애플리케이션 대시보드 살펴보기

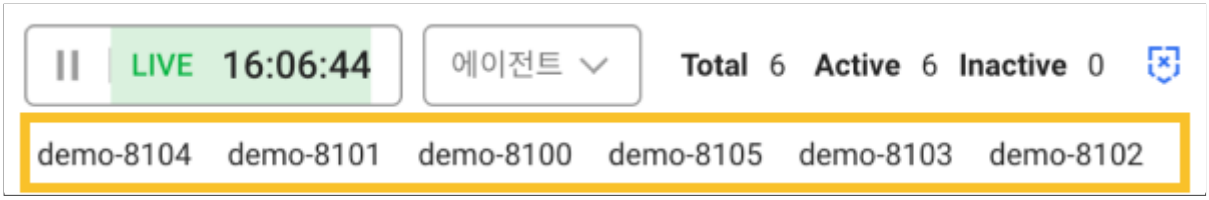
이 문서에서는 와탭 모니터링 서비스 중 하나인 애플리케이션 성능 모니터링(Application Performance Monotoring, 이하 APM)의 애플리케이션 대시보드 메뉴에서 차트형 위젯에 대한 분석 방법을 소개합니다. 웹 애플리케이션 서버(Web Application Server)의 문제를 애플리케이션 대시보드를 통해서 어떻게 파악하고 분석하는지 살펴보겠습니다. 애플리케이션 대시보드의 기능에 대한 소개는 다음 문서를 참조하세요.

하나의 웹 서비스는 수많은 애플리케이션과 플랫폼으로 구성되어 있어서 애플리케이션 관점에서 성능을 분석하게 되면 과정이 복잡합니다. 애플리케이션 대시보드에서는 다음의 관점에서 분석할 수 있도록 대시보드에 위젯을 구성했습니다.

- 애플리케이션 연결 상태: 에이전트
- 트랜잭션: 스피드 미터, 액티브 트랜잭션, 액티브 스테이터스, 히트맵
- 서비스: Apdex, TPS, 평균 응답시간
- 리소스: 시스템 CPU, 힙 메모리
- 사용자: 동시접속 사용자
- 1일 기준 비교: 금일 TPS, 금일 사용자

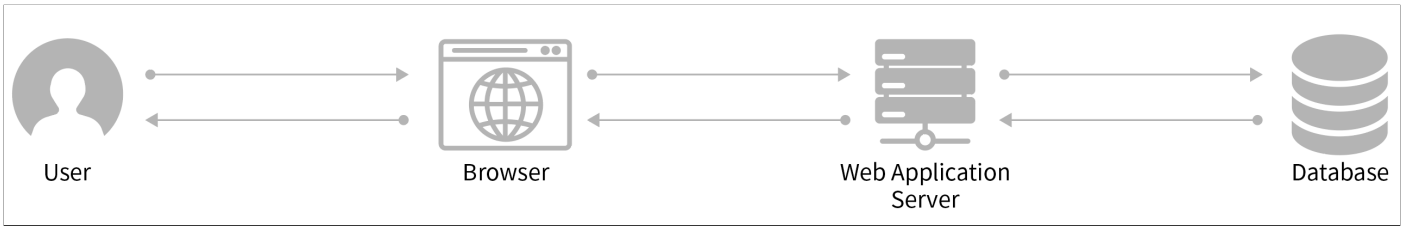
애플리케이션 연결 상태 확인하기

애플리케이션 대시보드 화면 위에는 와탭 모니터링 서비스와 연결된 애플리케이션 서버를 확인할 수 있습니다. 만약 연결된 에이전트가 비활성화되었거나 애플리케이션 서버와 연결이 끊어졌다면 Inactive 항목을 확인하세요. 이를 통해 에이전트들의 상태를 쉽게 파악할 수 있습니다.



연결된 에이전트들을 한줄에 나열해 화면을 효율적으로 이용하고 싶다면 에이전트의 이름을 짧게 설정해 볼 수도 있습니다. 에이전트 이름을 설정하는 방법에 대한 자세한 설명은 다음 문서를 참조하세요.

트랜잭션 분석하기



웹 애플리케이션 서버(WAS)를 이용하는 사용자는 브라우저를 통해 요청한 서비스(AP)의 결과가 제대로 수행되길 원합니다. 서비스(AP)는 제대로된 동작을 위해 서버의 리소스를 이용합니다. 리소스는 AP 프로세스, OS/HW 리소스(CPU, 메모리, 디스크 등), 외부 시스템(Database, 그 외 다른 서버) 등을 의미합니다.

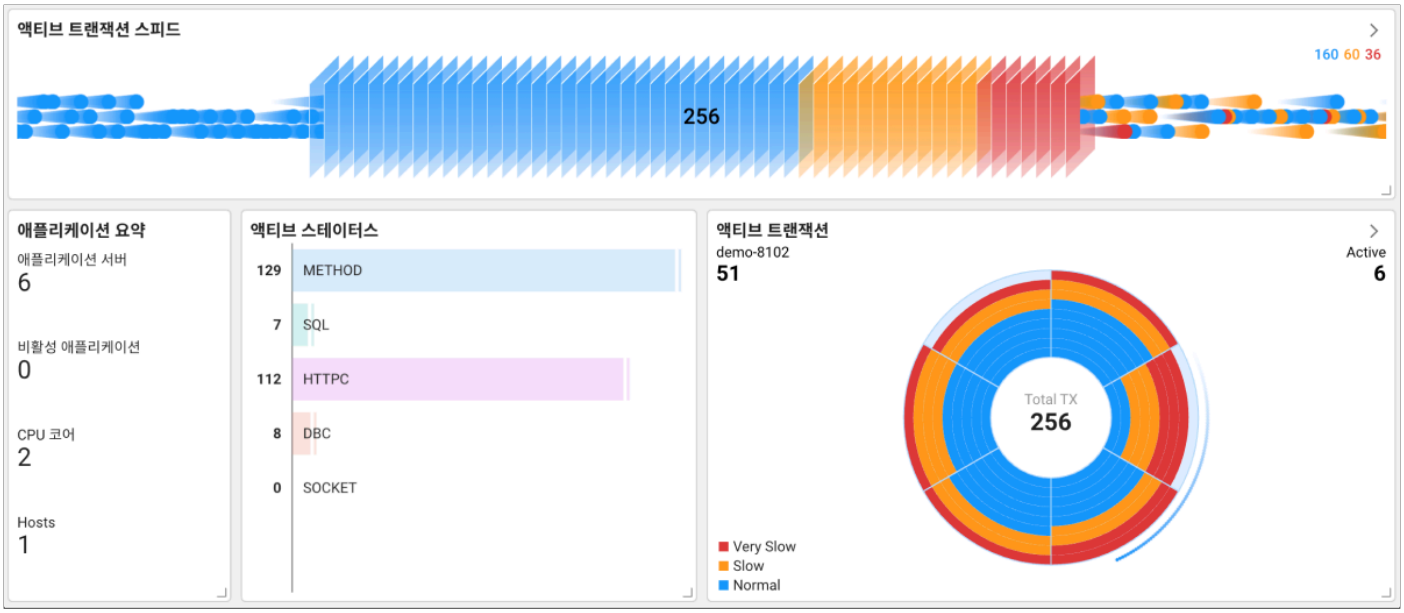
사용자가 원하는 결과를 얻지 못했거나 응답이 너무 느린 경우는 장애가 발생했거 성능에 문제가 있는 경우입니다. 이것은 서비스(AP)의 프로그래밍 문제일 수도 있고 리소스의 문제일 수도 있습니다. 애플리케이션 모니터링의 핵심은 문제의 원인이 프로그래밍인지, 리소스인지, 그 밖의 다른 원인이 있는지를 판별하는 것입니다. 이를 파악하기 위해 사용자의 요청이 제대로 수행되었는지를 확인하는 과정이 필요합니다. 사용자의 요청, 한건 한건을 Request라고 하고, 이 Request를 서버에서 처리하고 사용자에게 응답하는 과정을 **트랜잭션(Transaction)**이라고 정의합니다.

트랜잭션(Transaction)이란 사용자 브라우저의 요청을 처리하기 위한 서버 사이드의 LUW(Logical Unit of Work)를 말합니다. 개별 웹 서비스(URL) 요청에 대한 처리 과정이 트랜잭션입니다. 웹 애플리케이션에서 트랜잭션은 웹서비스에 대한 HTTP 요청(HTTP Request)을 받아 응답(Response)을 반환하는 과정입니다. 와탭은 트랜잭션의 이름을 URL로 저장합니다. 예를 들어, 브라우저 요청 URL이 https://www.whatap.io/hr/apply.do?name='kim' 이라면 트랜잭션 이름은 `/hr/apply.do` 입니다.

트랜잭션이 얼마나 빨리, 에러 없이 처리되는가는 **서비스의 응답이 에러 없이 처리되는가**와 같은 의미로 해석할 수 있습니다. 트랜잭션은 '진행 중 트랜잭션'과 '종료된 트랜잭션'으로 구분합니다. '진행 중 트랜잭션'은 요청을 보냈지만 응답을 받지 않은 상태를 의미하고, '종료된 트랜잭션'은 서비스에 요청을 보내고 응답을 받은 상태를 의미합니다. 와탭의 애플리케이션 모니터링은 트랜잭션이 처리되는 과정을 실시간으로 확인할 수 있는 서비스를 제공합니다.

진행 중 트랜잭션

'진행 중인 트랜잭션'을 통해서 보통 8초 이상 지연을 발생시키는 문제, 광범위하게 나타나지 않고 일부 트랜잭션에서만 발생하는 문제들을 특정할 수 있습니다. 다음 화면 가장 위에 위치한 스피드 미터 위젯에서는 현재 진행 중인 트랜잭션(가운데 영역)과 종료된 트랜잭션(오른쪽 영역) 현황 전체를 확인할 수 있습니다. 전체 현황 중 진행 중인 트랜잭션을 에이전트 별로 나누어 표현한 것이 액티브 트랜잭션 위젯입니다. 진행 중인 트랜잭션의 상태를 나누어 표현한 것은 액티브 스테이터스 위젯입니다.



트랜잭션을 통해 확인할 수 있는 장애의 현황은 우선 응답시간을 통해 알 수 있습니다. 또한 진행 중인 트랜잭션이 종료되지 않는다면 이 또한 장애로 인식해야 합니다. 와탭은 진행 중인 상태의 시간에 따라서 구간을 나누어 표시합니다. 파랑색은 응답 시간이 정상인 트랜잭션, 주황색은 응답 시간이 8초 정도의 느린 트랜잭션, 빨간색은 응답 시간이 보통의 2배 이상으로 느린 트랜잭션을 의미합니다. 이를 통해 사용자는 직관적으로 가장 빨리 장애를 인지할 수 있습니다.

액티브 트랜잭션 위젯에서는 지연이 발생하는 현황을 에이전트 별로 확인할 수 있습니다. 빨간색 영역이 각 에이전트별로 분산되어 있다면 지연을 발생시키는 요소를 에이전트가 설치된 애플리케이션 별로 확인해봐야 합니다. 반대로 빨간색 영역이 한 에이전트에서만 발생하다면 해당 애플리케이션 서버만을 확인하면 됩니다. 이를 통해 장애의 원인이 발생한 위치를 바로 확인할 수 있습니다.

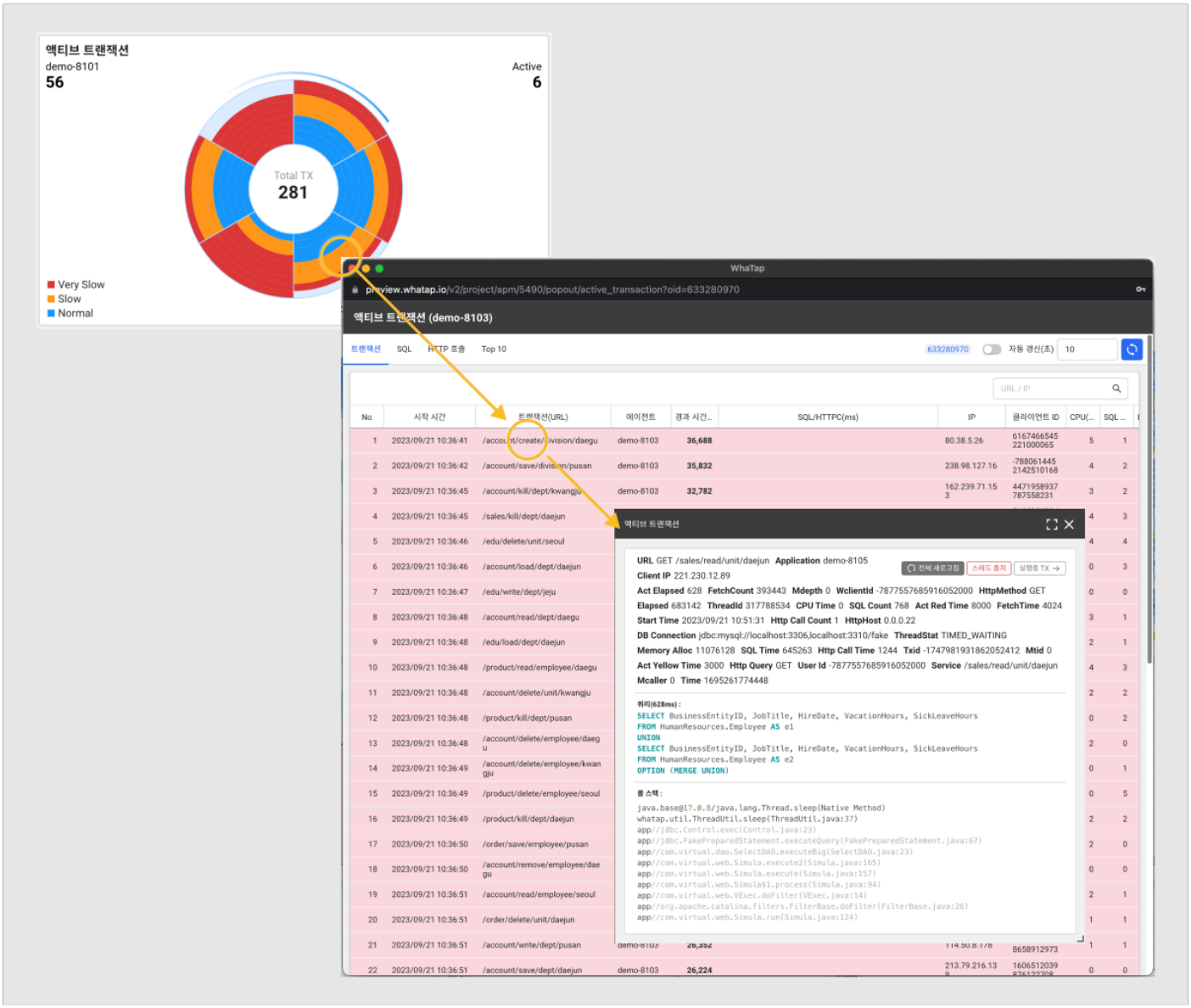
외부의 요소는 액티브 스테이터스 위젯을 통해 확인할 수 있습니다. METHOD는 기타 사항으로 정상 트랜잭션으로 간주합니다. 그 외의 4가지 상태 값을 통해 장애 원인을 파악할 수 있습니다. 외부 연결과 관련한 항목은 HTTPC, SQL 수치입니다. HTTPC 수치가 올라갈수록 외부 연결된 서버와의 응답이 제대로 이루어지지 않는다고 봐야합니다. Database 서버와 연결이 제대로 이뤄어 지지 않는다면 SQL 수치가 올라갑니다.

애플리케이션 서버에서 가장 대표하는 장애 현상 중 하나는 DB Connection Pool과 관련한 것입니다. DB Connection Pool의 개수가 부족하면 새로운 연결 요청이 발생할 때마다 지연이 되면서 성능 장애의 원인이 됩니다. 이 경우 액티브 스테이터스 위젯의 DBC 수치가 증가합니다.

SOCKET은 외부 시스템과의 TCP 연결 시도를 의미합니다. 마찬가지로 SOCKET 수치가 지속적으로 증가한다는 것은 외부 시스템과의 연결이 되지 않아 장애가 발생 중일 가능성이 높습니다.

진행 중인 트랜잭션에 문제가 발생했다면 액티브 트랜잭션 위젯의 차트를 클릭해 스택을 확인할 수 있습니다. 액티브 트랜잭션 위젯에서 장애가 발생한 영역의 에이전트를 클릭하면 해당 에이전트의 액티브 트랜잭션 창이 나타납니다. 이 창에는 진행 중인 트랜잭션을 목록으로 나타납니다. 이 목록에서 트랜잭션(URL) 또는 SQL/HTTPC(ms) 항목이 동일하게 출력된 것이 있는지 확인해보세요. 일차적으로

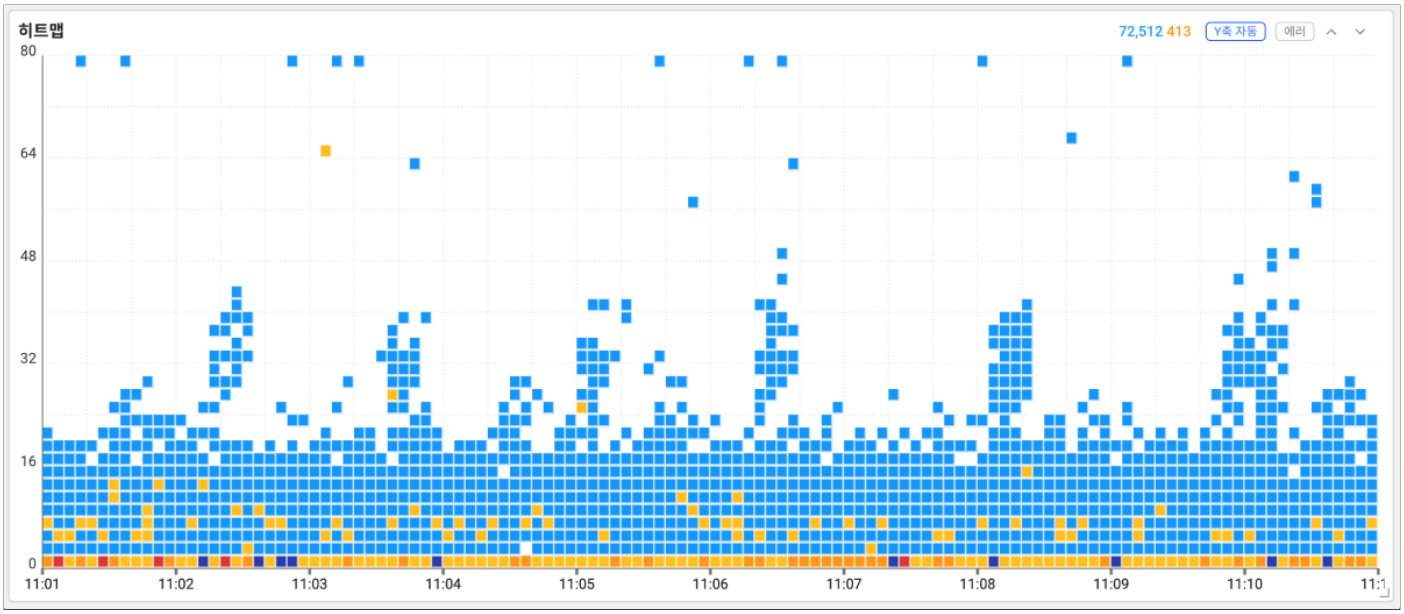
발생하는 장애의 원인이 동일한 현상인지 그 외에 다른 문제가 발생한 것인지 확인해볼 수 있습니다.



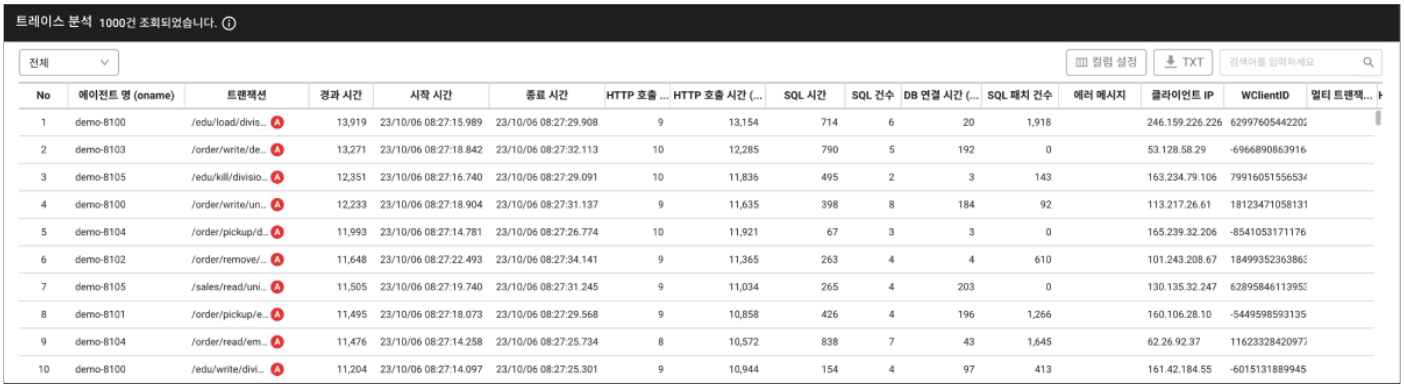
진행 중인 트랜잭션 목록에서 항목을 클릭하면 해당 트랜잭션의 스택을 확인할 수 있습니다. 이 스택 정보를 통해서 장애 원인을 분석할 수 있습니다.

종료된 트랜잭션

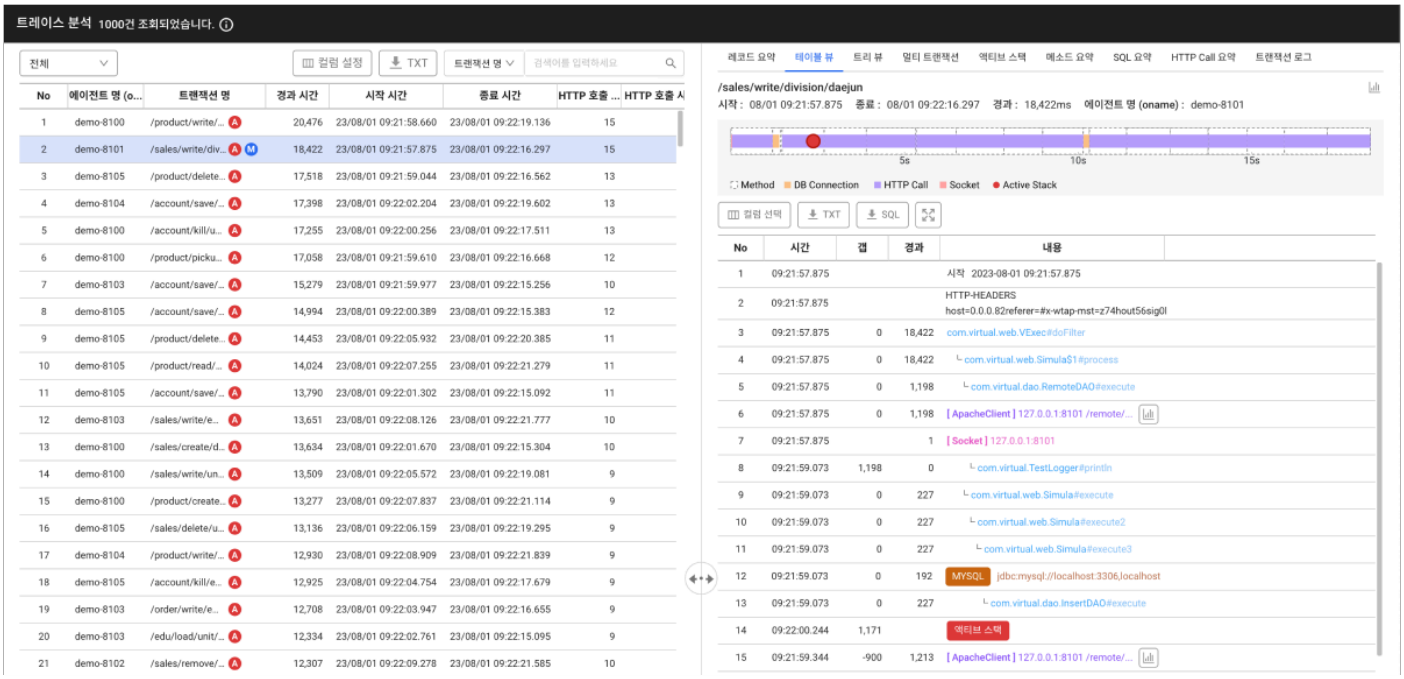
종료된 개별 트랜잭션을 분포도로 확인할 수 있는 것이 히트맵 위젯입니다. 히트맵 위젯에서는 본래 1초 걸려야 하는 트랜잭션이 예상과 달리 2초 정도 소요되는 경우, 즉 응답시간이 2배 이상 소요되는 트랜잭션을 찾아 장애 원인을 분석합니다.



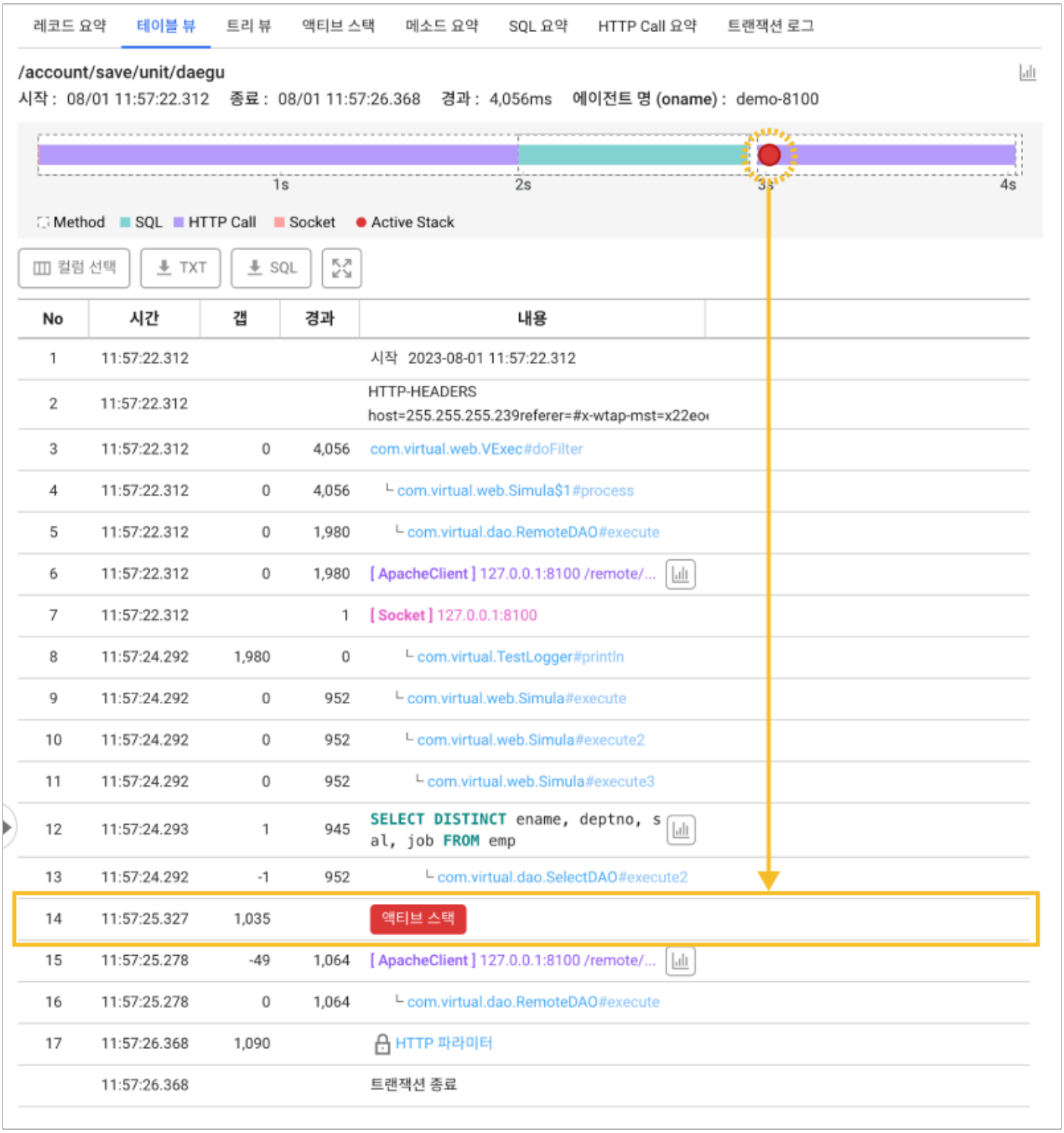
위젯 오른쪽 위에 ^ 또는 v 버튼을 눌러 차트를 확대 또는 축소할 수 있습니다. 차트 영역을 마우스로 드래그하면 트레이스 분석 창이 나타납니다.



트레이스 분석 창에서는 사용자가 드래그한 차트 영역의 트랜잭션 정보를 목록으로 표시합니다. 각각의 항목을 클릭하면 해당 트랜잭션의 상세 정보를 오른쪽에서 확인할 수 있습니다. 오른쪽 화면의 테이블 뷰 탭을 통해서 어떤 프로세스를 처리하면서 느려졌는지 단계별로 확인할 수 있습니다.

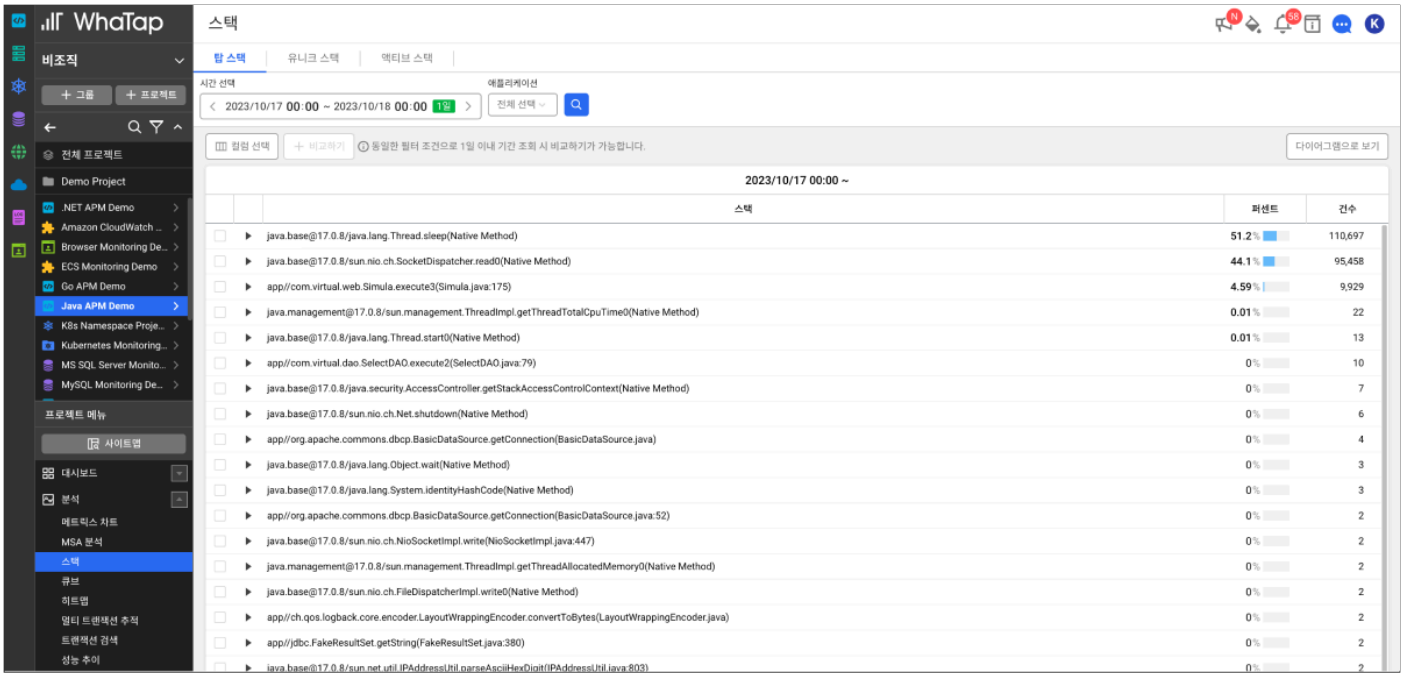


사용자의 요청이 서버로 와서, 서버에서 로직을 수행 후 결과를 나타낼 때까지 수많은 메소드가 실행됩니다. 여기서 어떤 메소드를 추적해야 느려지는 구간을 확인할 수 있을까요? 와탭은 여러 개의 사용자 요청을 기본 10초 간격마다 스냅샷으로 저장합니다. 스냅샷에 저장된 스택 정보를 트랜잭션에 포함시키도록 합니다. 이때의 스택 정보를 액티브 스택(Active Stack)이라고 정의합니다.



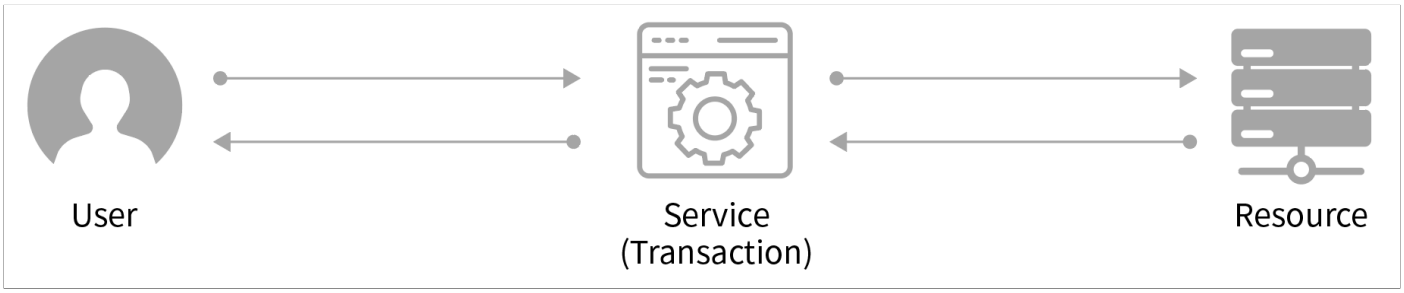
> ⓘ 트레이스 분석 창에 대한 자세한 내용은 다음 문서를 참조하세요.

스택들을 모아서 누적된 양을 기준으로 목록화하여 확인할 수 있는 기능이 분석 > 스택입니다. 리소스와 연계되지 않은 내부 로직에서의 지연되는 메소드 구간을 통계적으로 찾아낼 수 있습니다.

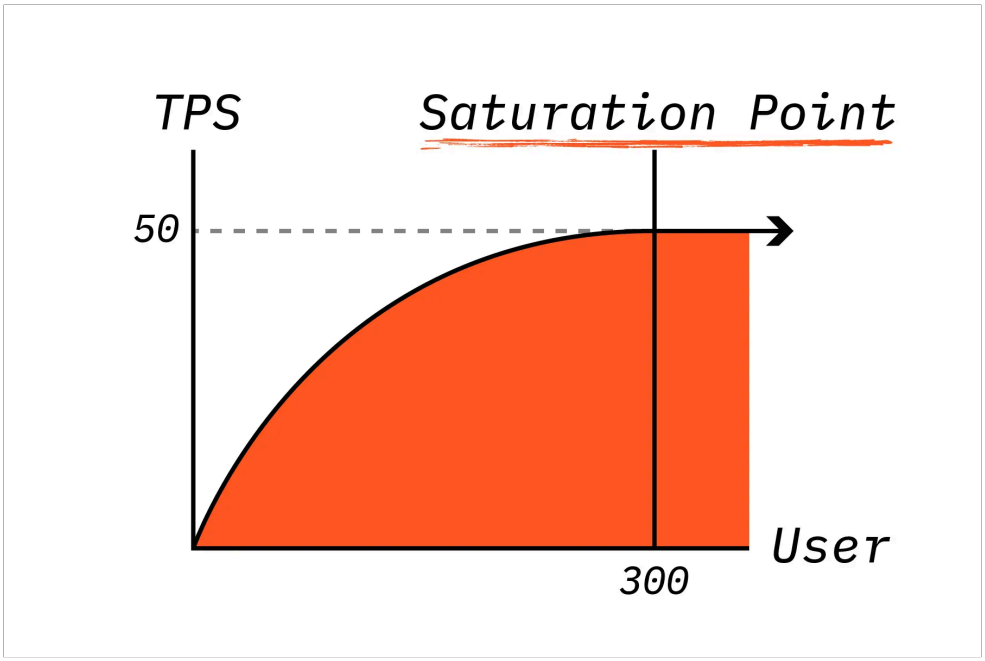


사용자 및 서비스, 리소스 분석

사용자와 서비스, 리소스 간의 상관 관계를 분석해 AP 튜닝을 진행하려면 사용자 수에 따른 TPS, 응답시간, CPU 사용률을 확인해야 합니다.

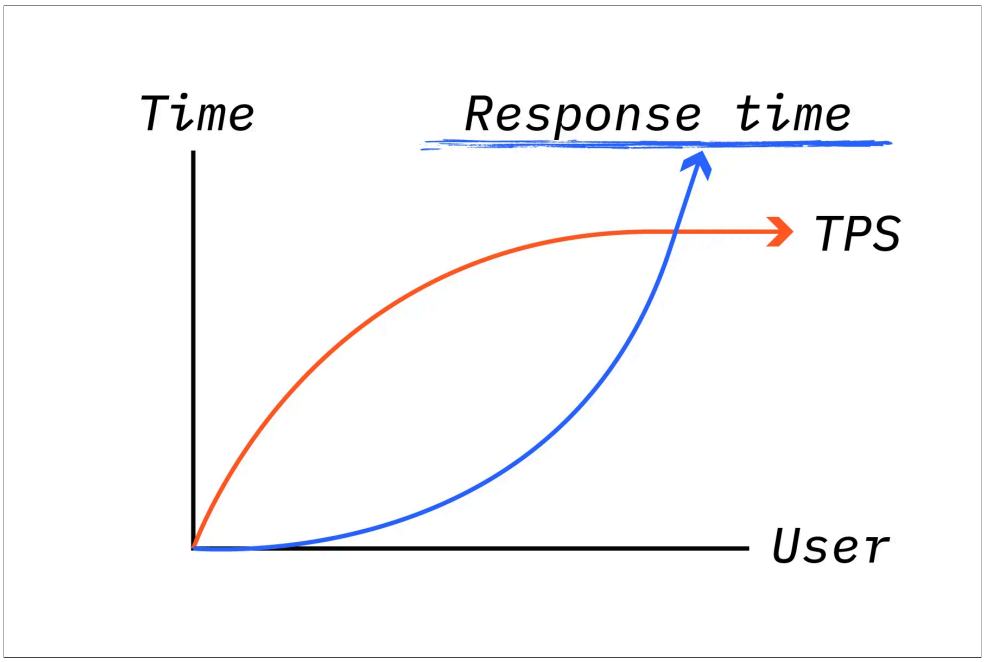


Transaction Per Second(TPS, 이하 TPS)는 초당 트랜잭션을 처리한 양을 의미합니다. 사용자가 증가함에 따라 TPS는 다음과 같이 그래프가 그려집니다.

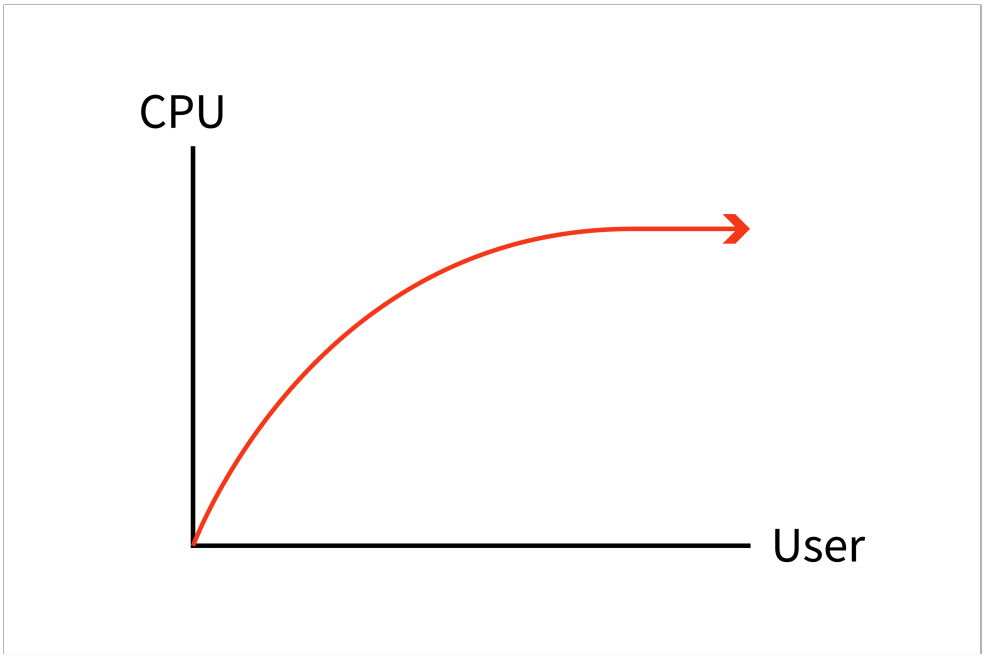


서비스에 사용자가 지속적으로 늘어나면 어느 순간부터 TPS는 더이상 증가하지 않는 상황이 발생합니다. 이렇게 증가하지 않는 지점을 **최대 임계 사용점(Saturation Point)**이라고 합니다. 위 그래프와 같은 서비스가 이상적인 상황입니다. 제대로 튜닝이 되지 않은 서비스는 오히려 TPS가 떨어지는 현상이 발생합니다.

다음 그래프를 참조하세요. TPS가 더이상 증가하지 않은 상태에서 사용자가 늘어나면 응답시간은 사용자에 비례하여 늘어나게 됩니다.



CPU 사용률은 사용자가 증가함에 따라 다음과 같은 그래프가 그려집니다.

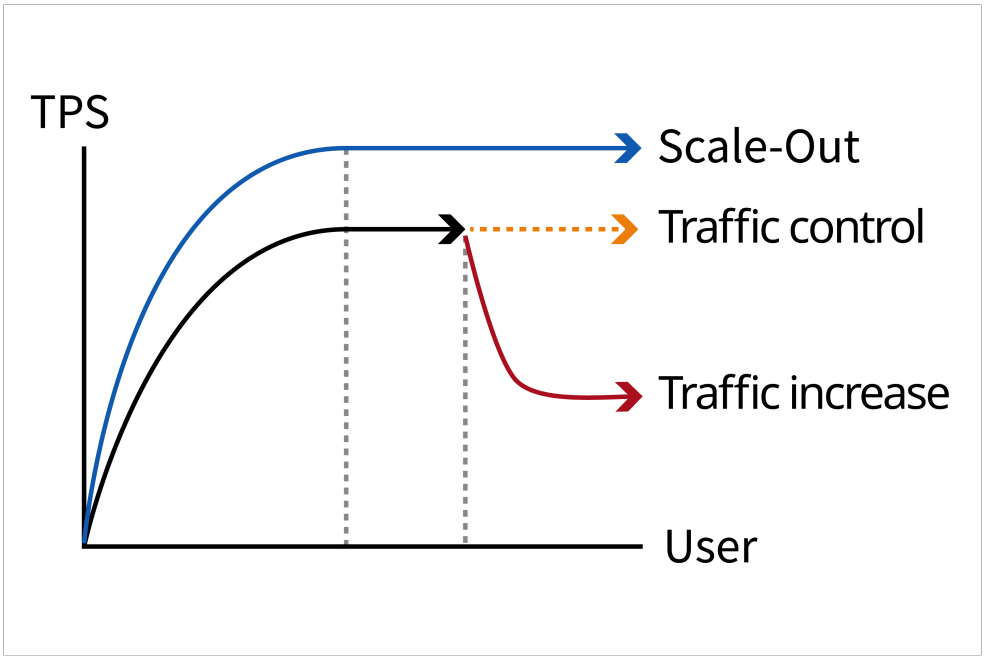


TPS의 최대값을 알기 어렵지만 CPU의 최대 수치는 100%입니다. 그러나 CPU의 사용률이 100%이거나 100%에 근접했더하더라도 CPU 사용률만으로는 리소스가 부족한지를 판단하기 어렵습니다. 서비스의 로직에 문제가 있어 CPU 사용률이 높아질 수도 있기 때문입니다. 그래서 위 3개의 그래프를 직관적으로 파악하고 비교해서 판단하는 것이 중요합니다.

TPS는 성능의 기준입니다. 즉 최대 TPS는 시스템의 최대 성능을 의미합니다. 최적의 튜닝은 최대 TPS 값을 늘리는 것입니다.

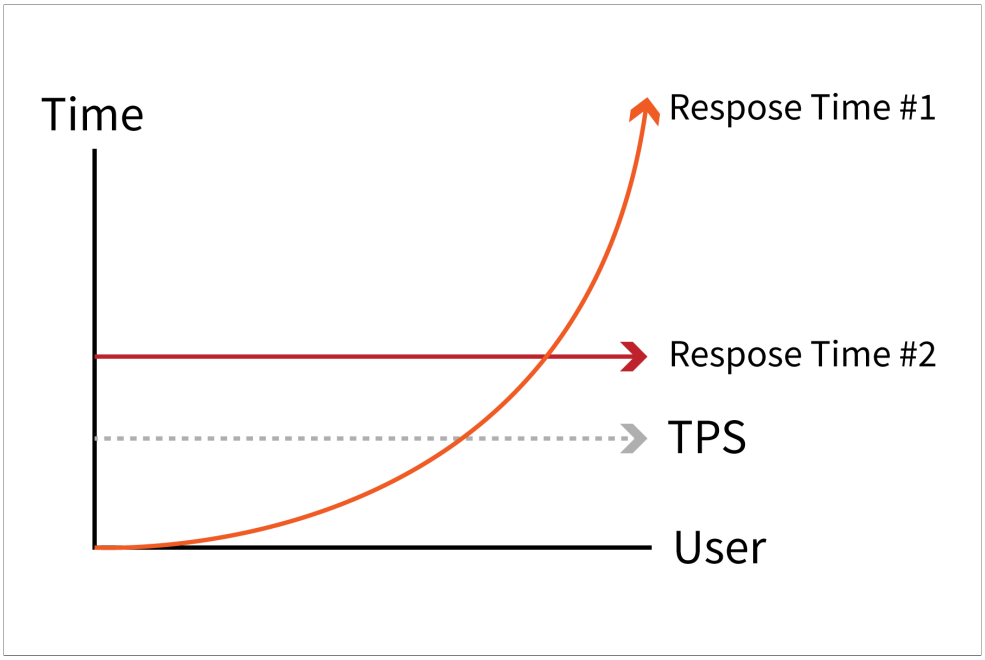
응답시간은 사용자 수가 늘어남에 따라 비례하여 증가하기 때문에 튜닝의 지표로 삼기는 어렵습니다. 다만 응답시간이 완만한 그래프가 되도록 하는 것이 중요합니다. 최대 TPS를 늘리려면 응답시간에 대한 그래프를 완만하게 만들어야 합니다. 그래서 튜닝의 대상은 응답시간입니다. 개별 URL, 즉 트랜잭션의 응답시간을 파악하고 분석하는 것이 필요합니다. 히트맵을 이용하면 트랜잭션의 응답시간을 분석하고 느려지는 구간을 찾아 원인을 파악할 수 있습니다.

위의 개념을 숙지한 상태에서 애플리케이션 대시보드의 TPS 위젯과 금일 사용자 또는 동시접속 사용자 위젯을 확인해 보세요. 사용자 수가 늘어나는 가운데 TPS 수치가 줄어든다면 서버에 트래픽이 증가한다는 의미입니다. 이런 경우 Scale-Out을 통해 서버 리소스를 증설하거나 트래픽을 제어하여 TPS 수치를 일정하게 유지하는 것이 좋습니다.



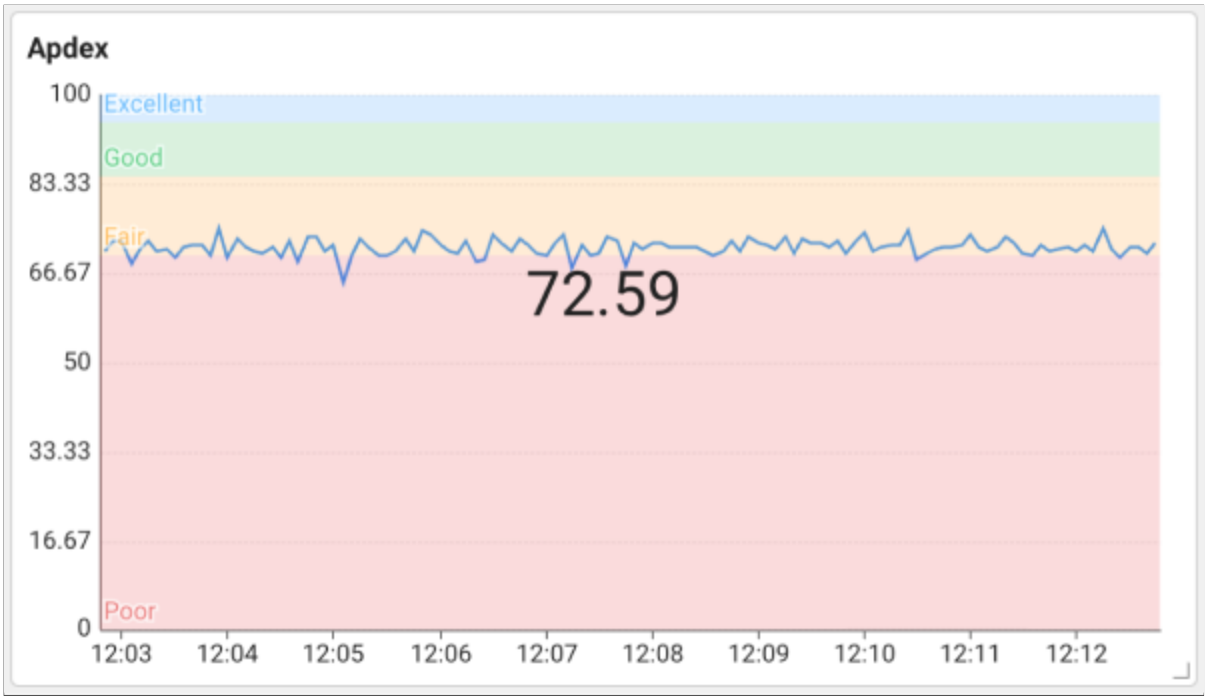
Scale-Out을 진행하게 된다면 와탭의 에이전트를 추가해 모니터링을 동시에 진행할 수 있도록 조치를 해줘야 합니다. Traffic 제어가 필요하다면 와탭의 에이전트 설정을 통해 진행할 수 있습니다.
애플리케이션의 최대 동시 처리 수를 제한하는 쓰로틀링 기능을 활성화할 수 있습니다. 자세한 설정 방법은 다음 문서를 참조하세요.

그 외 다음 그래프처럼 사용자는 증가하는데 TPS 수치가 최대로 오르지 않은 채 일정하게 유지되는 경우 2가지 상황을 고려해봐야 합니다. 그래프 **Response #1**과 같이 응답시간이 지속적으로 오르는 경우는 서비스 로직에 문제가 있을 수 있습니다. 반대로 그래프 **Response #2**와 같이 응답시간도 일정한 수치를 유지한다면 사용자의 요청을 서버가 수신하지 못하는 상황으로 네트워크 상태를 확인해봐야 합니다.



이 외에 웹 애플리케이션의 고객 만족도를 측정하는 Apdex 위젯, 힙 메모리 위젯을 확인할 수 있습니다.

Application Performance Index(Apdex)는 애플리케이션 성능 지표입니다.



사용자 만족도에 대한 지표로 활용할 수 있으며 0~1 사이의 값을 가지며, 0~0.7은 Poor(빨간색), 0.7~0.85는 Fair(주황색), 0.85~0.95는 Good(초록색), 0.95~1은 Excellent(파랑색)의 역역으로 표시합니다. Apdex 지표와 색상 영역을 통해서 애플리케이션의 응답 속도가 사용자에게 만족할 만한 수준인지 확인할 수 있습니다. Apdex 지표에 대한 자세한 내용은 다음 문서를 참조하세요.

힙 메모리(Heap Memory)위젯은 각 서버당 사용 가능한 최대 메모리와 현재 사용 중인 메모리를 표현합니다.

Heap 사용량이 부족한 경우 GC가 자주 일어나 CPU를 과도하게 점유하게 됩니다. 정상적인 Heap 사용량 패턴은 GC로 인해 오르락 내리락을 반복합니다. 힙 메모리 차트 분석에 대한 자세한 내용은 다음 링크를 참조하세요.

- 월간 와탭 : 모니터링에 주목해야할 지표
- Java 힙 메모리 차트 분석 : Ch.1 힙차트 관찰하기
- JAVA 힙메모리 차트 분석 : Ch.2 메모리 릭, 그리고 힙덤프 분석

통합 대시보드를 제공하는 가시성 높은 모니터링 서비스 와탭과 함께 모니터링 업무의 효율을 높여 보길 바랍니다.