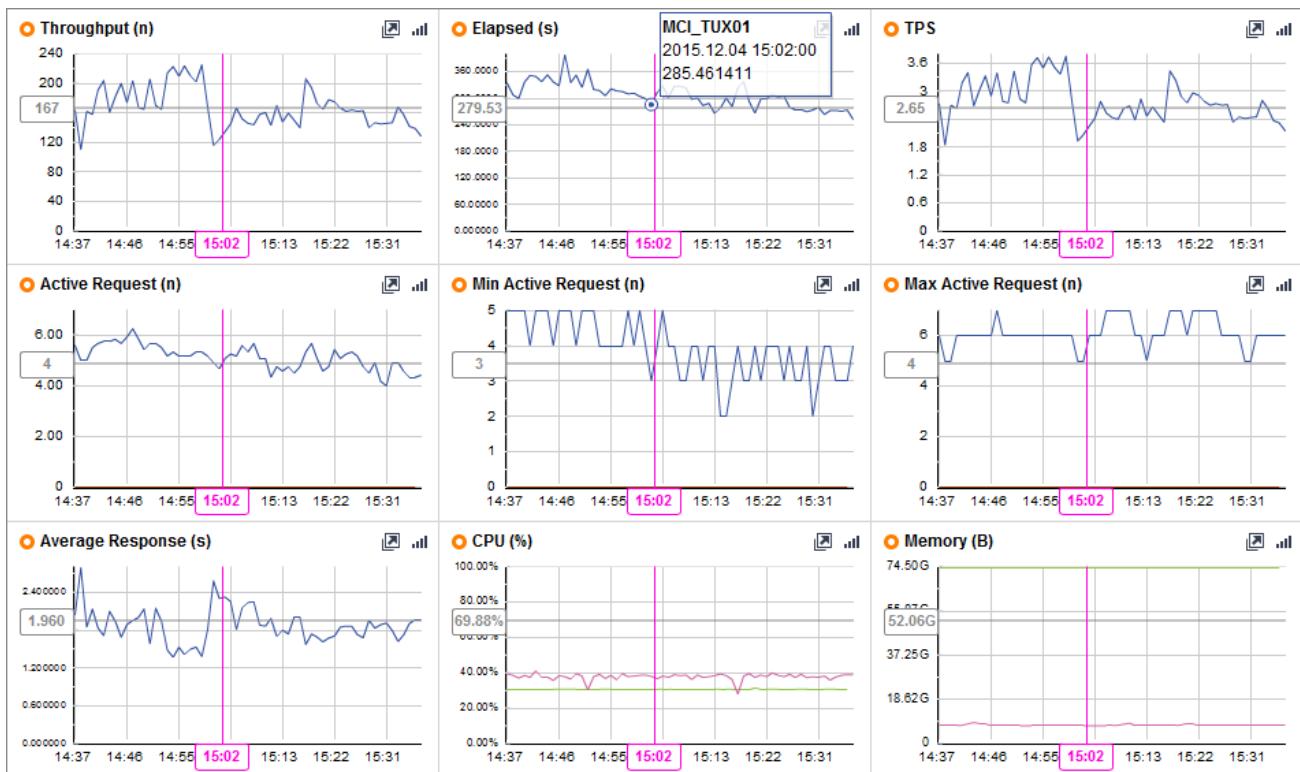


[그림] 4-2 Statistics Analyzer의 TPSVC 차트

다음은 각 차트 별 그래프의 증감에 따라 문제가 될 소지가 있는 케이스에 대한 예이다.

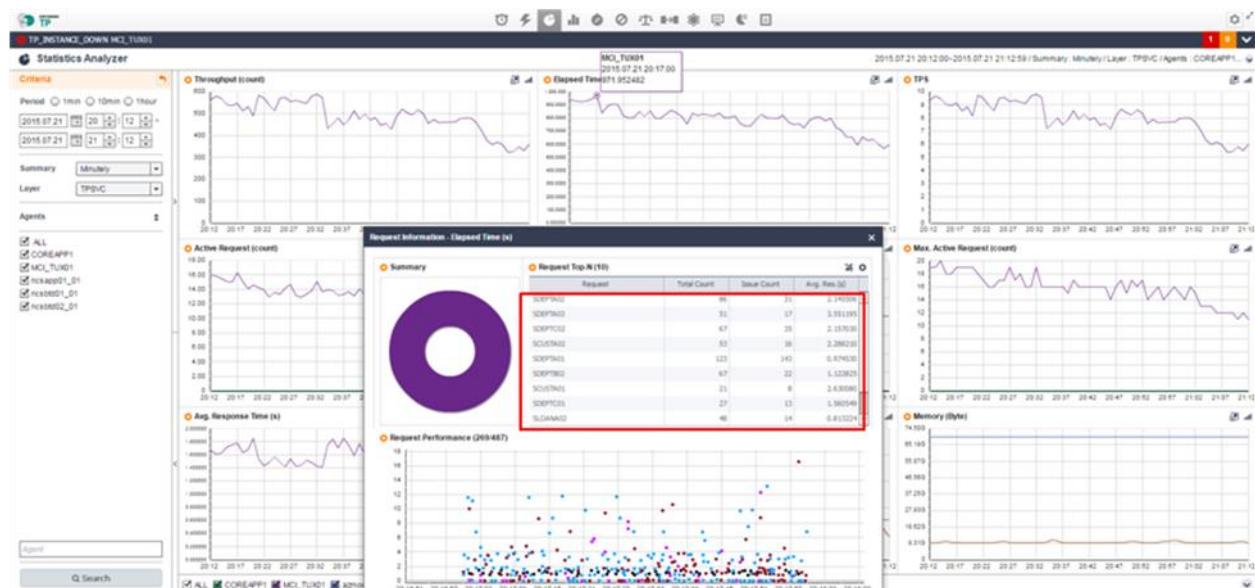
- CPU 사용률이 급격히 튀는 구간이 발생 → 각 차트 비교 후 이상이 없으면 Batch-Job 수행
- Memory 가 점차적으로 증가한다. → 일부 프로세스에 메모리 leak 발생 가능성 존재
- CPU 사용률은 얼마 안 되는데 TPS 가 낮음 → TP Monitor 및 TP서비스 튜닝 필요
- Throughput 와 TPS는 증가하는데 Elapsed Time 과 Response Time 은 감소 → TP서비스에 문제 발생 가능성 존재
- Throughput 은 일정하거나 감소하는데 Active Request 는 계속적으로 증가 → TP서비스 또는 DBMS에 문제 발생 가능성 존재

차트 위에 마우스를 오버하면 커서가 위치한 지점의 X축 값과 Y축 값을 표시하는 가로/세로 바가 표시되고, 각 XY축에 값이 표시된다. 이 XY값들을 통하여 특정 시점의 통계 데이터를 한눈에 비교 분석할 수 있다.



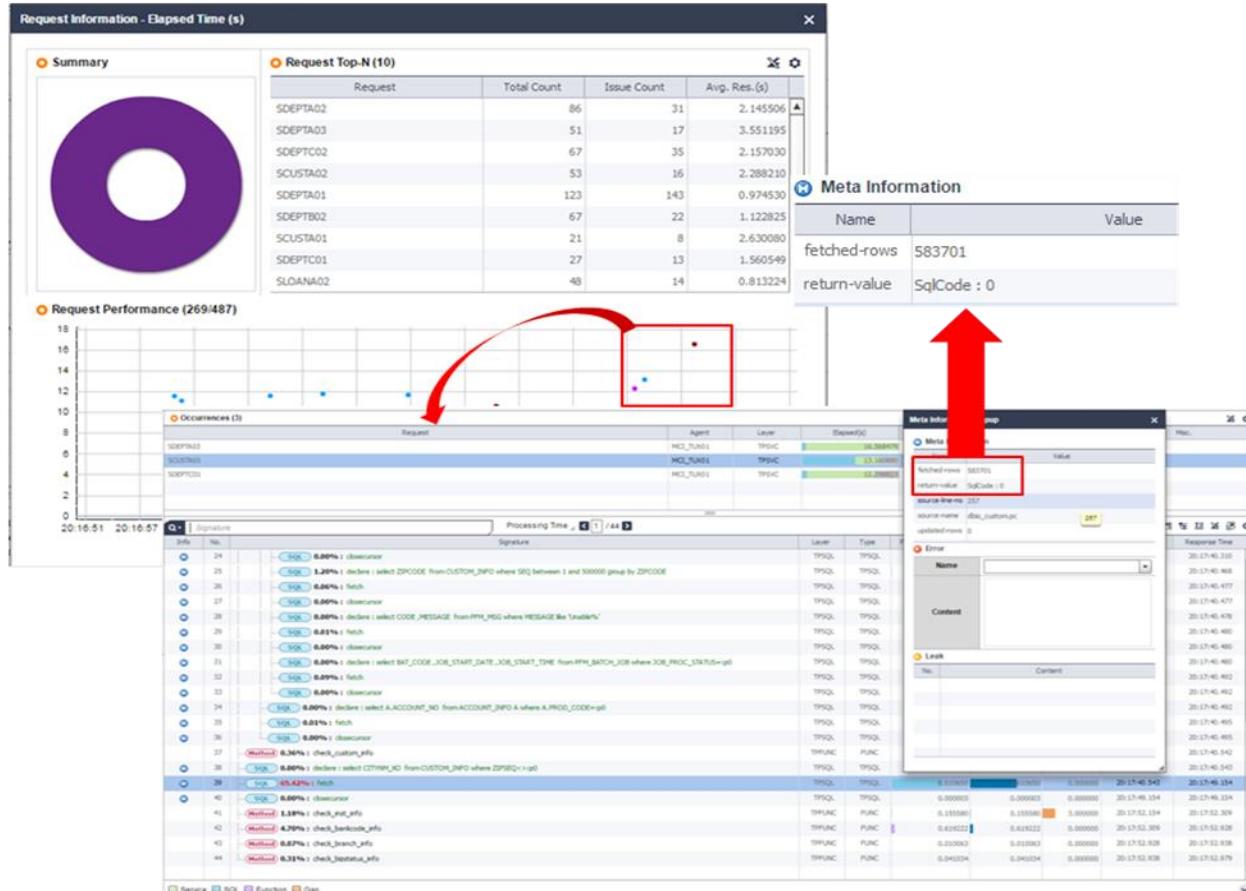
[그림] 4-3 XY축 막대-값 표시

[그림]4-3은 Elapsed Time 차트에 대하여 가장 오랜 시간을 차지했던 구역에 대한 Request 정보를 나타내는 차트로써 Request Top-N 에는 당시에 수행했던 TP서비스들이 평균 응답시간 기준으로 정렬되어 표현되고 있고, 아래의 Request Performance 차트에는 당시의 관련된 거래에 대한 처리결과 분포도가 형성되어 있다.



[그림] 4-4 TPSVC의 Elapsed Time에 대한 Request Top-N 차트

[그림] 4-4는 Request Performance 차트의 거래들 중 문제가 있는 3개의 거래에 대하여 드래그를 통해 해당 거래의 상세성능 정보를 확인 할 수 있는 Request Analyzer 화면으로 이동하여 Call-Tree 구조를 파악하는 그림이다. 이러한 과정을 통하여 문제가 되는 구간의 원인은 SQL fetch 건수가 많아 전체적으로 서비스 성능이 저하된 것을 알 수 있다.



[그림] 4-5 TPSVC의 Elapsed Time Request Top-N 차트와 Request Analyzer 연계

4.2. TPSQL 통계분석

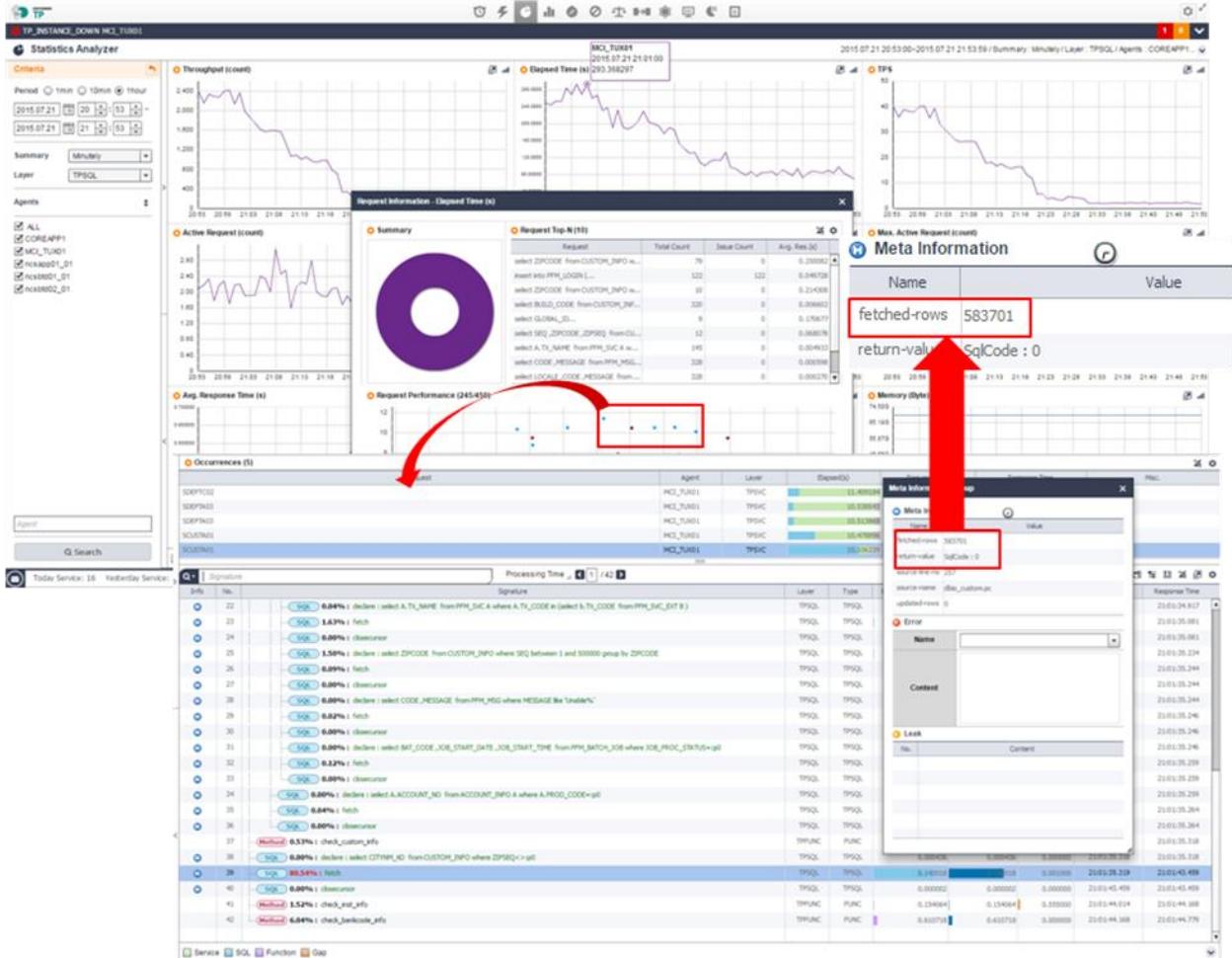
각 업무 프로그램에서 수행 되었던 SQL에 대한 통계분석을 위해서는 Layer를 'TPSQL'를 선택하고, 원하는 시간대를 설정하여 조회하고 각 차트들의 그래프 내용을 비교/분석한다.

'TPSQL'에 대한 통계 차트도 'TPSVC' 와 유사한 형태로 진행이 되므로 같은 방식으로 분석을 진행하면 된다.



[그림] 4-6 Statistics Analyzer의 TPSQL 차트

[그림] 4-6은 Elapsed Time 차트에 대하여 가장 오랜 시간을 차지했던 구역에 대한 Request 정보를 나타내는 차트로써 Request Top-N 에는 당시에 수행했던 TPSQL들이 평균 응답시간 기준으로 정렬되어 표현되고 있고, 아래의 Request Performance 차트에는 당시의 관련된 거래에 대한 처리결과 분포도가 형성되어 있는 상태이다. 이를 Request Performance 차트의 거래들 중 가장 Elapsed Time 이 오래 걸린 5개의 거래에 대하여 드래그를 통해 해당 거래의 상세성능 정보를 확인 할 수 있는 Request Analyzer 화면으로 이동하여 Call-Tree 구조를 파악하는 그림이다. 이러한 과정을 통하여 문제가 되는 구간의 성능저하 원인은 SQL fetch 건수가 많아 성능이 저하되었다는 것을 알 수 있다.



[그림] 4-7 TPSQL의 Elapsed Time Request Top-N 차트와 Request Analyzer 연계

05

Top-N Analyzer

- 5.1 Avg.Response Time
 - 5.2 Sum Response Time
 - 5.3 Execution Count
 - 5.4 Issue Count
-

5. Top-N Analyzer

Top-N Analyzer는 전체 업무 또는 선택한 특정 업무에 속하는 최하위 거래에 대해서 Criteria에 입력한 조건에 따라서 분/시간/일 단위로 처리시간의 합, 평균 처리시간, 실패 건수, 이슈 건수, 성공 건수 등에 대한 Top-N 정보를 제공한다. 동 정보를 이용하여 가장 많이 처리된 거래가 무엇이고, 처리 기간이 가장 오래 걸린 거래가 무엇인지, 실패 건수가 가장 많은 거래가 무엇인지를 분석할 수 있고, 이러한 분석을 통해 튜닝을 위한 기초 자료로써 사용이 된다.

화면의 기본구성은 각 TP서비스들의 처리상황에 대한 가시성을 부여하기 위하여 총건수/이슈발생 건수 기준의 Top10 차트 뷰 화면이 메인으로 구성되어 있으며 아래 부분의 원 차트들은 각각 Top-N 데이터의 응답시간에 대한 Min, Average, Max, Sum 등의 내용들을 포함하고 있어 한눈에 처리결과에 대한 통계치를 비교 확인할 수 있다.

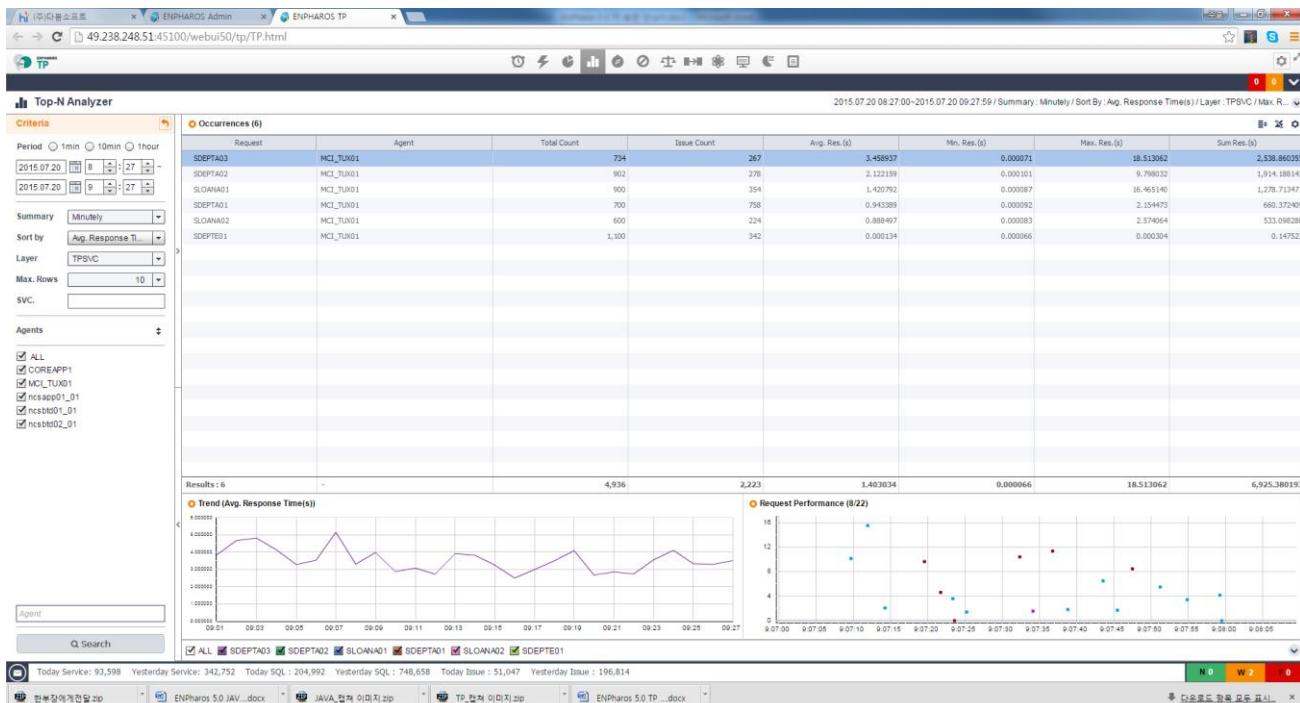
Top-N Analyzer에 대한 차트-뷰 화면 구성은 [그림] 5-1 과 같다.



[그림] 5-1 Top-N Analyzer Chart-View 화면 구성

오른쪽 상단의 뷰 전환 버튼을 클릭하면 리스트 뷰 화면으로 전환하게 된다. 화면 구성은 상단의 Occurrences에서는 처리완료 되어 통계 데이터로 산출 된 TP서비스에 대한 Top-N 을 Grid 데이터로 제공하고, 아래 부분에는 해당 시간 동안의 Trend 차트와 각 거래에 대한 처리결과를 보여주는 Request Performance 차트를 제공해 주고 있다. 이들 3가지의 데이터와 차트 분석을 통해 문제가 있는 TP서비스 와 Request 를 찾아 확인할 수 있고, 특히 Request Performance 차트를 통해 해당 거래에 대한 Call-Tree 를 직접 확인하여 문제가 발생한 구간에 대한 튜닝 및 문제해결이 가능하다.

Top-N Analyzer에 대한 리스트-뷰 화면 구성은 [그림] 5-2과 같다.



[그림] 5-2 Top-N Analyzer List-View 화면 구성

Layer 선택에 따라 'TPSVC' 또는 'TPSQL' 을 기준으로 Top-N을 검색할 수 있다. 'TPSVC' 선택이면 Query란에 TP서비스명 전체 또는 일부를 입력하여 입력한 서비스명을 포함하고 있는 TP서비스에 대한 내용을 검색할 수 있고, 'TPSQL' 선택이면 Query란에 관련 된 SQL 전체 또는 일부만 입력하여 입력된 Text를 포함하고 있는 SQL에 대한 내용을 검색할 수 있다. Query란에 입력없이 전체 대상으로 Top-N을 검색하고자 한다면 거래명 또는 TP서비스에 대한 내용은 'TPSVC'를, 거래 내에서 수행된 SQL에 대한 내용은 'TPSQL'을 선택하면 된다.

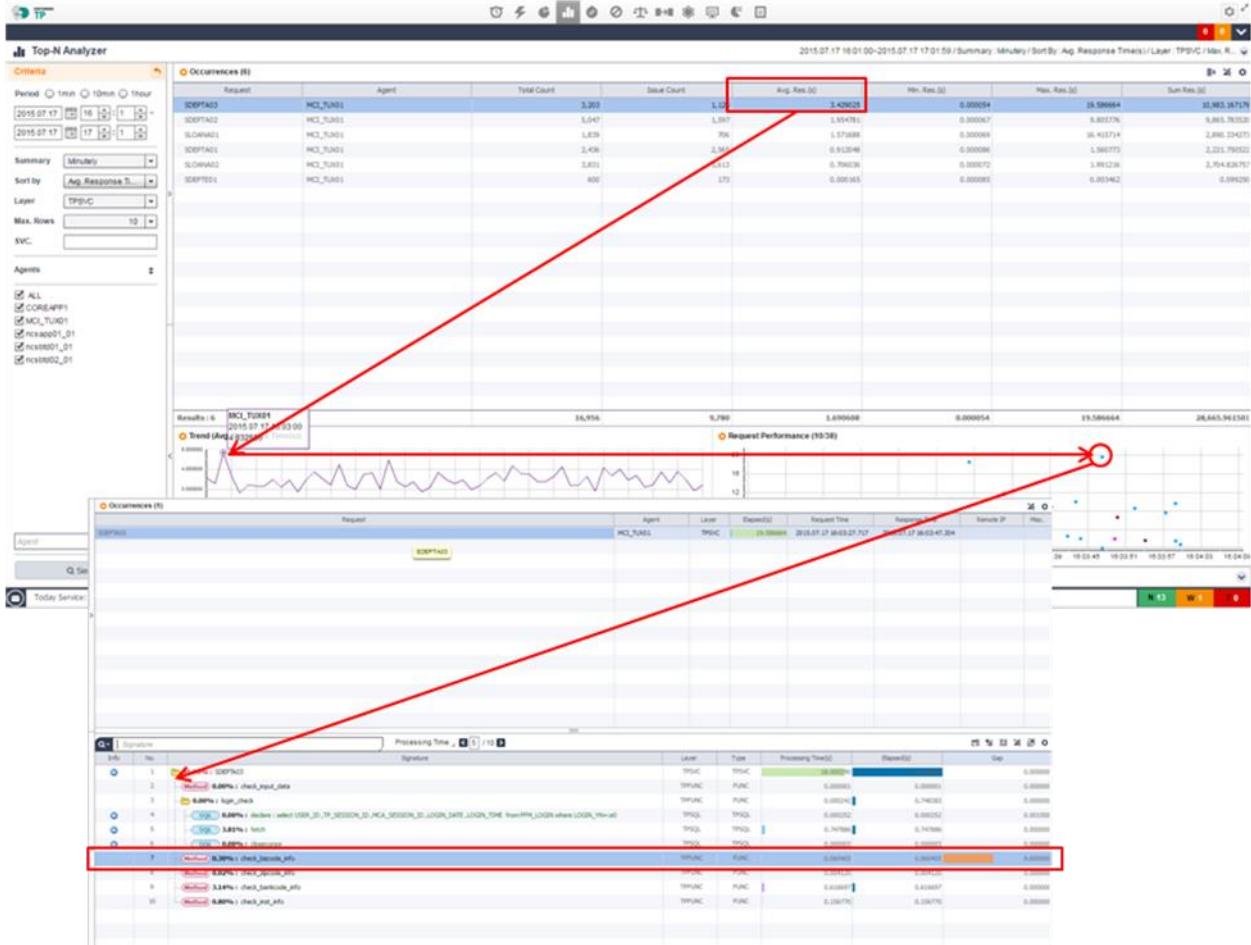
관련된 UI 사용법에 대한 설명은 <ENPHAROS TP_화면안내서 의 "6.Top-N Analyzer"> 부분을 참고한다.

5.1. Avg.Response Time

조회 하고자 하는 시간대에 발생한 모든 Request에 대해 평균 처리시간을 내림차순으로 정렬하여 상위 10개의 TP서비스 또는 SQL에 대하여 분석할 수 있는 데이터를 Chart-View 또는 List-View 형식의 데이터로 표현하여 제공한다.

[그림] 5-3은 TP서비스를 평균 응답시간에 대하여 산출한 Top-N 데이터를 List-View 형식으로 확인하고, Trend 차트를 통하여 평균 응답시간이 가장 오래 걸린 구간을 찾아 Request 를 확인한 후 그 중에서도 가장 오래 걸린 request를 드래그하여 Request Analyzer 연계를 통한 거래의 Call-Tree를 상세 분석하는 과정을 나타낸 그림이다.

이와 같은 과정을 통해 Request Analyzer 연계를 통한 거래의 Call-Tree 데이터에서 확인 할 수 있는 부분은 Gab Time 때문에 평균응답 시간이 느려졌다는 것을 알 수 있다.



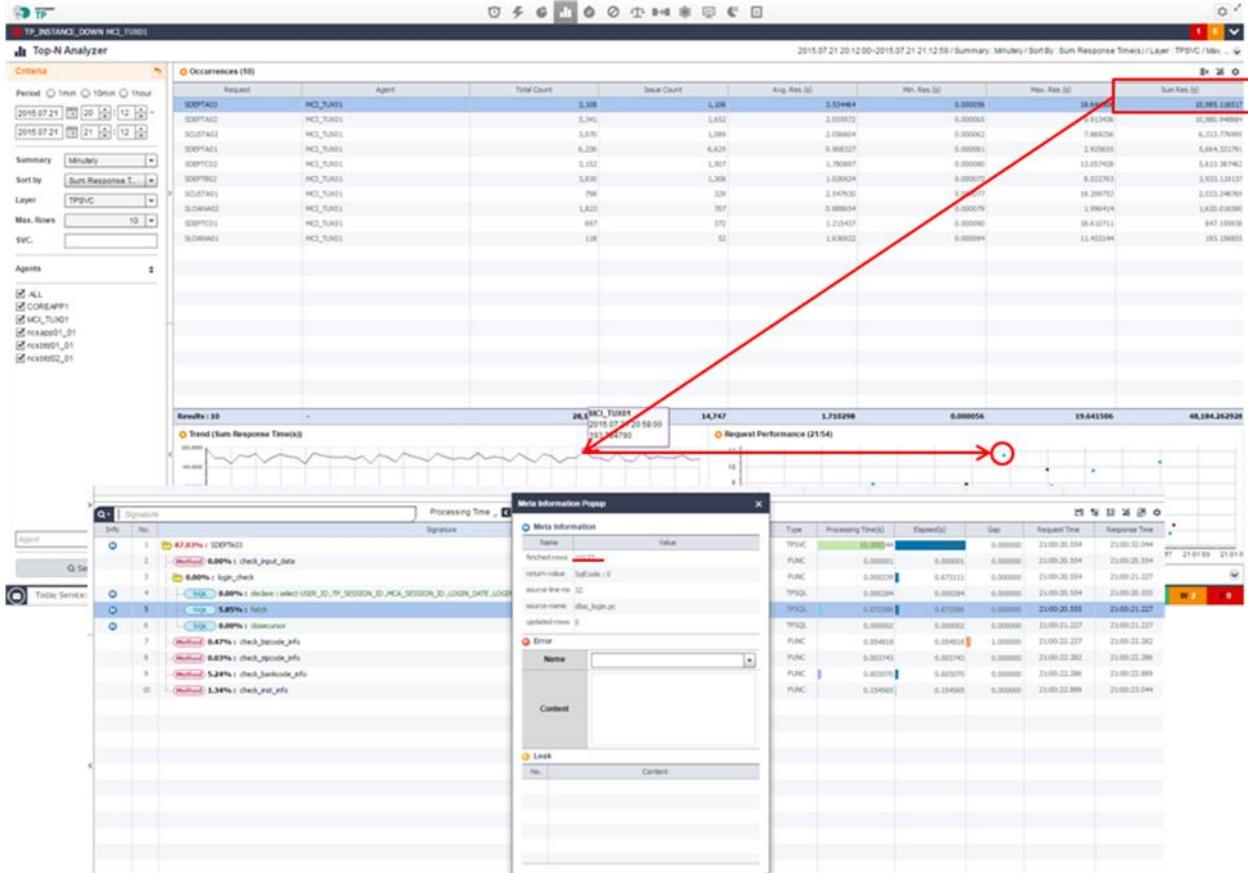
[그림] 5-3 Avg.Response Time 과 Request Analyzer 연계에 의한 Top-N 분석

5.2. Sum Response Time

조회 하고자 하는 시간대에 발생한 모든 Request에 대해 전체 처리시간의 합을 내림차순으로 정렬하여 상위 10개의 TP서비스 또는 SQL에 대하여 분석할 수 있는 데이터를 Chart-View 또는 List-View 형식의 데이터로 표현하여 제공한다.

[그림] 5-4은 TP서비스를 전체 응답시간에 대하여 산출한 Top-N 데이터를 List-View 형식으로 확인하고, Trend 차트를 통하여 전체 응답시간이 가장 오래 걸린 구간을 찾아 Request 를 확인한 후 그 중에서도 가장 오래 걸린 request를 드래그하여 Request Analyzer 연계를 통한 거래의 Call-Tree를 상세 분석하는 과정을 나타낸 그림이다.

이와 같은 과정을 통해 Request Analyzer 연계를 통한 거래의 Call-Tree 데이터에서 확인 할 수 있는 부분은 SQL fetch 처리가 많은 데이터 양으로 인해 평균응답 시간이 느려졌고, 이로 인해 해당 거래를 처리하는 TP서비스의 전체 응답시간이 많아졌다는 것을 알 수 있다.



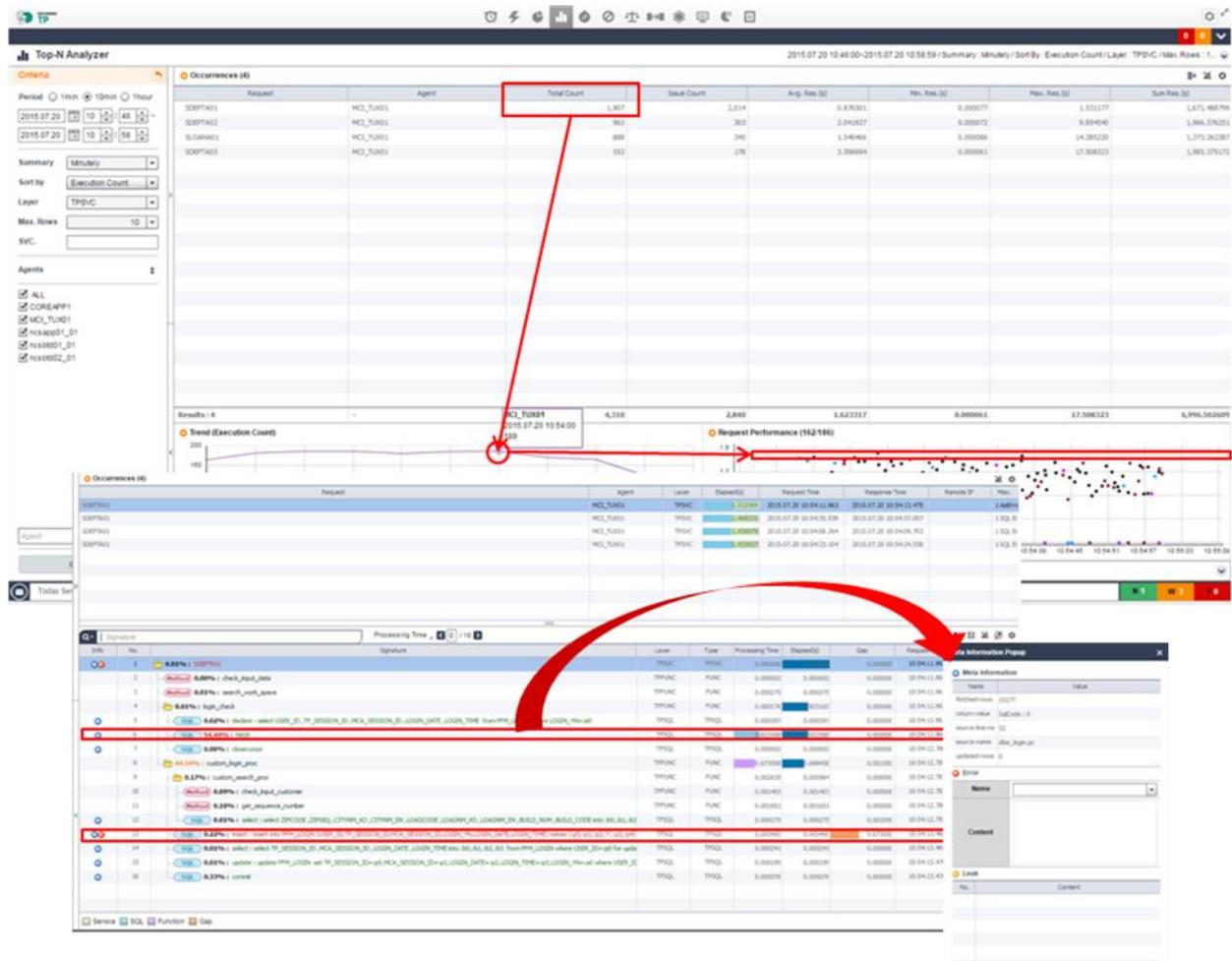
[그림] 5-4 Sum Response Time 과 Request Analyzer 연계에 의한 Top-N 분석

5.3. Execution Count

조회하고자 하는 시간대에 발생한 모든 Request에 대해 요청 건수가 많은 Request 순으로 내림차순 정렬하여 상위 10개의 TP서비스 또는 SQL에 대하여 분석할 수 있는 데이터를 Chart-View 또는 List-View 형식의 데이터로 표현하여 제공한다.

[그림] 5-5은 TP서비스의 총 수행건수에 대하여 산출한 Top-N 데이터를 List-View 형식으로 확인하고, Trend 차트를 통하여 가장 많이 호출된 구간을 찾아 Request들의 분포도를 확인하고, 그 중에 가장 오랜 시간 동안 처리 되었던 4건의 Request를 드래그하여 Request Analyzer 연계를 통한 Call-Tree 분석과정을 나타낸 그림이다.

이와 같은 과정을 통해 Request Analyzer 연계를 통한 거래의 Call-Tree 데이터에서 확인 할 수 있는 부분은 SQL처리 부분에서 2가지의 문제가 발생하였는데 첫째 fetch 건수가 많았고, 둘째 INSERT 시 Table lock으로 인한 Pending('Unique constraint violated' 에러가 발생) 이 발생하였다는 것을 알 수 있다.



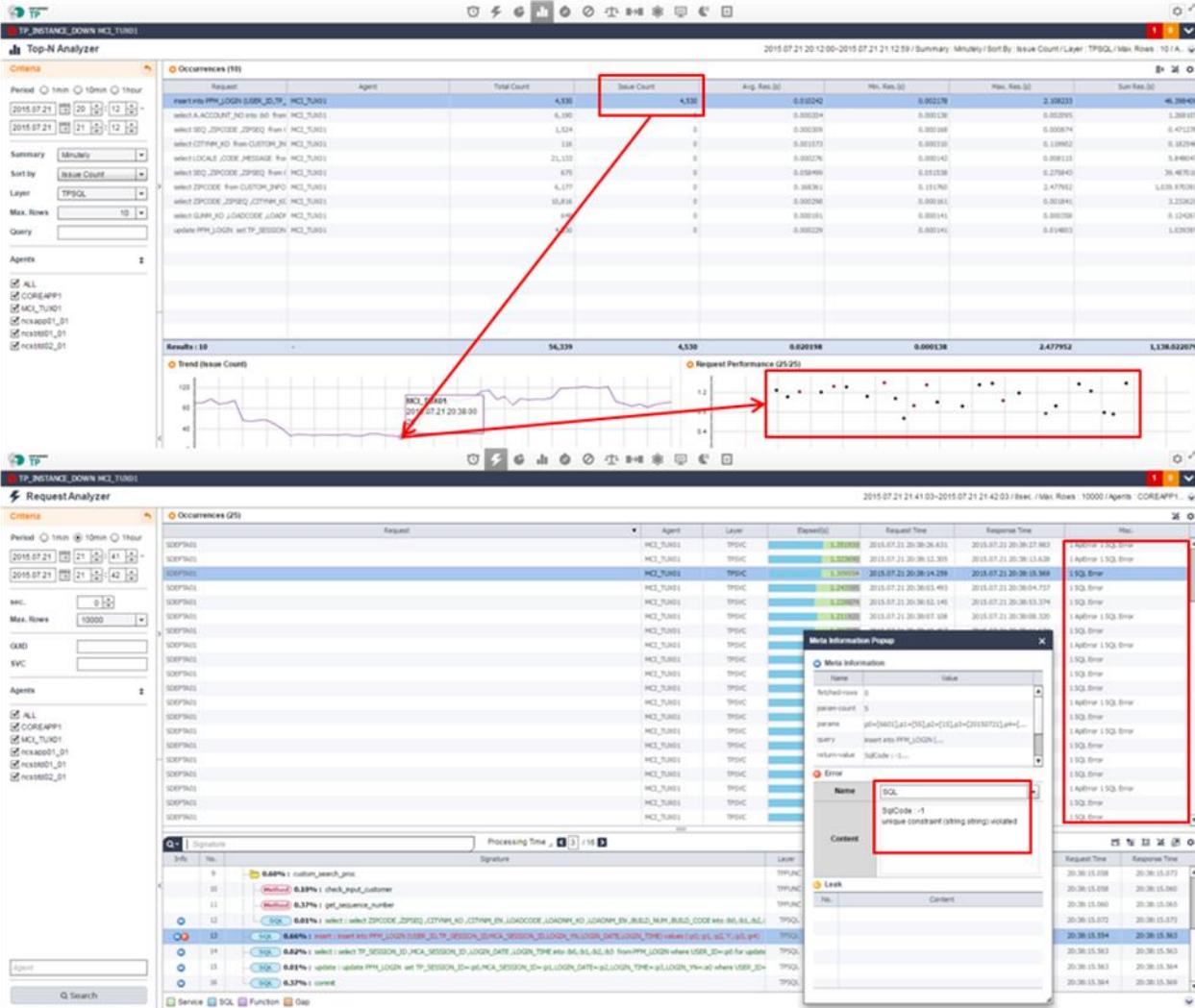
[그림] 5-5 Execution Count 과 Request Analyzer 연계에 의한 Top-N 분석

5.4. Issue Count

조회하고자 하는 시간대에 발생한 모든 Request에 대해 이슈 건수가 많은 Request 순으로 내림차순 정렬하여 상위 10개의 TP서비스 또는 SQL에 대하여 분석할 수 있는 데이터를 Chart-View 또는 List-View 형식의 데이터로 표현하여 제공한다.

[그림] 5-6은 SQL의 이슈발생 횟수에 대하여 산출한 Top-N 데이터를 List-View 형식으로 확인하고, Trend 차트를 통하여 가장 적게 호출된 구간을 찾아 Request들의 분포도를 확인하고, 그 구간에 해당되는 Request 모두를 드래그하여 Request Analyzer 연계를 통한 Call-Tree 분석과정을 나타낸 그림이다.

이와 같은 과정을 통해 Request Analyzer 연계를 통한 거래의 Call-Tree 데이터에서 확인 할 수 있는 부분은 모든 거래가 1개 이상의 에러가 발생 하였으며, 그 중 SQL 에러는 모두가 INSERT 시 'Unique constraint violated' 에러가 발생하였다는 것을 알 수 있다.



[그림] 5-6 Issue Count 과 Request Analyzer 연계에 의한 Top-N 분석

06

Issue Analyzer

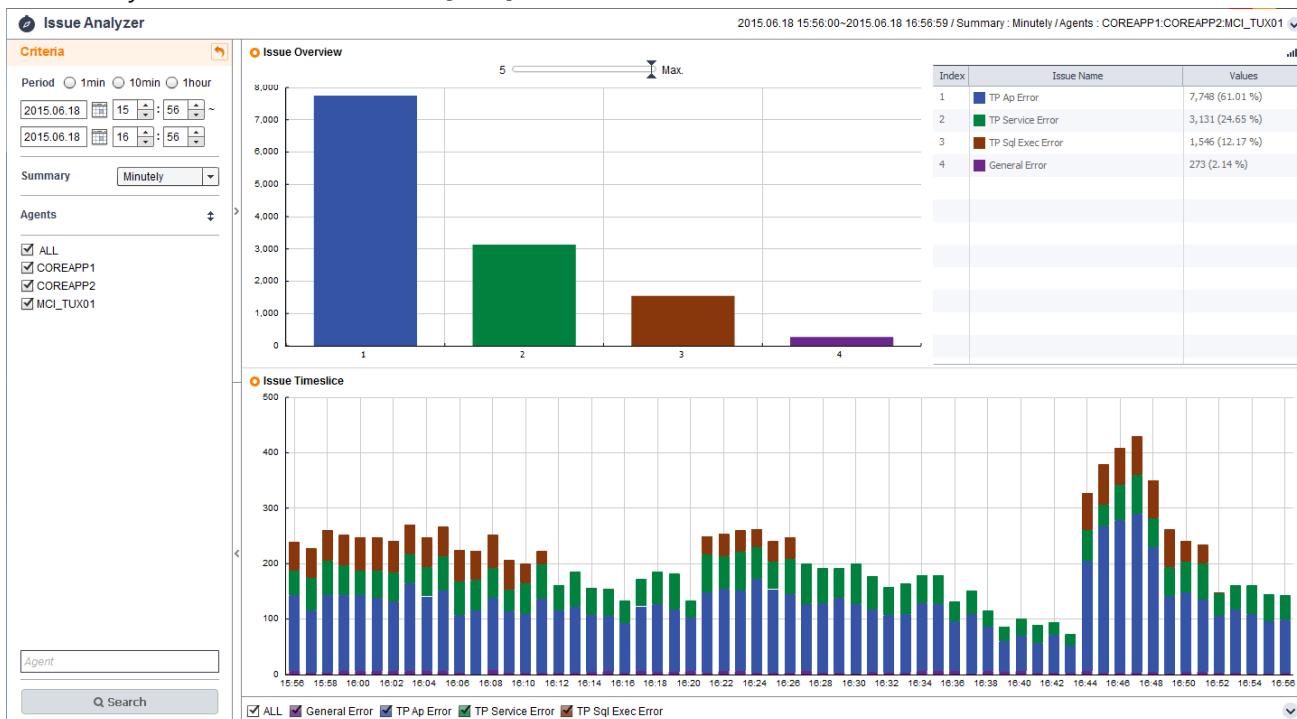
■ 6.1 TP Issue 종류

6. Issue Analyzer

Issue Analyzer 화면은 거래 처리 중 발생한 Issue에 대해 분, 시, 일 단위로 생성된 통계정보를 분석할 수 있는 화면이다. 통계 정보는 이슈 종류별로, 발생한 에이전트 별로 통계가 생성되기 때문에 어느 시스템에서 문제가 많이 발생했는지를 쉽게 파악할 수 있다.

특히 Issue Analyzer 화면은 제공되는 다른 분석 화면과 다르게 업무별로 분석할 수 있는 화면이 아니라 시스템 단위로 분석할 수 있는 화면이다.

Issue Analyzer에 대한 화면 구성은 [그림] 6-1과 같다.



[그림] 6-1 Issue Analyzer 화면 구성

6.1. TP Issue 종류

Issue Analyzer 화면의 우측에는 Issue Count가 높은 순서대로 Issue List를 제공한다. Issue의 종류에 따라 각각 다른 색상으로 표현이 되는데 어떠한 Issue가 상대적으로 많이 발생 했느냐에 따라 문제 해결방법에 대한 접근방식이 달라지게 된다.

ENPHAROS TP에서는 일반적으로 다음과 같이 5가지의 Issue를 정의하고 있다.

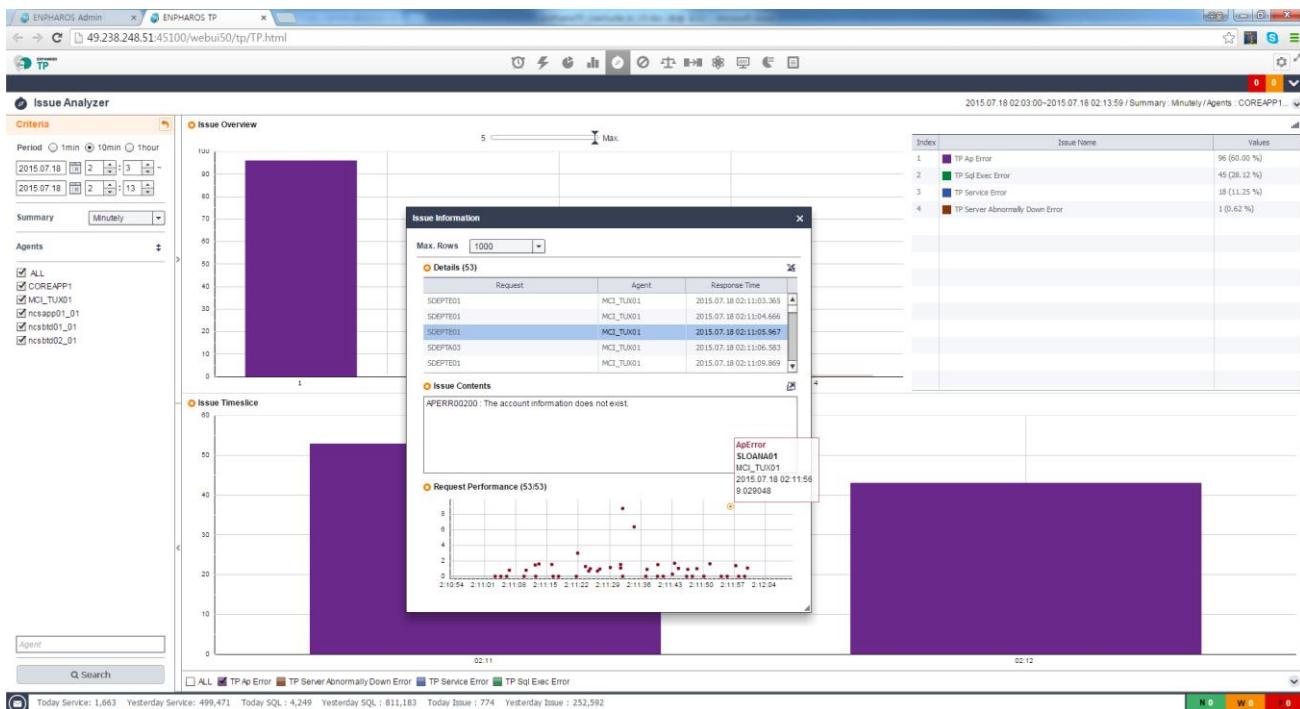
Issue 명	내용	발생위치
TP AP Error	업무로직에 의해 발생한 에러	업무로직에 의해 발생
TP Service Error	TP Monitor 또는 TP서비스에 의해 발생한 에러	TP Monitor에 의해 발생

	예) TPETIME, TPNOENT	
TP Sql Exec Error	SQL 수행 중 발생한 에러	DBMS에 의해 발생
TP Server Abnormally Down Error	Core dump가 발생하여 프로세스가 다운 되었을 때 발생하는 Issue	OS signal에 의해 발생
General Error	일반적인 함수레벨의 에러로써 그 결과가 거래의 성공 처리여부에 영향을 미치는 경우	업무로직에 의해 발생

[표] 6-1 Issue 종류

6.1.1. TP AP Error

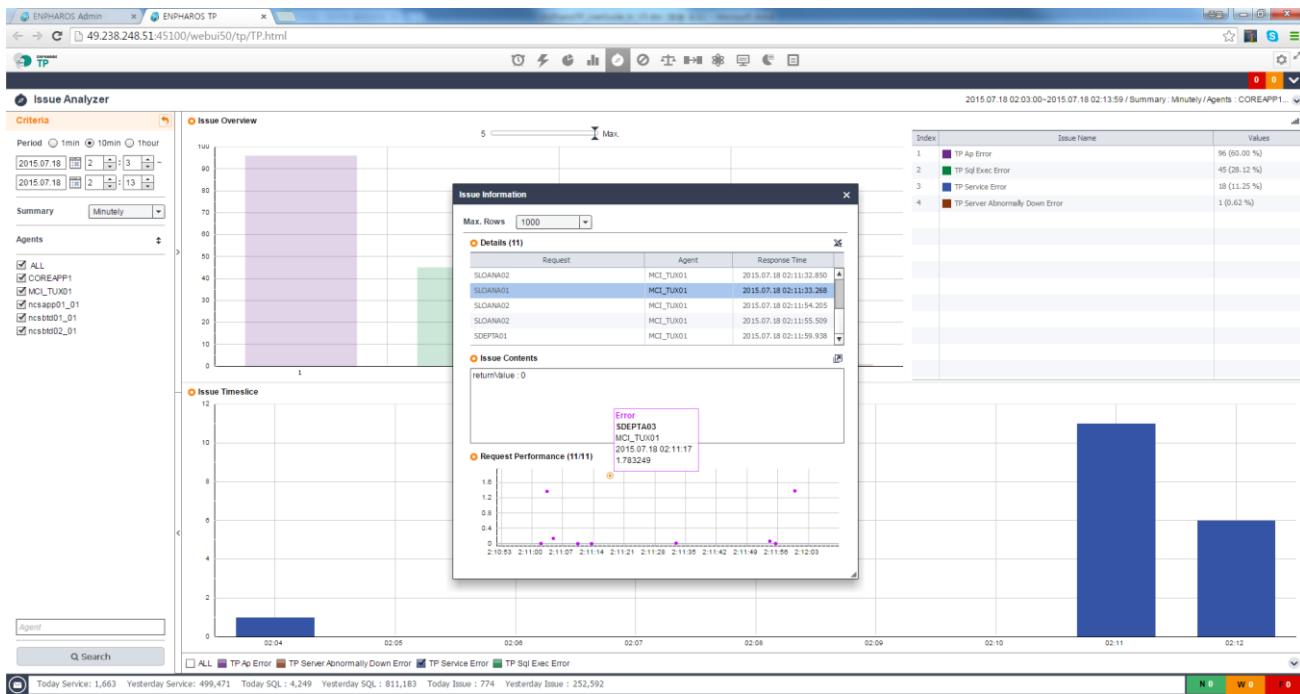
TP서비스 내의 업무로직에 의해 발생되는 에러를 'TP AP Error'로 정의한다. 처리결과 코드로 에러 발생 유무를 판단하는데, 관련 설정은 pharostp.cfg의 [app_errinfo] 라벨 항목들에 설정 항목들이 있다. 에러 판단 우선순위가 가장 높으므로, 모든 처리과정이 정상으로 처리가 되었더라도 사용자 출력전문의 에러코드 영역에 값이 설정되어 있으면 해당 거래는 에러로 판단하게 된다. 에러코드와 함께 에러메시지도 출력전문에 포함되어 있으면 그 내용도 'Issue Contents'로써 사용자에게 출력하게 된다.



[그림] 6-2 TP AP Error

6.1.2. TP Service Error

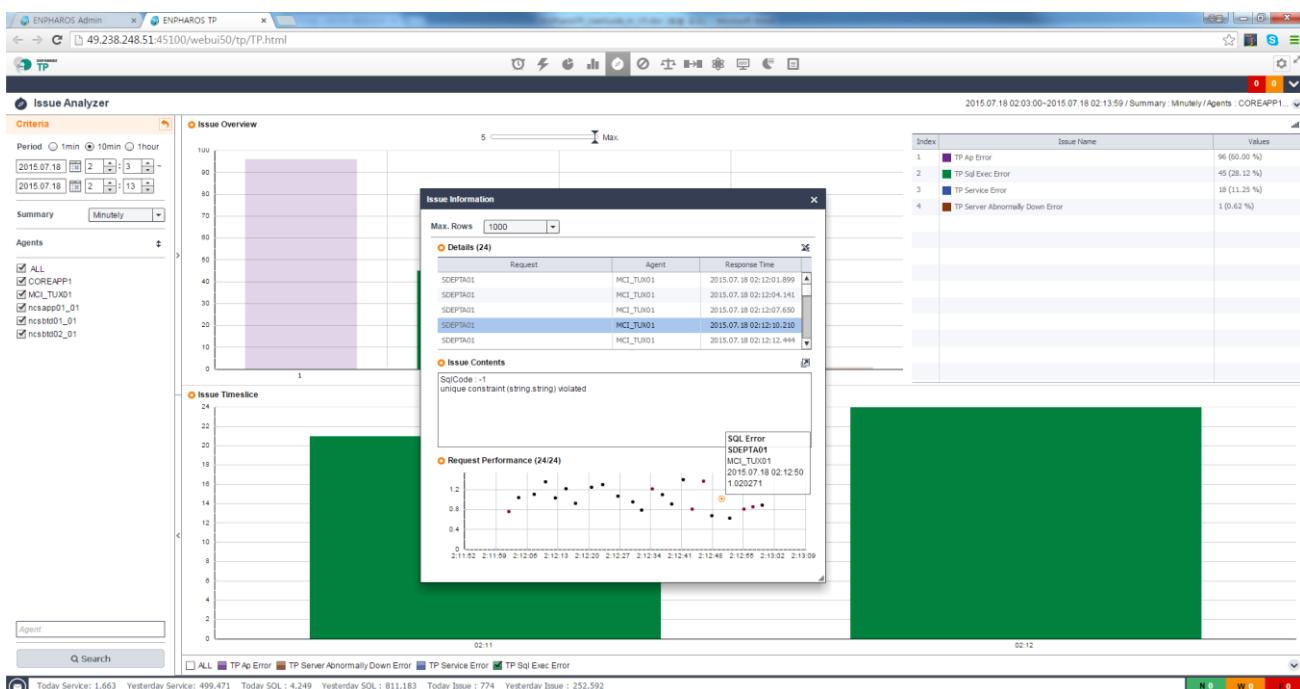
TP서비스 내의 ATMI API 호출에 대한 수행결과 값에 따른 에러로써 호출 받은 TP서비스가 'TPFAIL' 결과를 리턴 할 경우에 발생한 에러로 정의한다. 즉, TP서비스 콜 에러로 이해하면 된다. 보통 'TP AP Error'와 함께 사용되므로 우선순위가 높은 'TP AP Error'의 내용이 출력 되지만, 이마저도 없다면 단순하게 'returnValue 0'으로 표현된다.



[그림] 6-3 TP Service Error

6.1.3. TP Sql Error

TP서비스 내의 업무로직에서 SQL 문을 사용 시 DBMS에서 발생하는 에러로 정의한다. 즉, SQL 수행 후 결과코드(sql code) 값으로 'TP Sql Error'를 판단한다. 일반적으로 "Data Not Found(1403)"를 제외한 모든 sql code 값이 대상이다.(ENPHAROS TP에서 ORA-1403 코드는 에러코드로 인식하지 않음) 만약 업무적인 요건에 의해 추가적으로 에러코드에서 제외 해야 할 sql code 값이 있으면, 해당 코드 값을 pharostp.cfg 파일의 'sql.skip_err_code' 항목에 설정 해 주면 된다.

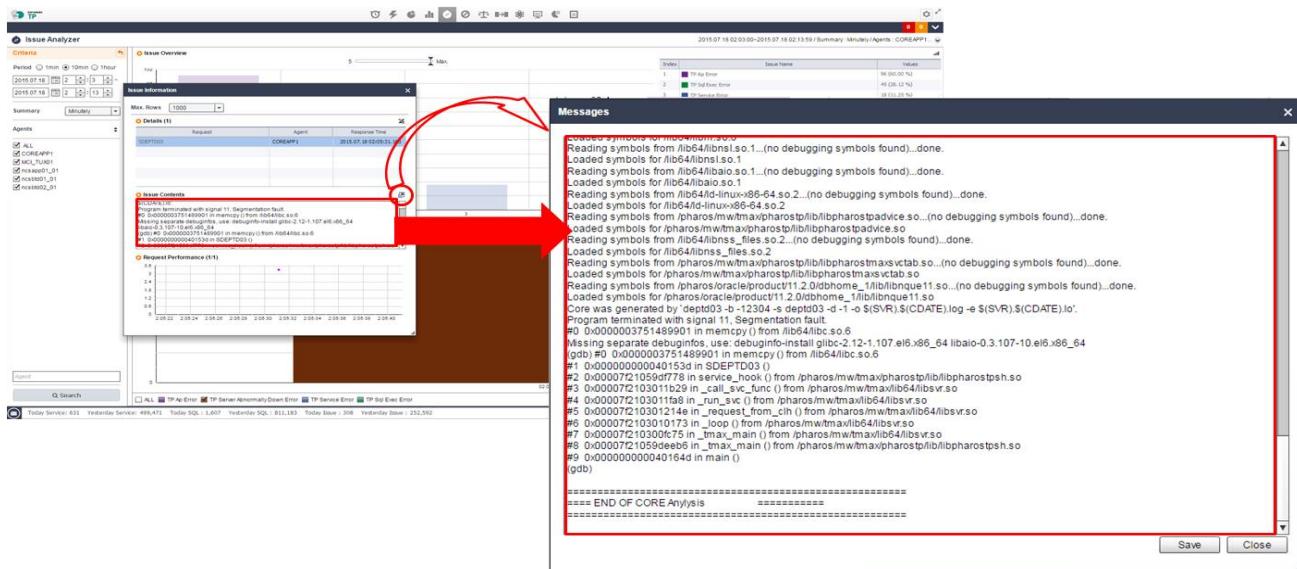


[그림] 6-4 TP Sql Error

6.1.4. TP Server Abnormally Down Error

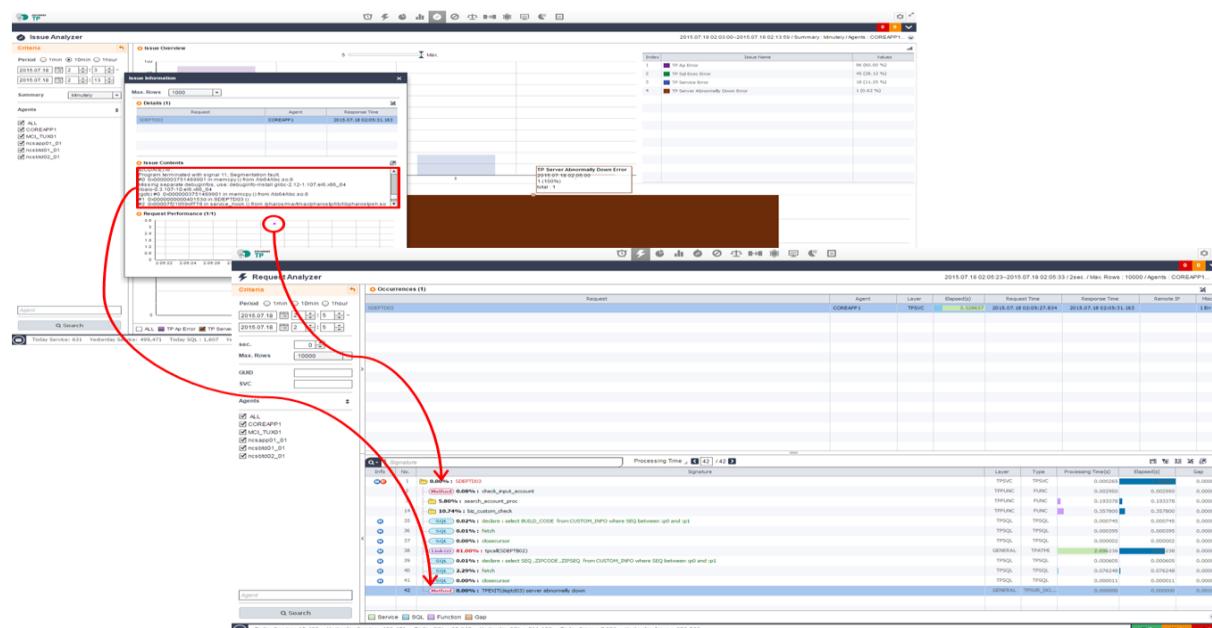
TP서비스가 처리 도중 OS로 부터 signal을 수신하여 core dump 를 발생하고 down 되는 경우에 대한 예제로 정의한다. 이러한 경우 ENPHAROS TP에서는 자동으로 core dump file을 분석하고 그 결과를 관리하는 기능이 있어 TP Monitor 콘솔에 접속하지 않고서도 또한, 어려운 디버깅 명령어를 몰라도 언제든지 관련 내용을 Issue Analyzer 화면을 통하여 확인 할 수가 있다.

관련 설정은 < *ENPharosTP_AdministrationGuide* 의 “G.Core Dump Analysis 설정 방법” > 을 참고한다.



[그림] 6-5 TP Server Abnormal Down Error

TP서비스가 Core Dump 가 발생한 경우 Request Analyzer 화면과의 연계를 통하여 해당 Request를 어디까지 처리하였고, Core Dump Analysis 메시지를 통해 원본 소스의 어느 부위에서 문제가 발생하였는지를 연계하여 한눈에 파악할 수가 있다.



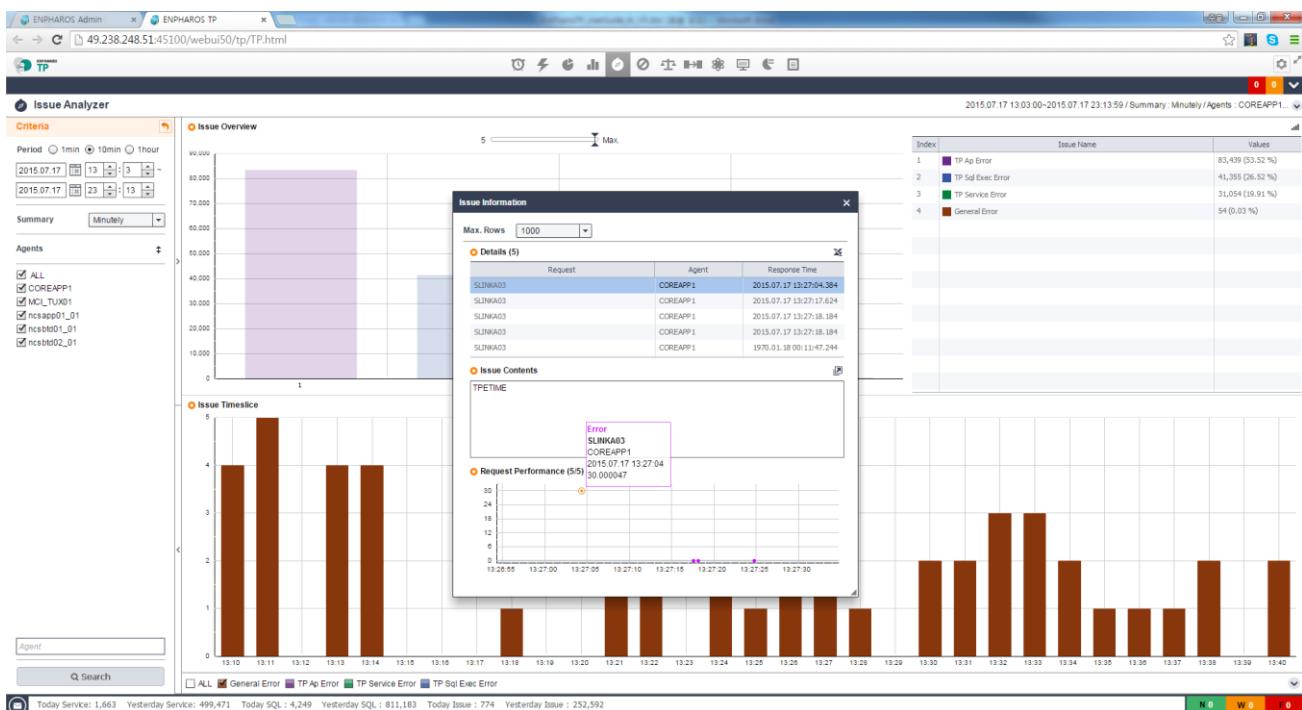
[그림] 6-6 Core Dump 발생과 Request Analyzer 연계

6.1.5. General Error

위에서 열거하지 않은 나머지 부분에 대한 에러로 정의한다. General Error로써 발생되는 대표적인 에러 유형에는 TP Monitor 엔진에서 발생하는 에러가 있다. TP Monitor 엔진에서 발생하는 에러는 보통 대상 TP 서비스로의 데이터 전달이 안 되었을 경우에 발생을 한다.

대표적인 TP Monitor 에러코드로는 다음과 같다.

- TPETIME : TP서비스 타임아웃 발생
- TPENOENT : 호출한 TP서비스가 존재하지 않음
- TPEOS : 운영 시스템에 에러가 발생
- TPESYSTEM : TP Monitor 시스템 엔진에 에러가 발생



[그림] 6-7 General Error

TP Service Error의 발생빈도가 높다면 이는 TP Monitor 시스템이 불안정한 경우이므로, 반드시 원인이 무엇인지 확인하고, 그에 대한 조치를 취해야 향후 시스템의 장애 상황을 예방할 수 있다. 특히 TP Monitor에서 제공하는 system log 파일을 확인하여 그에 알맞은 조치를 취해야 한다.

다음은 발생 빈도가 높은 TP서비스에 대한 에러와 해결 방법이다.

- TPETIME
 - 대상 서비스가 거래를 처리하기에 충분한 타임아웃 시간을 설정했는지 확인
 - 대상 서비스의 업무로직이 looping 상태이거나 hang 상태인지 확인
- TPENOENT
 - 대상 서비스가 abnormal down 이 되었는지 확인
 - 대상 서비스가 TP Monitor에 등록이 되었는지 확인

- TPESYSTEM
 - TP Monitor 엔진이 abnormal down 이 되었는지 확인

이 외에도 많은 유형의 TP Monitor 에러코드와 그에 따르는 해결방법들이 있다. 이에 대한 원인 분석 및 해결법은 각 TP Monitor 제조사의 메뉴얼을 참조한다.

07

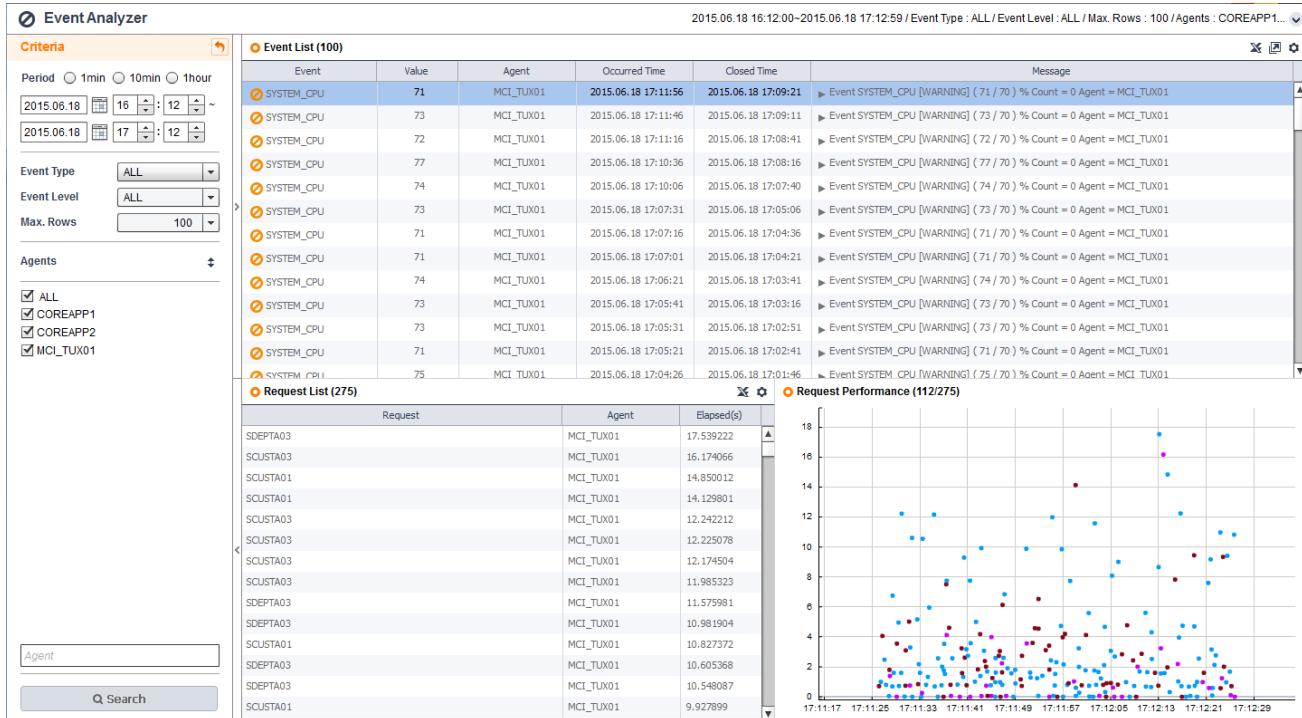
Event Analyzer

- 7.1 Event Info
 - 7.2 Request Analyzer 화면 연계
-

7. Event Analyzer

Event Analyzer는 모니터링 대상 시스템의 요청 처리 시 발생한 이벤트가 어떤 Request에 의해서 발생했는지 이벤트가 발생한 시점을 기준으로 Request 정보를 분석하여 이벤트의 원인을 분석할 수 있다.

Event Analyzer에 대한 화면 구성은 [그림] 7-1과 같다.



[그림] 7-1 Event Analyzer 화면 구성

ENPHAROS TP가 기본적으로 제공하는 이벤트 타입은 다음과 같다.

이벤트 유형	설명
SYSTEM_CPU	시스템의 CPU 사용량이 초과한 경우 발생하는 이벤트
SYSTEM_MEMORY	시스템의 메모리 사용량이 초과한 경우 발생하는 이벤트
SYSTEM_MEMORY_TYPE2	모니터링 대상 서버의 시스템 메모리 사용량이 백분율(%)로 환산하여 초과한 경우 발생하는 이벤트
INSTANCE_LIVE	TP 에이전트가 종료된 경우에 발생하는 이벤트
TP_INSTANCE_DOWN	TP Monitor 자체가 종료된 경우에 발생하는 이벤트
CORE_FILE	모니터링 대상 서버가 비정상적으로 종료된 경우에 발생하는 이벤트
TP_ALERT	모니터링 대상 서버에서 기타 에러가 발생한 경우에 발생하는 이벤트
TP_RUNNING_TIME	TP 서비스의 수행시간이 초과한 경우에 발생하는 이벤트
TP_RUNNING_서비스_COUNT	TP Active 서비스 개수가 초과한 경우에 발생하는 이벤트
TP_QUEUE_COUNT	TP Queue 적체 건수가 초과한 경우에 발생하는 이벤트
TP_SQL_RUNNING_TIME	수행되고 있는 SQL의 수행시간이 초과한 경우에 발생하는 이벤트

TP_LEAK	메모리 Leak이 발생하는 경우에 발생하는 이벤트
TP_SPR_CPU	서버 프로세스의 CPU 사용량이 초과한 경우에 발생하는 이벤트
TP_서비스_ERROR	TP 서비스에서 에러가 발생한 경우에 발생하는 이벤트
AP_FAIL_COUNT	통계 데이터에 저장된 Issue 건수가 초과한 경우 발생하는 이벤트

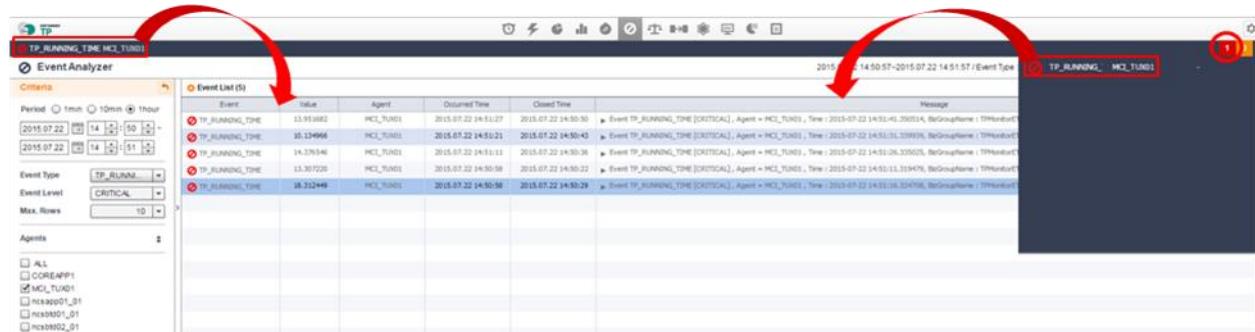
[표] 7-1 이벤트 유형별 상세 정보

7.1. Event Info

7.1.1. Event List

ENPHAROS TP 모든 화면에서 어떠한 화면으로 이동해도 바뀌지 않는 바(Bar) 영역이 3곳이 있는데, 각각 메뉴 바, 이벤트 바, 통계 바가 그것이다. 이들 중 이벤트 바에는 처리되지 못한 이벤트를 포함하여 최근에 발생한 이벤트까지 모두 관리가 되고 있고, 이를 클릭 시 Event Analyzer 와 연계가 되어, 해당 이벤트에 해당하는 이벤트 리스트가 자동선택 되어진다. 이미 처리가 된 상태의 이벤트를 검색하여 확인하기 위해서는 직접 Event Analyzer 화면으로 이동하여 검색란에 검색하고자 하는 키워드와 시간대를 입력하고 검색하면 관련 된 이벤트내역이 이벤트 리스트에 출력된다.

[그림] 7-2는 이벤트 바를 통해서 Event Analyzer 연계를 수행한 그림이다.



[그림] 7-2 이벤트 리스트

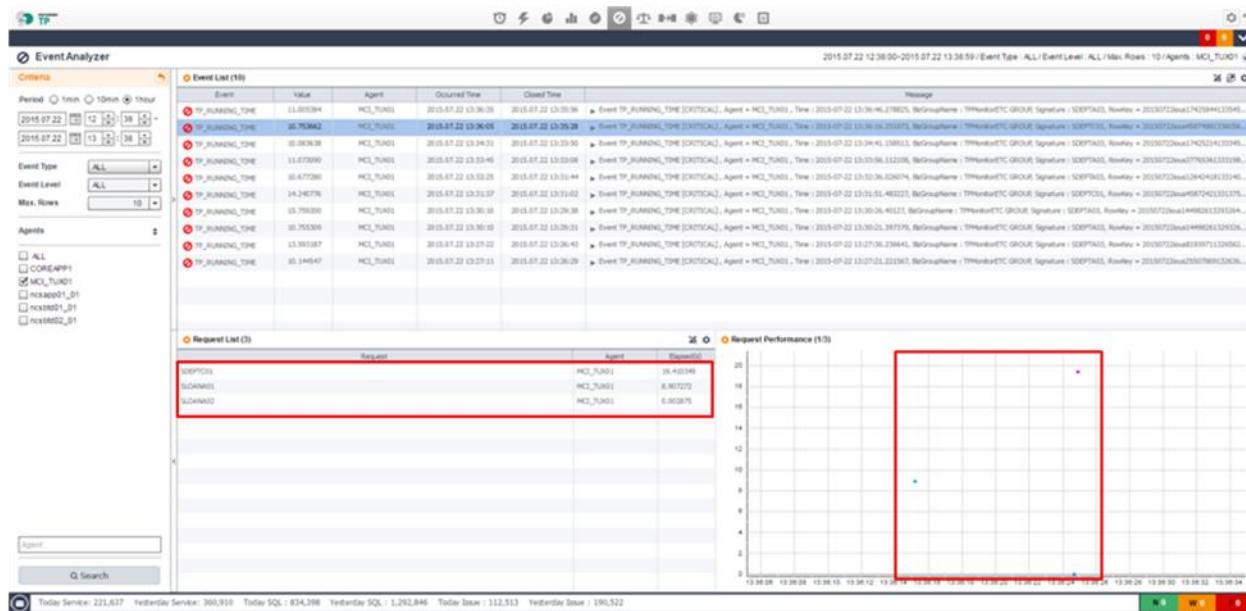
이벤트 리스트 항목 정보들 중에서 메시지부분을 잘 확인해야 한다. 메시지 부분에 관련 된 TP서비스명에 대한 정보와 왜 이벤트가 발생하였는지에 대한 상세정보들이 존재하기 때문이다.

목록	설명
Event	발생한 이벤트 명
Level	발생한 이벤트의 등급 (WARNING: 경고, CRITICAL: 심각)
Value	발생한 이벤트의 수치 값
Agent	이벤트 발생 원인
Occurred Time	이벤트 발생 시간
Closed Time	이벤트 종료 시간
Message	이벤트 발생 상세 정보

[표] 7-2 이벤트 리스트 항목 정보

7.1.2. Request List & Request Performance

이벤트 리스트를 통해 선택된 이벤트에 대한 거래정보를 확인 할 수 있는 화면이 Request List 와 Request Performance 차트이다. 이 영역에는 해당 이벤트를 발생 시켰던 거래들에 대한 요약 정보와 거래가 발생 했던 시기에 대한 내용을 점 차트를 통해 제공해준다. Request Performance 차트는 Request Analyzer 화면으로 연계된다.



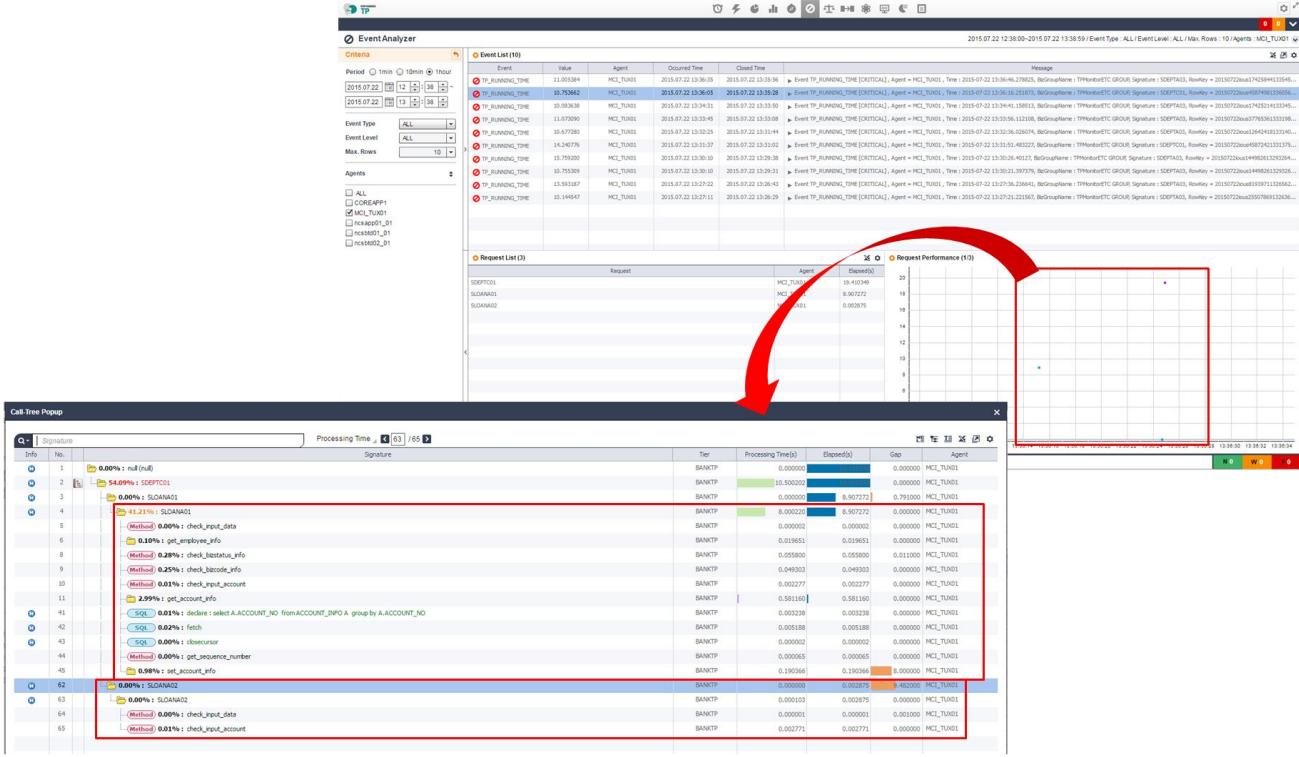
[그림] 7-3 Request List & Request Performance

7.2. Event Analyzer를 통한 상세분석

Request Performance 차트를 드래그 하여 특정 Request들을 선택하면, Request Analyzer 화면으로 전환되어 해당 Request들에 대한 상세 분석 정보를 제공한다. 이를 통해 해당 이벤트가 발생한 원인을 Request로부터 찾을 수 있으며, 상세 Call-Tree 를 통해 관련 서비스의 처리 내용 중 문제가 있는 부분의 내용 파악과 그에 대한 근본적 문제 해결이 가능하여 서비스의 튜닝 및 업무 시스템의 장애예방이 가능하다.

[그림] 7-4는 Event Analyzer 와 Request Analyzer를 이용한 원인 분석과정을 보여주는 그림이다.

모든 이벤트 탑과 모든 이벤트 레벨을 대상으로 특정 시간대에 발생했던 이벤트들을 검색한 결과 'TP_RUNNING_TIME' 이벤트가 많이 발생했다는 것을 인지하였다. 이벤트 리스트들 중 특정한 이벤트를 선택하여 'TP_RUNNING_TIME' 이벤트를 발생 시키는 TP서비스가 3개가 존재한다는 것을 찾았고, 이들 3개의 TP서비스에 대한 처리과정 및 상세성능 정보를 분석하기 위해 Request Performance 차트의 3점을 드래그하여 Request Analyzer 화면과 연계하여 분석하였다. 그 결과 TP서비스 연계를 포함하는 전체 Call-Tree 데이터를 이용한 분석을 통해 TP서비스 내에서 2번의 TP서비스 콜(tpcall)이 발생하였고, 호출 받은 각각의 TP서비스에서 모두 짧지 않은 Gap Time이 발생하였다는 것을 인지할 수 있다. 각각의 많은 시간을 소모하는 Gap Time 을 유발하는 소스코드를 찾아 Gap Time이 발생하지 않도록 수정을 한다면 TP서비스의 성능 향상과 더불어 관련 이벤트의 발생을 줄일 수 있을 것이다.



[그림] 7-4 Request Analyzer 화면 연계

08

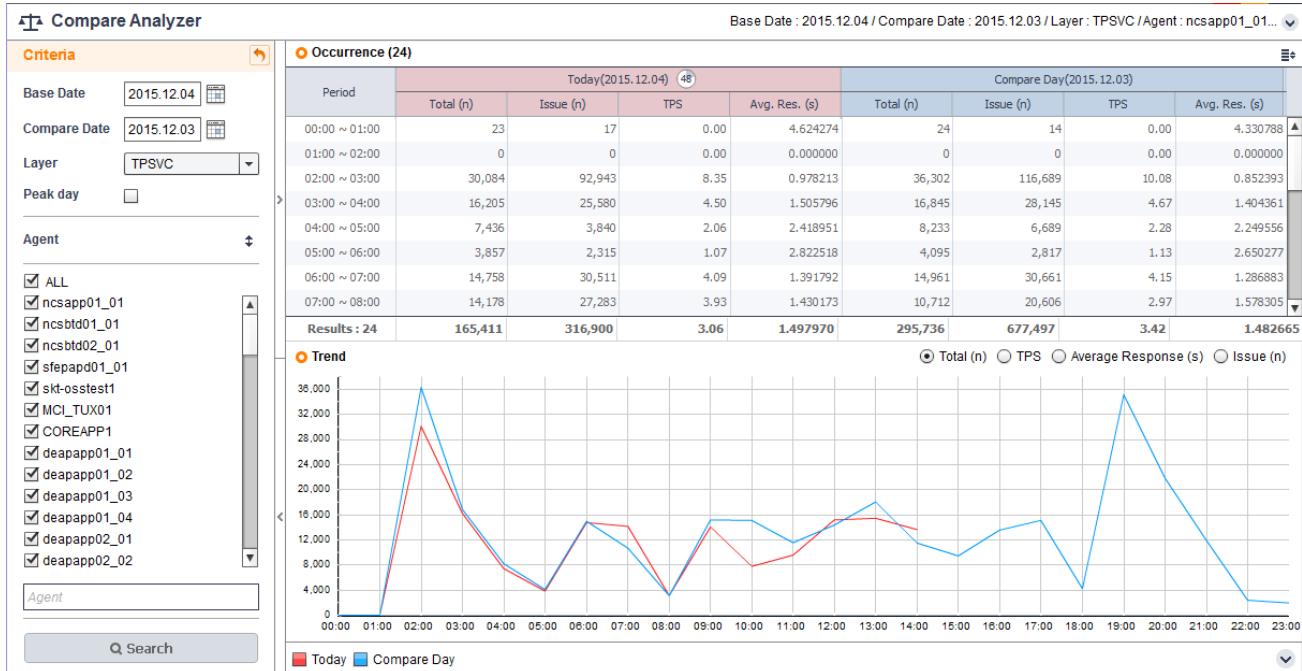
Compare Analyzer

- 8.1 Compare 리스트
 - 8.2 Trend
-

8. Compare Analyzer

Compare Analyzer는 기준일과 비교 대상일의 Request 요청 건수, TPS, Response Time을 1시간 간격으로 비교하여 보여주고, 트랜드 차트로 표현하여 처리량이 어떤 추이를 보이고 있는지 비교 분석할 수 있는 화면이다. Base Date나 Compare Date를 오늘로 선택한 경우, 1분 간격으로 갱신되어 최신 정보를 확인할 수 있다.

Compare Analyzer의 화면 구성은 [그림] 9-1과 같다.



[그림] 8-1 Compare Analyzer 화면 구성

8.1. Compare 리스트

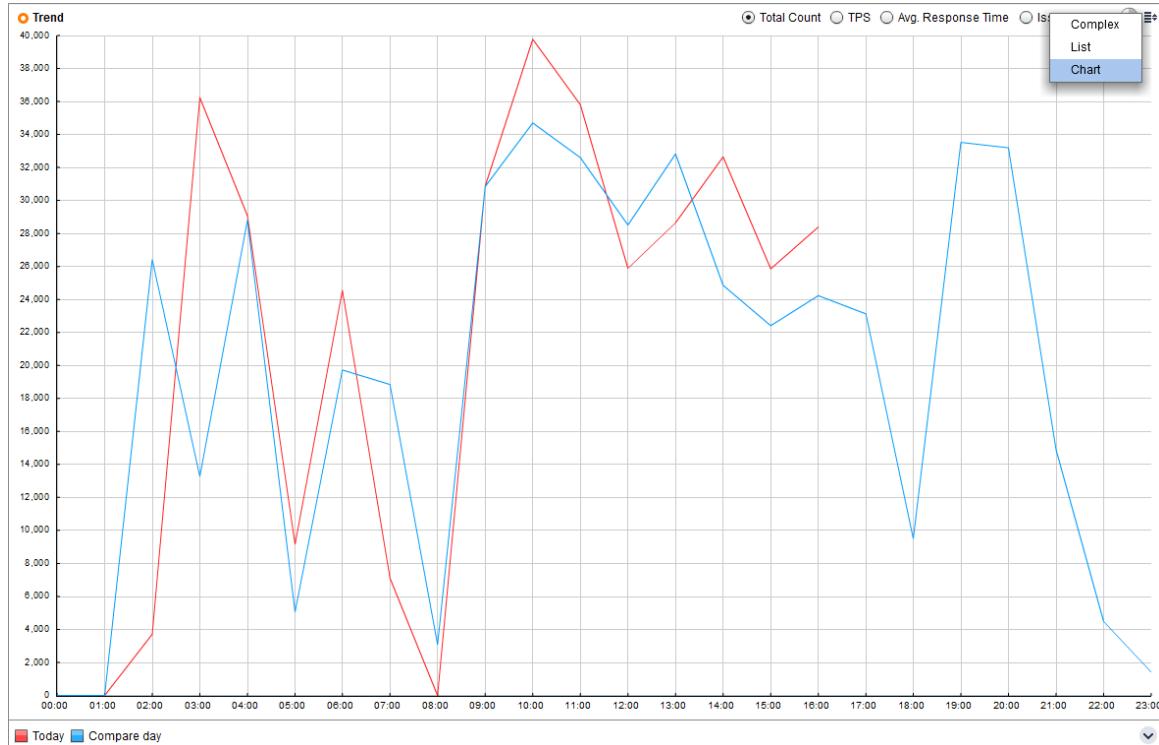
Compare 리스트는 기준일과 비교 일자의 처리건수, 실패건수, TPS, 평균 응답시간 데이터를 1시간 단위로 비교한다.

Occurrences (24)								(updated) 2015.06.18 17:05:40		
Period	Today (2015.06.18)				Compare Day (2015.06.17)				Complex	
	Total Count	Issue Count	TPS	Avg. Res.(s)	Total Count	Issue Count	TPS			
00:00 ~ 01:00	11	6	0.00	3.272178	2	1	0.00	5.512515	Complex	
01:00 ~ 02:00	0	0	0.00	0.000000	0	0	0.00	0.000000	List	
02:00 ~ 03:00	3,708	1,680	1.03	1.615022	26,429	11,902	7.34	2.078973	Chart	
03:00 ~ 04:00	36,243	15,090	10.06	1.742120	13,267	5,778	3.68	1.942869		
04:00 ~ 05:00	29,085	11,956	8.07	1.768629	28,814	12,350	8.00	2.137872		
05:00 ~ 06:00	9,150	3,373	2.54	2.545904	5,059	1,834	1.40	2.442403		
06:00 ~ 07:00	24,562	11,101	6.82	1.684483	19,731	8,888	5.48	1.937669		
07:00 ~ 08:00	7,113	3,381	1.97	1.702948	18,846	8,530	5.23	2.096918		
08:00 ~ 09:00	0	0	0.00	0.000000	3,072	1,404	0.85	1.775326		
09:00 ~ 10:00	30,875	13,606	8.57	1.819961	30,843	12,958	8.56	1.827057		
10:00 ~ 11:00	39,775	19,741	11.04	1.743774	34,699	15,723	9.63	1.832709		
11:00 ~ 12:00	35,804	16,685	9.94	1.886964	32,600	13,809	9.05	2.084978		
12:00 ~ 13:00	25,898	11,756	7.19	1.764370	28,517	12,745	7.92	2.043253		
13:00 ~ 14:00	28,648	13,134	7.95	1.548719	32,833	16,162	9.12	1.615128		
14:00 ~ 15:00	32,641	14,926	9.06	1.770116	24,857	11,182	6.90	2.013309		
15:00 ~ 16:00	25,856	12,101	7.18	1.740224	22,422	11,229	6.22	1.649432		
16:00 ~ 17:00	28,402	12,116	7.88	1.921170	24,243	10,914	6.73	1.779437		
17:00 ~ 18:00	-	-	-	-	23,135	10,205	6.42	1.903266		
18:00 ~ 19:00	-	-	-	-	9,495	4,126	2.63	2.271392		
19:00 ~ 20:00	-	-	-	-	33,533	14,635	9.31	1.570388		
20:00 ~ 21:00	-	-	-	-	33,205	15,916	9.22	1.829534		
21:00 ~ 22:00	-	-	-	-	14,902	6,539	4.13	2.230849		
22:00 ~ 23:00	-	-	-	-	4,488	2,043	1.24	2.197121		
23:00 ~ 24:00	-	-	-	-	1,417	588	0.39	2.536260		
Results : 24		357,771	160,652	99.37	1.782978	466,409	209,561	129.54	1.914586	
<input checked="" type="checkbox"/> Today <input type="checkbox"/> Compare day										

[그림] 8-2 Compare 리스트

8.2. Trend

트랜드는 시간대별 기준일과 비교대상일 간의 차트를 선형그래프로 비교한다. 기준일 데이터는 빨간색, 비교날짜의 데이터는 파란색으로 표현한다.



[그림] 8-3 Compare 트랜드 차트

9

Caller Callee Analyzer

9. Caller Callee Analyzer

일반적으로 서비스들간의 호출관계는 관련 프로그램의 소스 내부를 직접 확인을 해야 알 수 있고, 이마저도 호출대상 서비스 명이 하드-코딩이 되어 있는 경우에만 확인이 가능하다. Caller Callee Analyzer 화면은 프로그램의 소스확인 없이도 어떠한 서비스가 어떠한 서비스를 총 몇 번 호출하였고, 호출결과에 대한 이슈발생 횟수를 얼마인지를 호출 시간대 별로 확인 할 수 있도록 제공하는 분석화면이다.

이 분석화면을 통하여 문제가 있는 서비스에 대한 영향도를 실시간으로 파악할 수 있다.

Criteria		Occurrences (83)			
Request Time	Caller Service	Callee Service	Count	Issue Count	Sum Res.(s)
2015.06.28 ~ 2015.06.29	SDEPTC02	SLOANA02	705	342	2,176.021423
	SDEPTC02	SCUSTA02	1	1	11.873165
	SDEPTC02	SLOANA02	2,242	1,069	6,838.817813
	SDEPTB01	SDEPTA03	1	0	1.761252
	SDEPTB03	SDEPTA02	1	0	11.024097
	SDEPTC01	SLOANA01	1	0	1.603188
	SDEPTC02	SLOANA02	1	0	2.600528
	SDEPTC02	SLOANA02	1,779	859	5,619.723036
	SDEPTE02	SCUSTA02	1	1	3.597695
	SDEPTB01	SDEPTA03	149	61	828.002694
	SDEPTB02	SCUSTA02	383	496	878.180999
	SDEPTB03	SCUSTA01	156	67	92.273193
	SDEPTB03	SDEPTA02	158	56	579.022277
	SDEPTC01	SCUSTA21	95	135	474.972667
	SDEPTC01	SLOANA01	166	129	772.040001
	SDEPTC01	SLOANA02	67	64	421.166522
	SDEPTC02	SLOANA02	233	123	804.254088
	SDEPTC02	SLOANA02	2,201	1,039	7,045.475632
	SDEPTE02	SCUSTA02	177	220	445.056482
	SDEPTE03	SCUSTA03	2	3	20.432909
	SLOANB01	SCUSTA21	115	171	587.739830
	SLOANB01	SLOANA02	72	79	464.649451
	SLOANB01	SLOANA01	188	166	916.325156
	SLOANB02	SLOANA02	325	171	1,232.065726
	SDEPTB01	SDEPTA03	6	2	33.333208
	SDEPTB02	SCUSTA02	11	5	42.275392

[그림] 9-1 Caller Callee Analyzer 화면 구성

10

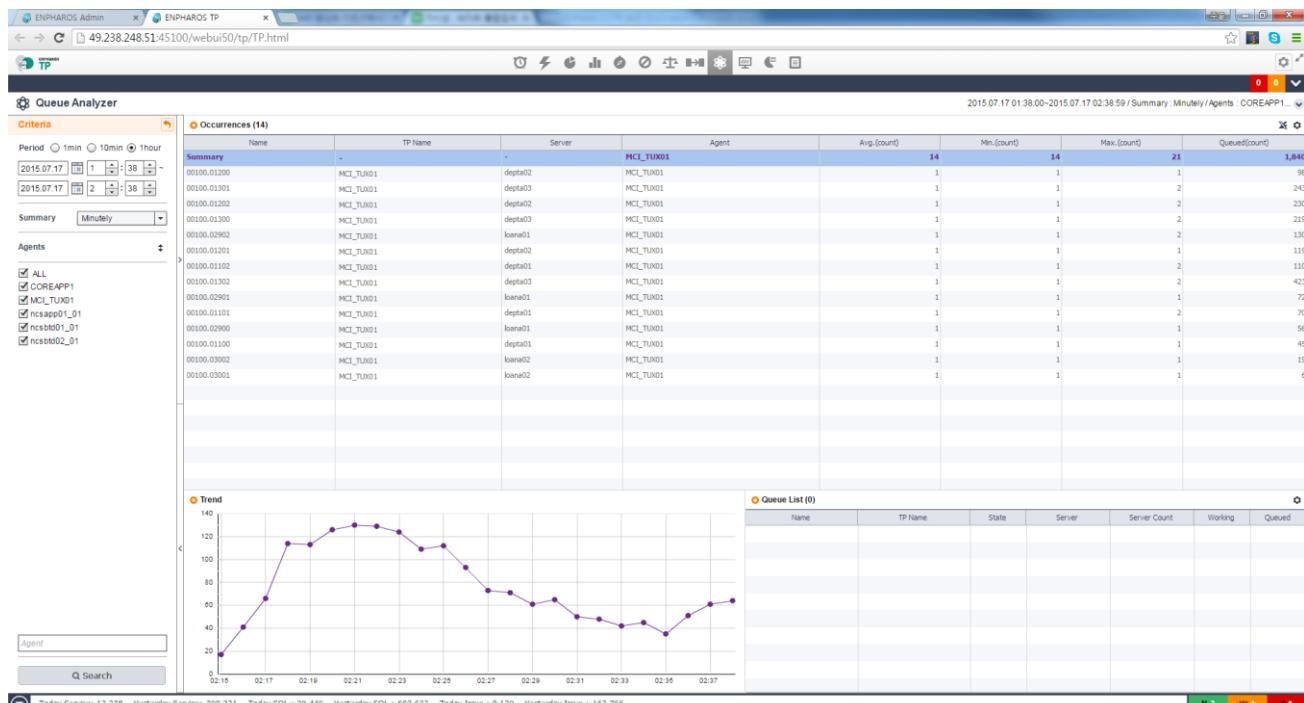
Queue Analyzer

- 10.1 Tuxedo Queue List
 - 10.2 Tmax Queue List
-

10. Queue Analyzer

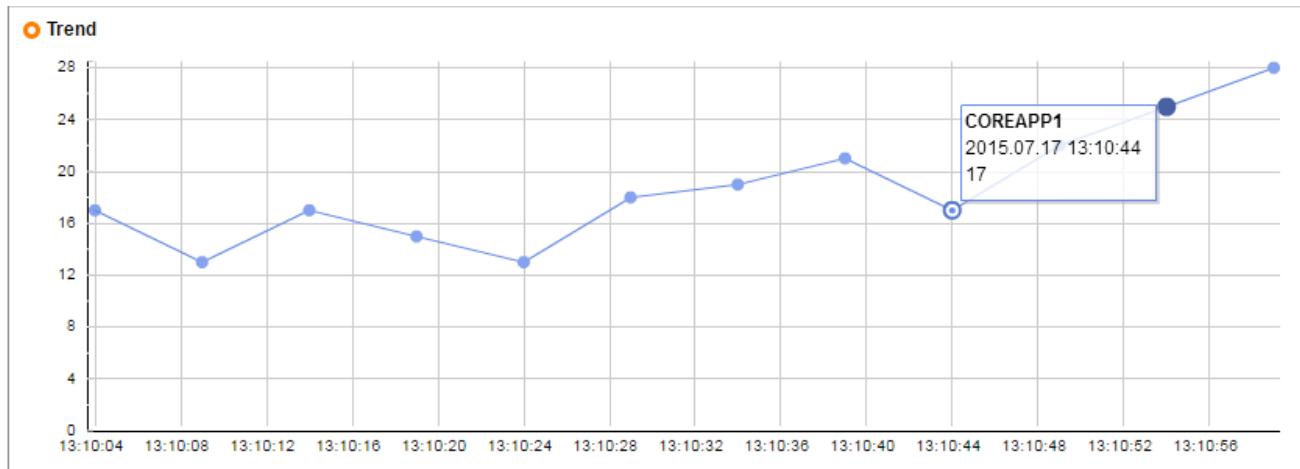
특정 TP서비스에 대하여 많은 거래부하가 발생하거나, 특정 TP서비스의 성능저하가 발생하였을 경우에 처리되지 못한 데이터는 TP서비스 큐에 쌓이게 되고, 이는 곧 업무처리의 지연효과로 이어지게 된다. 이러한 문제들을 해결 또는 방지하고자 사용하는 화면이 Queue Analyzer 화면이다. Queue Analyzer 화면은 TP서비스에 대한 큐 상태를 시간대별로 분석할 수 있는 화면으로 각 시간대별로 가장 많이 큐를 점유한 서비스목록과 각 TP서비스가 최대 몇 개까지 큐에 데이터를 적재했었는지를 파악할 수가 있고, 이를 통하여 기본적인 TP Monitor 서비스 구성에 대한 튜닝의 단초를 제공해 준다.

일반적인 TP Queue Analyzer에 대한 화면 구성은 [그림] 10-1과 같다.



[그림] 10-1 TP Queue Analyzer 화면 구성

Queue Analyzer 화면에서 제공하는 트랜드 차트는 검색조건에 따른 일/시/분 단위의 그래프가 출력되는 영역이다. 그래프에 마우스를 위치하면 해당 시간대에 발생한 에이전트 명, 평균 처리시간, 큐 건수를 확인 할 수 있다. 차트에서 해당 시간의 점을 클릭하면 5초 단위의 트랜드 그래프를 표시하며 해당 시점의 Queue 리스트를 제공한다.



[그림] 10-2 TP Queue Analyzer Trend

10.1. Tuxedo Queue List

Tuxedo에서의 큐 리스트에 대한 내용은 [그림]10-3과 같이 표현이 된다. Name 항목의 값은 Tuxedo에서 내부적으로 사용되는 큐 명이 표현이 되고, State 항목에는 "Active"로 표현이 된다.

Name	TP Name	State	Server	Server Count	Working	Queued
00100.01302	MCI_TUX01	Active	depta03	1	0	4
00100.01301	MCI_TUX01	Active	depta03	1	0	3
00100.01300	MCI_TUX01	Active	depta03	1	0	3
00100.01202	MCI_TUX01	Active	depta02	1	0	1
00100.01201	MCI_TUX01	Active	depta02	1	0	1
00100.01200	MCI_TUX01	Active	depta02	1	0	1

[그림] 10-3 Tuxedo Queue List

Tuxedo에서는 큐 정보를 가져올 때 Tuxedo에서 제공하는 관리 유틸리티인 tmadmin 을 이용하여 가져온다. 일반적으로 Tuxedo 환경에서 ENPHAROS TP를 구축할 때 보안상의 이유로 TP Monitor계정과는 별도의 성능관리 계정을 생성하여 구축하는 경우가 있다. 이때 성능관리 계정으로는 Tuxedo 유틸리티인 tmadmin 을 실행권한 문제로 인해 실행할 수가 없는 경우가 있기 때문에 ENPHAROS TP 에서는 큐 파일을 이용하여 관련 정보를 가져오는 방법을 제공한다.

관련된 자세한 내용은 <*ENPharosTP_AdministrationGuide* 의 "3.2.5 TUXEDO 어드민 모니터링 정보"> 부분을 참고한다.

참고

이들 값들은 TP Monitor 계정의 콘솔에서 다음의 명령어로 확인이 가능하다.

```
#> echo printqueue | tmadmin -r
```

10.2. Tmax Queue List

Tmax에서의 큐 리스트에 대한 내용은 [그림]10-4과 같이 표현이 된다. Name 항목의 값은 Tmax에서 내부적으로 사용되는 큐 명이 표현이 되는데 이는 곧 Tmax 서버명이 된다. State 항목에는 Tuxedo의 State 와는 다르게 다양한 값들이 올 수가 있다. 보통 평상시에는 "RDY" 로 표현이 되고, Tmax 서비스의 상태에 따라 NRDY, RUN, Unregistered, BLK, PBLK, UNADV, PUNADV 가 될 수 있다.

Queue List (6)						
Name	TP Name	State	Server	Server Count	Working	Queued
prod01	COREAPP1	RDY	prod01	1	0	7
order02	COREAPP1	RDY	order02	1	0	1
linka05	COREAPP1	RDY	linka05	1	0	6
linka03	COREAPP1	RDY	linka03	1	0	3
depta03	COREAPP1	RDY	depta03	3	0	3
comon02	COREAPP1	RDY	comon02	1	0	5

[그림] 10-4 Tmax Queue List

참고

이들 값들은 TP Monitor 계정의 콘솔에서 다음의 명령어로 확인이 가능하다.

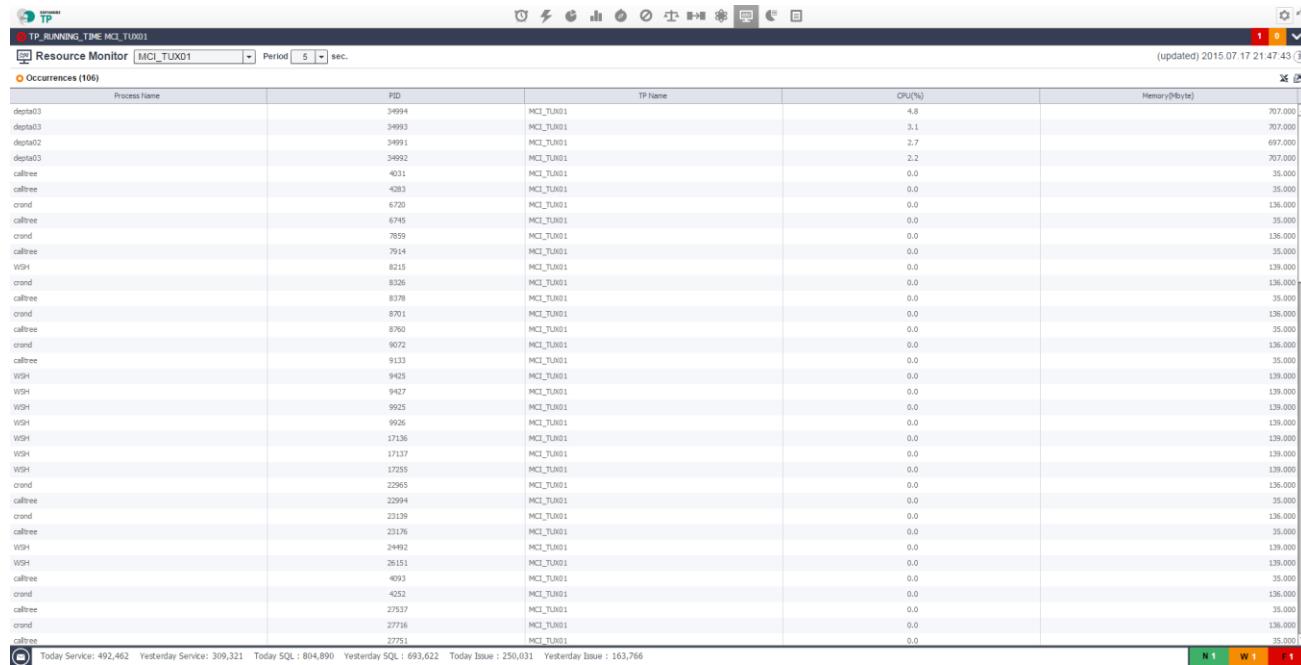
```
#> echo "st -s" | tmadmin
```

11

Resource Monitor

11. Resource Monitor

TP서비스를 포함한 TP Monitor 계정으로 구동된 모든 프로세스에 대하여 각각의 프로세스에 대한 시스템 자원 사용률을 매 5초 간격으로 자동 갱신하여 산출하여 제공하는 화면이다. 시스템 자원의 주요 항목은 CPU 사용률과 메모리 사용률로써 실시간으로 어떠한 프로세스의 자원사용률이 높은지를 프로세스 별로 비교분석 할 수 있다.



[그림] 11-1 Resource Monitor 화면구성

참고

이 기능을 활성화 하기 위해서는 TP 에이전트의 환경파일 pharostp.cfg에서 아래의 항목에 설정이 되어 있어야 한다.

TP Monitor.on.username=XXXX (TP Monitor 계정명)

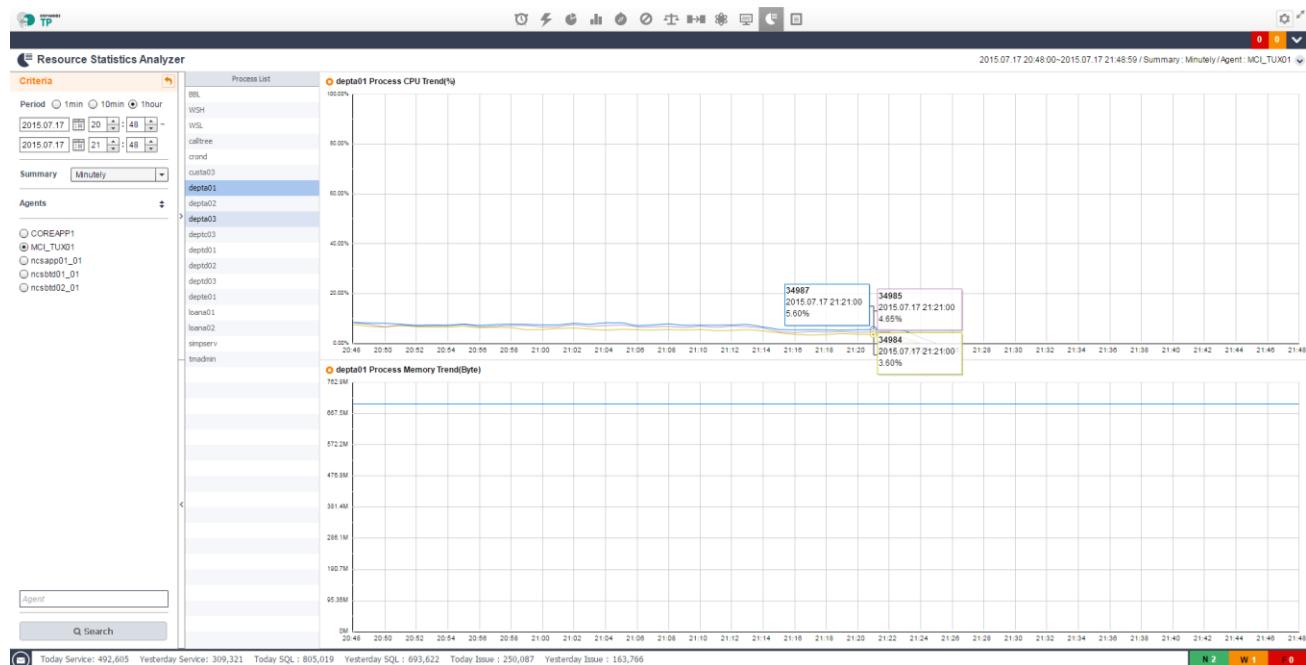
mon.proc_info_flag=true (true: 사용, false: 미사용)

12

Resource Statistics Analyzer

12. Resource Statistics Analyzer

TP서비스를 포함한 TP Monitor 계정으로 구동된 모든 프로세스에 대하여 각각의 프로세스에 대한 시스템 자원 사용률에 대한 추이를 각각의 프로세스 별로 시간대에 맞춰 표현한 화면이다. 특정 시간대 동안 TP 서비스를 포함한 프로세스가 시스템의 자원(CPU, 메모리)을 얼마나 많이 사용하여 수행 했는지에 대한 그림을 확인함으로써 해당 프로세스가 문제가 있었는지에 대한 유추를 할 수가 있다.



[그림] 12-1 Tmax Queue List

예를 든다면, 다음과 같은 기준으로 프로세스의 상태를 진단 할 수 있다.

- CPU 사용률이 0 이면서 메모리 사용률에는 변화가 없다. → 프로세스가 idle 상태
- CPU 사용률이 0 이면서 메모리 사용률이 0 → 프로세스 다운
- CPU 사용률이 일정 하면서 메모리 사용률이 증가 → 메모리 leakage 발생
- CPU 사용률이 계속적으로 100% → 프로세스 looping 상태
- CPU 사용률이 불규칙적으로 높고 낮음이 계속된다. → 거래 부하가 높은 상태

참고

이 기능을 활성화 하기 위해서는 TP 에이전트의 환경파일 pharostp.cfg에서 아래의 항목에 설정이 되어 있어야 한다.

TP Monitor.on_tp_username=XXXX (TP Monitor 계정명)
mon.proc_info_flag=true (true: 사용, false: 미사용)

13

System Log Analyzer

- 13.1 Tuxedo ULOG
 - 13.2 Tmax slog
-

13. System Log Analyzer

TP Monitor에서는 TP서비스 및 TP Monitor 엔진레벨에서 발생한 모든 이벤트들을 TP Monitor Log에 기록을 한다. Tuxedo에서는 이를 ULOG라 하고, Tmax에서는 slog라 한다. ENPHAROS TP는 ULOG, slog 모두 지원을 하며 이들 로그파일에 발생되는 각종 이벤트 데이터를 중요도에 따라 필터링하여 System Log Analyzer 화면으로 제공한다.

이를 통하여 사용자는 TP Monitor의 중요 로그 데이터를 확인하기 위해 TP Monitor 시스템에 접속할 필요가 없이 본 화면을 통하여 언제든지 관련 내용을 확인 할 수가 있다. 로그 데이터에 대한 폴링주기는 pharostp.cfg 파일에 정의되어 있으며 디폴트 5초 주기로 새로운 이벤트 발생 여부를 감지하여 출력하게 된다. 또한 특정 시간대에 대한 이벤트들을 문자열 비교로 관심있는 내용만 필터링하여 확인 할 수 있다.

The screenshot shows the System Log Analyzer application window. On the left, there is a sidebar with a 'Criteria' section containing fields for 'Period' (set to 1min), 'Log Level' (set to ALL), 'Process Name', 'SVC. Code', 'MSG. Code', and 'Max. Rows' (set to 10). Below this is an 'Agents' section with checkboxes for 'COREAPP01', 'MCL_TUX01' (which is checked), and three other unselected options. At the bottom of the sidebar are 'Agent' and 'Q. Search' buttons. The main area is titled 'System Log (10)' and contains a table with the following data:

TP Name	Occurred Time	Agent	Process Name	Log Level	SVC. Code	Msg. Code	Log Message
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1027	Unable to connect to SHH
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1027	Unable to connect to SHH
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1020	Unable to obtain authentication level
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1055	Unable to establish WSL connection
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1027	Unable to connect to SHH
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1055	Unable to establish WSL connection
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1027	Unable to connect to SHH
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1027	Unable to connect to SHH
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1027	Unable to connect to SHH
MCL_TUX01	2015.07.17 02:33:01		'proc	D	LIBWSC_CAT	1055	Unable to establish WSL connection

At the bottom of the main window, there is a status bar with the following information: Today Service: 493,967 Yesterday Service: 309,321 Today SQL : 806,249 Yesterday SQL : 693,622 Today Issue : 250,574 Yesterday Issue : 163,766. To the right of the status bar are three colored buttons: green (N 7), yellow (W 1), and red (F 0).

[그림] 13-1 System Log Analyzer

13.1. Tuxedo ULOG

Tuxedo ULOG 파일의 포맷은 “시간:호스트명:프로세스명:PID:쓰레드ID:에러코드:에러번호:메시지Type:메시지”으로 구성되어 있다. 이를 ENPHAROS TP에서 파싱 및 필터링하여 [그림]13-2 Tuxedo ULOG 와 같은 내용으로 사용자에게 출력한다.

Tuxedo ULOG에서 중요한 항목은 Svc.Code 와 Msg.code 의 값이다. 두 개의 코드 값은 Tuxedo에서 정의한 에러코드 값으로 해당 메시지의 발생 원인과 해결 방법을 Tuxedo 매뉴얼을 통해 알 수 있다.

System Log (10)							
TP Name	Occurred Time	Agent	Process Name	Log Level	SVC. Code	Msg. Code	Log Message
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1027	Unable to connect to SSH
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1027	Unable to connect to SSH
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1020	Unable to obtain authentication level
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1055	Unable to establish WSL connection
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1027	Unable to connect to SSH
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1055	Unable to establish WSL connection
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1027	Unable to connect to SSH
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1027	Unable to connect to SSH
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1027	Unable to connect to SSH
MCI_TUX01	2015.07.17 02:33:01	MCI_TUX01	?proc	D	LIBWSC_CAT	1055	Unable to establish WSL connection

[그림] 13-2 Tuxedo ULOG

참고

이 기능을 활성화 하기 위해서는 TP 에이전트의 환경파일 pharostp.cfg에서 아래의 항목에 설정이 되어 있어야 한다.

log.sys_log_path=/tuxedo/log/ULOG.\$(MMDDYY) (ULOG 파일의 절대경로)

log.sys_log_pattern=ERROR|WARN

13.2. Tmax slog

Tmax slog 파일의 포맷은 “프로그램명:시간:메시지Type:에러코드:메시지”로 구성되어 있다. 이를 ENPHAROS TP에서 파싱 및 필터링하여 [그림]13-3 Tmax slog 와 같은 내용으로 사용자에게 출력한다.

Tmax slog에서 중요한 항목은 Svc.Code 와 Msg.code 의 값이다. 두 개의 코드 값은 Tmax에서 정의한 에러코드 값으로 해당 메시지의 발생 원인과 해결 방법을 Tmax 매뉴얼을 통해 알 수 있다.

System Log (10)							
TP Name	Occurred Time	Agent	Process Name	Log Level	SVC. Code	Msg. Code	Log Message
COREAPP1	2015.07.17 13:40:46	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x8081) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:48	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x03F) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:48	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x4053) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:46	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x108) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:46	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x404) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:50	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x041) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:44	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x8070) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:44	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x406) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:40	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x9C) connection : svc = SLINKA05
COREAPP1	2015.07.17 13:40:39	COREAPP1	CLH	Error	CLH0518	CLH2052	msg discarded due to closed client(0x03a) connection : svc = SLINKA05

[그림] 13-3 Tmax slog

참고

이 기능을 활성화 하기 위해서는 TP 에이전트의 환경파일 pharostp.cfg에서 아래의 항목에 설정이 되어 있어야 한다.

log.sys_log_path=/tmax/log/slog/slog.\$(MMDDYYYY) (slog 파일의 절대경로)

log.sys_log_pattern=(E)|(W)|(I)