

분산 추적이 APM에 필수적인 이유

복잡한 분산 애플리케이션 환경에서 MTTR 감소



목차

03 복잡성 제거

04 분산 시스템에서의 경로 추적

- 06 > 트레이스가 필요한 경우
- 07 > 트레이스의 작동 방식
- 08 > 점 연결
- 09 >

10 데이터 파이프라인에 대한 가시성 확보

- 10 > 효율적인 헤드 기반 샘플링
- 11 > 테일 기반 샘플링으로 실행 가능한 트레이스
- 13 > 분석 및 시각화

14 관리 부담 해소

15 앞면 또는 뒷면? 동전 던지기로 결정할 필요는 없습니다.

16 다음 단계

17 뉴렐릭 소개



복잡성 제거

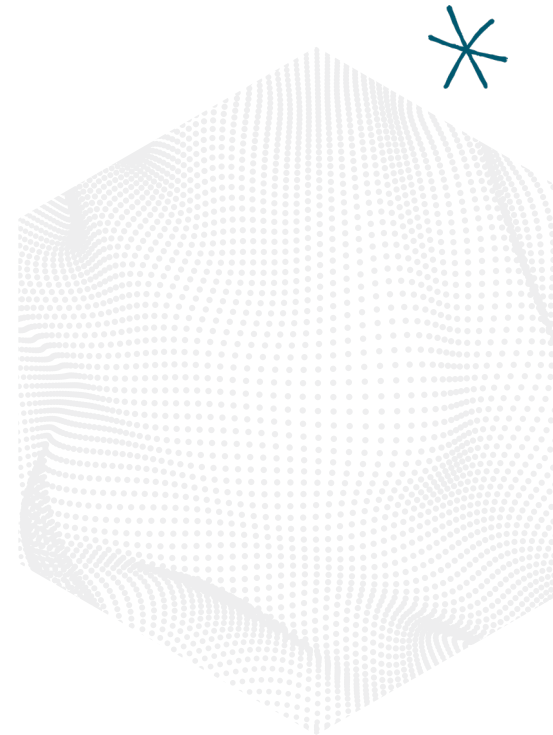
마이크로서비스 같은 최신 소프트웨어 환경과 아키텍처에서는 애플리케이션 개발을 가속화할 수 있습니다. 그러나 많은 조직에서, 환경의 복잡성으로 인해 소프트웨어 엔지니어링 팀은 성능 문제와 오류가 안정성과 고객 경험에 영향을 미치기 전에 진단하여 해결하는 데 어려움을 겪고 있습니다.

마이크로서비스 환경은 수십 개에서 수백 개의 서비스를 포함할 수 있으므로, 요청 경로를 결정하고 문제를 진단하기가 어렵습니다. 또한 빈번한 소프트웨어 구현을 지원하기 위한 오케스트레이션, 자동화 및 연속 통합/연속 개발(CI/CD)로 인해, 애플리케이션 성능 모니터링(APM)의 부담은 지속적으로 늘어납니다. 적절한 모니터링 장비가 없으면 팀이 분산 시스템에서 반복적으로 답을 찾아야 하며, 이는 평균 해결 시간(MTTR)을 증가시키고 혁신적인 소프트웨어 개발에 할애할 시간을 빼앗아 갑니다.

옵저버빌리티는 소프트웨어의 복잡성을 줄이고 포괄적인 가시성을 제공하여, 팀이 문제를 더 빠르게 해결하고, 더 스마트하게 작업하며, 고객을 위해 더 나은 디지털 경험을 구축할 수 있도록 해줍니다. 옵저버빌리티는 메트릭, 이벤트, 로그 및 트레이스라는 네 가지 필수 옵저버빌리티 데이터 유형을 결합함으로써, 컨텍스트와 실행 가능한 인사이트를 생성합니다.

트레이스(Trace), 더욱 정확하게 말해 분산된 트레이스는 클라우드로 전환했거나 전환을 고려 중인 소프트웨어 팀에서 마이크로서비스 아키텍처를 도입하는 데 필수적입니다. 분산 추적은 분산 애플리케이션을 구성하는 마이크로서비스를 통해 요청이 전송되는 동안 발생하는 상황을 신속하게 파악할 수 있는 가장 좋은 방법이기 때문입니다.

비즈니스 리더, 데브옵스(DevOps) 엔지니어, 제품 소유자, 사이트 안정성 엔지니어(SRE), 소프트웨어 팀 리더 또는 기타 이해 관계자는 분산 추적을 사용해 병목 현상이나 오류를 찾고 더 빠르게 문제를 해결할 수 있습니다.



분산 시스템에서의 경로 추적

분산 추적은 이제 최신 애플리케이션 환경의 운영 및 모니터링을 위해 필요한 기본 요소가 되었습니다. 추적은 팀이 오피저빌리티를 얻기 위해 소프트웨어와 시스템 성능을 모니터링할 때, 요청이 여러 분산된 환경들로 전달되어 서비스에서 서비스로 이동하는 동안 이를 모니터링하고 분석하는 방법입니다.

분산 추적은 요청이 한 서비스에서 다른 서비스로 이동할 때 데이터를 수집하여 분산 시스템을 통해 흐르는 서비스 요청을 추적하고 관찰하는 역량을 말합니다. 추적 데이터는 팀이 마이크로서비스 환경 전반에서 요청의 흐름을 이해하고 시스템에서 장애나 성능 문제가 발생한 위치와 그 원인을 정확히 파악하는 데 도움을 줍니다.

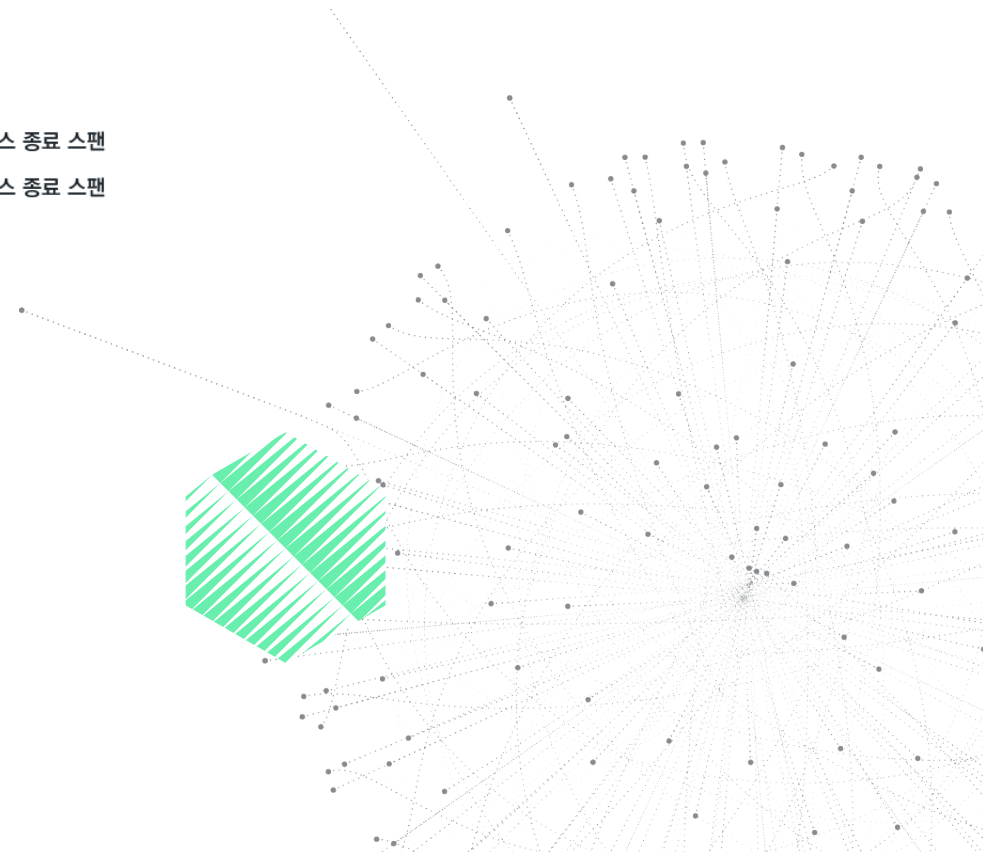
팀이 분산 추적을 위해 시스템을 계측할 때, 모든 트랜잭션은 프런트 엔드 사용자부터 백 엔드 데이터베이스 호출에 이르기까지 트레이스 텔레메트리(trace telemetry)를 생성합니다. 예를 들어, 고객이 이커머스 애플리케이션에서 장바구니를 클릭해 구매하는 경우, 그 요청은 여러 컨테이너, 서버리스 환경, 가상 머신, 다양한 클라우드 공급업체, 온프레미스 또는 이들의 조합을 거칩니다. 요청에는 재고 여부 확인 서비스, 결제 서비스 및 배송 서비스가 포함되어 있을 수 있습니다. 요청이 완료되면 사용자에게 다시 돌아옵니다. 한 서비스에서 다른 서비스로 요청이 이동될 때마다 트레이스 텔레메트리를 통해 스패н(span)이 방출됩니다. 요청이 완료되면, 스패ん들은 함께 연결되어 시스템 전반에서 이뤄진 요청의 여정에 대한 완전한 트레이스를 생성합니다.

분산 추적을 통해, 팀은 다음과 같은 이점을 얻을 수 있습니다.

- 요청이 복잡한 시스템을 통과할 때 해당 요청의 경로 추적
- 업스트림 및 다운스트림 서비스 종속성 이해
- 해당 경로를 따라 구성 요소의 레이턴시 파악
- 요청 경로에서 병목 현상이 발생하는 위치 파악
- 개별 서비스 수준에서 트랜잭션의 오류 발생 위치를 확인 및 분석



애플리케이션 서비스의 각 단계에서 각 요청이 소요된 시간을 보여주는 분산형 차트 및 워터폴 시각화



추적이 필요한 경우

일반적으로 분산 추적은 데브옵스, 운영, 소프트웨어 및 SRE 팀이 소프트웨어가 분산되어 있거나 서버리스 아키텍처에 의존하는 환경에서 특정 질문에 대한 답변을 신속하게 얻을 수 있는 가장 좋은 방법입니다. 요청이 소수의 마이크로서비스와 관련되는 즉시, 모든 다른 서비스가 어떻게 함께 작동하는지 확인할 수 있는 방법이 필요합니다.

트레이스 데이터는 애플리케이션 전체와 서비스 및 엔터티 간에 발생하는 상황에 대한 컨텍스트를 제공합니다. 각 서비스에 대한 원시 이벤트만 격리된 경우, 서비스 간에 특정 트랜잭션을 위한 단일 체인을 구성할 수가 없습니다.

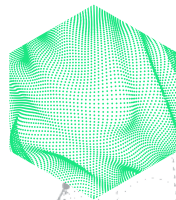
애플리케이션은 수행하려는 작업에 따라 여러 애플리케이션을 호출하는 경우가 많고, 데이터를 병렬로 처리하는 경우도 많습니다. 따라서 호출 체인은 일관성이 없고 타이밍을 기반으로 상관 관계를 구축할 수 없습니다. 일관된 호출 체인을 보장하는 유일한 방법은 각 서비스 간에 트레이스 컨텍스트를 전달하여 전체 체인에서 단일 트랜잭션을 식별하는 것입니다.

이는 팀이 분산 추적을 사용하여 다음과 같은 질문에 대한 답을 얻어야 한다는 것을 의미합니다.

- 분산 시스템을 구성하는 서비스의 상태는 어떠한가?
- 분산 시스템 내에서 오류 및 결함의 근본 원인은 무엇인가?
- 고객 경험에 영향을 줄 수 있는 성능 병목 현상은 어디에 있는가?
- 서비스에 문제가 있거나 최적화를 위해 팀이 우선순위를 지정해야 하는 비효율적인 코드가 있는가?

분산 추적 용어 간단 가이드

- **트랜잭션(Transaction)**은 소프트웨어 애플리케이션에서 작업 단위를 구성하는 함수 및 메서드 호출입니다. 트랜잭션은 메서드가 호출될 때 시작되고 메서드가 반환되거나 오류가 발생하면 종료됩니다.
- **요청(Request)**은 애플리케이션, 마이크로서비스 및 기능이 서로 통신할 수 있는 방법입니다.
- **트레이스(Trace)**는 마이크로서비스들 간을 이동하는 요청에 대한 성능 데이터입니다.
- **스팬(Span)**은 트레이스의 일부인 작업이나 세그먼트입니다.
- **루트 스팸(Root span)**은 트레이스의 첫 번째 스팸입니다.
- **자식 스팸(Child span)**은 중첩될 수 있는 후속 스팸입니다.



추적의 작동 방식

함께 연결된 트레이스들은 단일 트랜잭션을 위한 마이크로서비스 에코시스템을 통해 인과 체인을 추적하는 데 도움을 주는 '스팬(span)'이라는 특별한 이벤트를 형성합니다. 스패를 완료하기 위해, 각 서비스는 트레이스 컨텍스트라고 하는 상관 관계 식별자를 서로에게 전달합니다. 이 트레이스 컨텍스트는 스패에 속성을 추가하는 데 사용됩니다.

타임스탬프	EventType	TraceID	SpanID	ParentID	ServiceID	소요 시간
11/8/2022 15:34:23	스팬	2ec68b32	aaa111	bbb111	자동 판매기	23
11/8/2022 15:34:22	스팬	2ec68b32	bbb111	aaa111	자동 판매기 백엔드	18
11/8/2022 15:34:20	스팬	2ec68b32	ccc111	bbb111	신용카드 회사	15
11/8/2022 15:34:19	스팬	2ec68b32	ddd111	ccc111	카드 발급 은행	3

신용 카드 트랜잭션에서 스패로 구성된 분산 추적의 예

위의 표에서 타임스탬프와 기간 데이터를 보면, 트랜잭션에서 신용 카드 회사의 서비스 속도가 가장 느리며 전체 트랜잭션 23초 중 12초가 걸려 전체 트레이스 시간의 절반 이상을 차지한다는 것을 알 수 있습니다.



어떻게 12초라는 수치가 나왔을까요?

카드 발행 은행에 연락하는 스패를 자식 스패이라고 합니다. 신용 카드 회사에 연락하는 스패는 부모 스패입니다. 은행 요청이 3초 걸렸고 신용 카드 회사에서 15초가 걸렸으므로, 부모 스패에서 자녀 스패를 빼면 신용 카드 트랜잭션을 처리하는 데 12초가 걸린 것입니다.

점 연결

분산 애플리케이션으로 전환하기 시작한 조직은 개별 마이크로서비스와 전체 요청의 흐름을 독립적으로 파악할 수 있는 방법이 필요하다는 사실을 곧 깨닫게 되었습니다. 이러한 마이그레이션 때문에, 분산 추적은 현재 상황을 파악하는 데 필요한 모범 사례로 자리를 잡았습니다. 또한 트레이스를 다른 세 가지 필수 텔레메트리 데이터(메트릭, 이벤트, 로그)와 결합하면 완전한 옴저버빌리티를 확보해 소프트웨어 환경과 성능을 완전하게 파악할 수 있습니다.

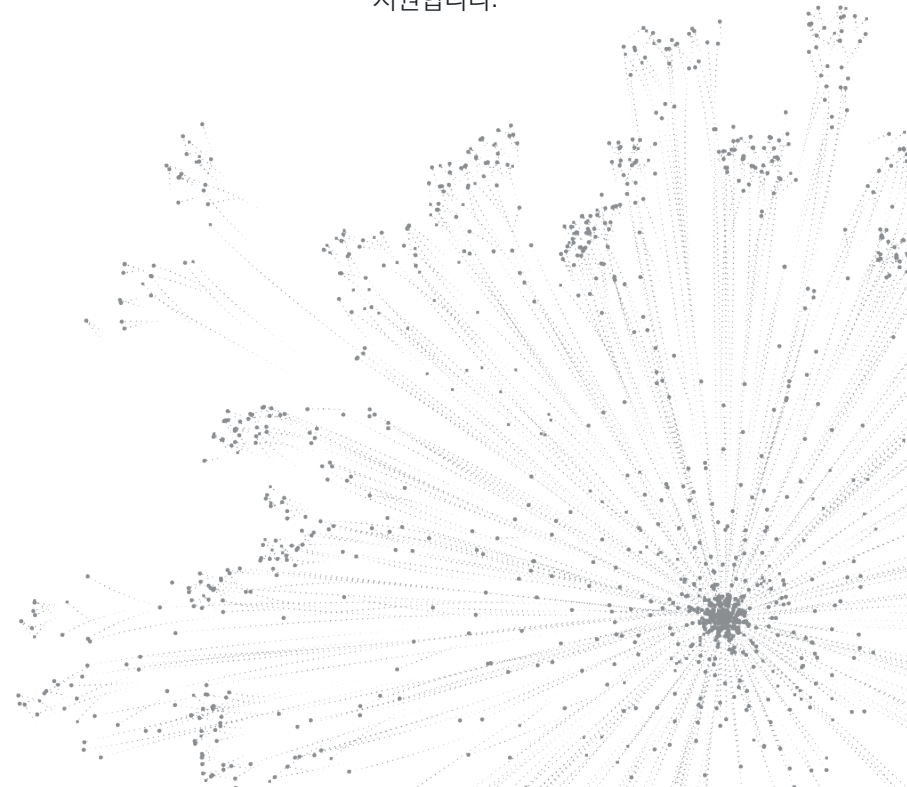
분산 추적에는 트레이스 컨텍스트도 필요합니다. 이는 각 요청에 고유 ID를 할당하고, 추적의 각 단계에 고유 ID를 할당하여, 이 컨텍스트 정보를 인코딩하고, 요청이 애플리케이션 환경을 통과할 때 인코딩된 컨텍스트를 한 서비스에서 다음 서비스로 전달(전파)해야 한다는 것을 의미합니다. 이러한 프로세스는 분산 추적 톨이 트레이스의 각 단계를 정확한 순서로 상호 연관시킬 수 있도록 하며, 성능을 모니터링하고 추적하는 데 필요한 다른 정보를 제공합니다.

단일 트레이스는 일반적으로 다음에 대한 데이터를 캡처합니다.

- 스펜(서비스 이름, 작업 이름, 소요 시간 및 기타 메타데이터)
- 오류
- 각 서비스 내에서 중요한 작업의 소요 시간(예: 내부 메서드 호출 및 함수)
- 커스텀 속성



W3C 트레이스 컨텍스트는 프로세스 경계 간에 트레이스 컨텍스트를 전파하기 위한 표준으로 자리를 잡았습니다. 이를 사용하면 루트 서비스에서 터미널 서비스로 전달되는 트레이스 데이터와 함께, 표준을 준수하는 모든 트레이서와 에이전트가 트레이스에 참여할 수 있습니다. 뉴렐릭을 비롯한 많은 옴저버빌리티 공급업체들은 W3C 트레이스 컨텍스트 표준을 완전하게 지원합니다.



조직에 분산 추적이 필요한 이유

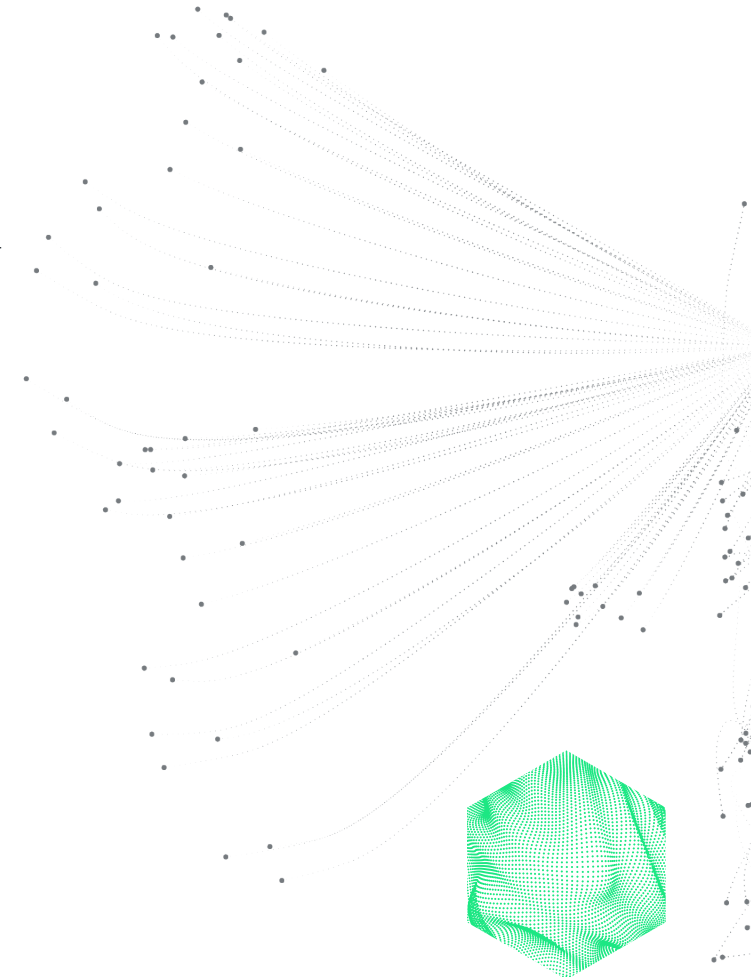
클라우드, 마이크로서비스, 컨테이너, 서버리스, 데브옵스(DevOps), 사이트 안정성 엔지니어링 등 새로운 기술과 관행의 속도가 빨라지고, 코드에서 운영 환경으로 소프트웨어를 가져오는 과정에서 발생하는 문제가 줄어들자, 새로운 도전과제가 생겨나고 있습니다.

- 애플리케이션 스택 내에서 더 많은 오류 지점 발생
- 애플리케이션 환경의 복잡성으로 인해 MTTR 증가
- 문제를 진단하는 데 더 많은 시간이 필요하여 혁신을 위한 시간 감소

예를 들어, 요청이 느리게 실행되면 고객 경험에 영향을 줄 수 있습니다. 이 요청은 여러 마이크로서비스와 서버리스 기능에 분산됩니다. 여러 팀이 요청과 관련된 다양한 서비스를 소유 및 모니터링하며, 마이크로서비스에 대한 성능 문제는 보고하지 않습니다. 여러 서비스에서 전체 요청의 성능을 파악할 수 있는 방법이 없으면, 레이턴시가 많이 발생하는 위치와 이유, 문제를 해결해야 할 팀을 정확히 파악하는 것이 불가능합니다. 포괄적인 오피저빌리티 전략의 일환으로 분산 추적은 최신 애플리케이션 환경의 도전과제를 해결해줍니다.

소프트웨어 팀은 업스트림 및 다운스트림 모두에서 모든 서비스의 성능을 깊이 있게 이해함으로써 다음과 같은 혜택을 더욱 효과적이고 빠르게 얻을 수 있습니다.

- 문제를 식별하고 해결하여 고객 경험과 비즈니스 결과에 미치는 영향을 최소화할 수 있습니다.
- 전반적인 시스템 상태를 측정하고 변경 사항이 고객 경험에 미치는 영향을 이해할 수 있습니다.
- 디지털 고객 경험을 최적화하기 위해 개선해야 할 가치가 높은 영역의 우선순위를 지정할 수 있습니다.
- 혁신을 가지고 지속적으로 혁신하여 경쟁에서 앞서 나갈 수 있습니다.



데이터 파이프라인에 대한 가시성 확보

분산 추적에는 트레이스 텔레메트리의 보고 및 처리가 필요합니다. 트레이스 데이터의 양은 시간이 지나고 요청량이 증가하며 팀이 환경 내에 더 많은 마이크로서비스를 구축함에 따라 기하급수적으로 증가할 수 있습니다.

이러한 이유로 많은 조직에서 트레이스 작업 전송과 관련된 복잡성과 비용을 관리하기 위해 데이터 샘플링을 사용하고 있습니다. 이상적인 경우 샘플링된 데이터는 더 큰 데이터 모집단의 특성을 나타냅니다.

소프트웨어 팀은 각 애플리케이션의 모니터링 요구 사항을 충족하기 위해 헤드 또는 테일 기반 샘플링을 선택할 수 있는 유연성이 필요합니다.

효율적인 헤드 기반 샘플링

헤드 기반 샘플링은 트레이스 데이터를 무작위로 수집하고 저장하고, 루트(첫 번째) 스패ンは 접촉하는 모든 서비스에서 트랜잭션이 어떻게 되는지를 추적하고 분석하기 위해 처리됩니다. 일반적으로 헤드 기반 샘플링은 분석을 위해 샘플링할 트레이스를 임의로 선택함으로써 트레이스 텔레메트리를 수집하는 에이전트 내에서 수행됩니다. 샘플링 결정은 트레이스가 완료되기 전에 이루어집니다. 문제가 발생할 수 있는 트레이스를 알 수 있는 방법이 없기 때문에, 팀이 비정상적으로 느린 프로세스나 오류가 포함된 트레이스를 놓칠 수 있습니다.

헤드 기반 샘플링은 분산 시스템을 통과하는 요청의 전반적인 통계적 샘플링을 제공하는 데 적합합니다. 모노리스 및 마이크로서비스 기반 아키텍처가 혼합된 환경이나 트랜잭션의 양이 적은 애플리케이션에서 오류나 레이턴시가 발생한 트레이스를 포착하는 데 유용합니다. 헤드 기반 샘플링은 방대한 양의 트레이스 데이터를 실시간으로 샘플링할 수 있는 효율적인 방법이며 애플리케이션 성능에 거의 영향을 미치지 않습니다.

헤드 기반 샘플링의 장점

- 트랜잭션 처리량이 적은 애플리케이션에 적합합니다.
- 빠르고 간편하게 시작하고 실행할 수 있습니다.
- 모노리스가 우위에 있는 모노리스와 마이크로서비스 혼합 환경에 적합합니다.
- 애플리케이션 성능에 거의 또는 전혀 영향을 주지 않습니다.
- 낮은 비용으로 서드파티 공급업체에 트레이스 데이터를 전송하기 위한 솔루션입니다.
- 통계 샘플링이 분산 시스템에 대한 적절한 투명성을 제공합니다.

헤드 기반 샘플링의 단점

- 트레이스가 무작위로 샘플링됩니다.
- 샘플링은 트레이스의 경로가 여러 서비스를 통해 완전히 완료되기 전에 수행되므로 어떤 트레이스에 문제가 발생할 수 있는지 미리 알 수 없습니다.
- 처리량이 많은 시스템에서는 오류나 비정상적인 레이턴시가 있는 트레이스가 샘플링에서 누락될 수 있습니다.

테일 기반 샘플링으로 실행 가능한 트레이스 확보

테일 기반 샘플링을 통한 분산 추적은 소프트웨어 팀이 모든 트레이스 텔레메트리를 관찰하고 오류나 비정상적인 레이턴시가 포함된 트레이스를 샘플링해야 하는 고도로 분산된 대규모 시스템에서 문제를 해결하는 데 도움이 됩니다. 테일 기반 샘플링은 트레이스 완료 시 해당 트레이스에 대한 모든 정보를 수집합니다.

테일 기반 샘플링은 팀에서 문제 해결을 위해 최고 수준의 세분화가 필요한 경우 반드시 필요합니다.

일부 조직은 서비스 간 모든 흐름을 관찰하고 분석하며, 특히 피크 이벤트에서 다운타임이 발생하면 수백만 달러의 비용이 야기될 수 있으므로 문제 해결에 활용할 수 있는 최고의 트레이스 정보를 제공하는 분산 추적 툴이 필요합니다.

예를 들어, 평균 스펠 로드가 분당 300만 스펠인 조직이 신제품을 출시하는 경우 스펠이 분당 3억 개로 급증합니다. 기존의 헤드 기반 샘플링은 트랜잭션 양이 많은 이러한 유형의 조직에 적합하지 않습니다.

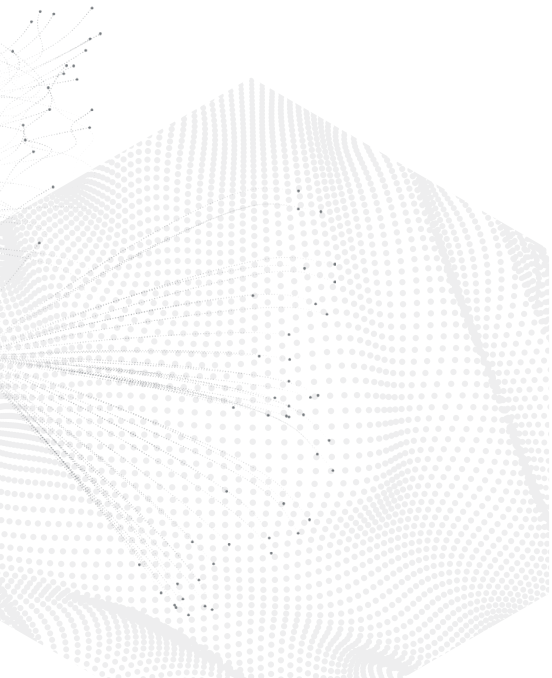
트레이스라고 모두 같은 것은 아닙니다. 최적의 샘플링 방법을 선택하려면, 팀은 사용 사례 및 비용 대비 이익 분석을 기반으로 평가하고 각 애플리케이션의 모니터링 요구 사항을 고려해야 합니다.

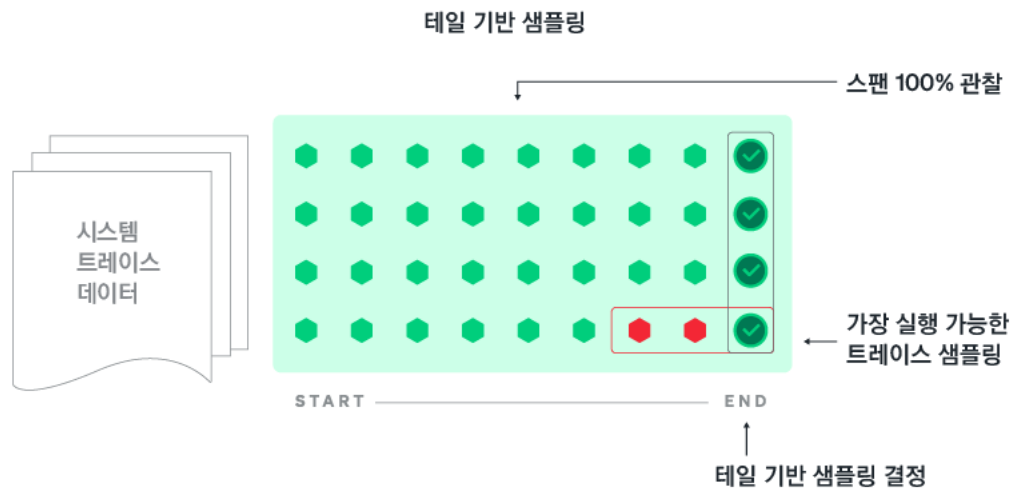
테일 기반 샘플링의 장점

- 트레이스의 100%를 관찰하고 분석할 수 있습니다.
- 트레이스가 완전히 완료된 후에 샘플링을 합니다.
- 오류나 특징이 없는 속도 저하가 있는 트레이스를 보다 빠르게 시각화해줍니다.

테일 기반 샘플링의 단점

- 샘플링 소프트웨어를 실행하려면 게이트웨이, 프록시 및 위성이 추가로 필요할 수 있습니다.
- 경우에 따라 서드파티 소프트웨어를 관리하고 확장하려면 추가적인 작업이 필요합니다.
- 더 많은 데이터를 전송 및 저장하는 데 추가 비용이 발생합니다.





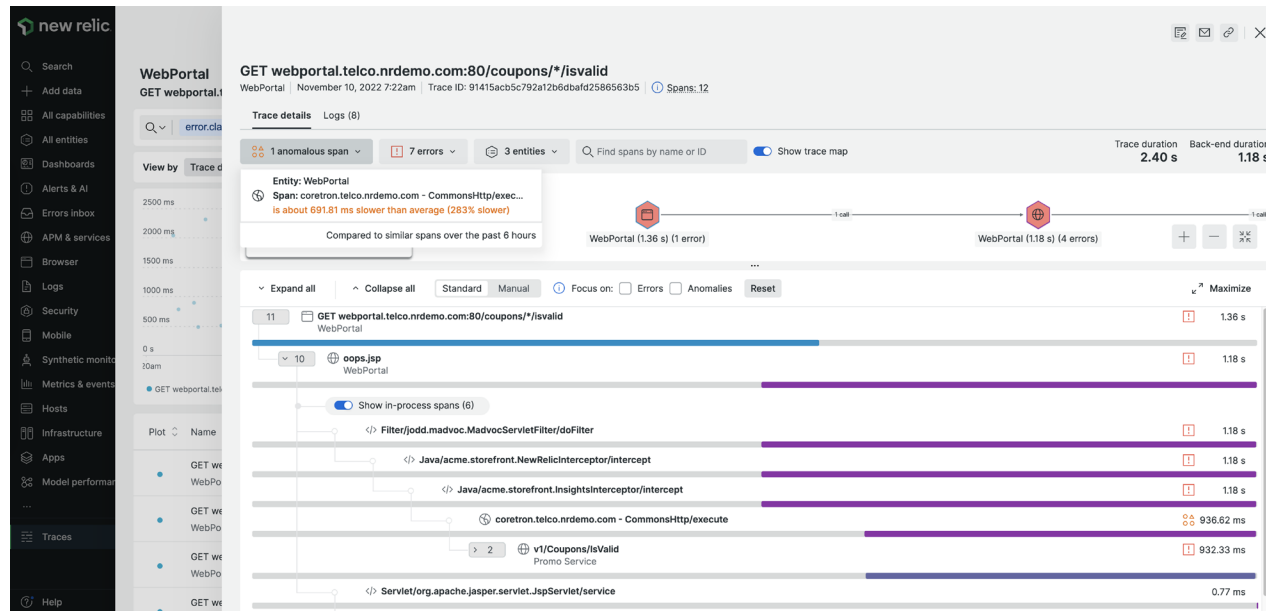
전통적인 헤드 기반 샘플링(상단) 및 테일 기반 샘플링(하단)

분석 및 시각화

소프트웨어 팀이 복잡한 아키텍처 전반에서 데이터를 쉽게 분석하고 시각화할 수 없는 경우 트레이스 데이터를 수집하는 것은 시간 낭비입니다. 포괄적인 오픈버빌리티 플랫폼은 팀이 모든 텔레메트리와 비즈니스 데이터를 한 곳에서 볼 수 있도록 해줍니다. 또한 의미를 도출하고 올바른 조치를 신속하게 취하며 의미 있는 방식으로 데이터를 사용하는 데 필요한 컨텍스트를 제공합니다.

분산 트레이스 시각화는 이상적으로 나무와 같은 구조를 가지고 있습니다. 시각화에는 하나의 부모 스패를 참조하는 여러 자식 스패가 포함되어야 하며, 이를 통해 팀은 트레이스 내에서 높은 레이턴시와 오류가 있는 스패를 확인할 수 있습니다. 시각화는 팀이 정확한 오류 세부 정보와 느린 서비스를 이해하고 문제를 찾아 신속하게 해결할 수 있는 세부 특성을 파악할 수 있도록 지원합니다.

뉴렐릭 같은 오픈버빌리티 공급업체는 이러한 시각화 구조를 문제 해결 및 분석에 사용합니다.



뉴렐릭의 분산 추적

관리 부담 해소

분산 시스템에서 문제 진단은 모래 사장에서 바늘을 찾는 것과 같습니다. 데이터를 추적하고 수집하고 시각화하기 위한 시스템을 구축하는 것은 노동 집약적이고 구현하기가 복잡합니다. 완벽하게 관리되는 서비스형 소프트웨어(SaaS) 솔루션을 통해, 데이터 수집을 위해 서드파티 게이트웨이나 위성을 배포, 관리 및 확장하는 데 따르는 부담을 제거할 수 있습니다.

뉴렐릭의 오피버빌리티 플랫폼은 거의 모든 프로그래밍 언어 및 프레임워크에서 단일 에이전트를 배포하여 애플리케이션을 쉽게 계측할 수 있도록 합니다. 또한, 계측 환경에 오픈소스 툴과 개방형 계측 표준을 사용할 수 있습니다. OpenTelemetry는 오픈소스 계측과 텔레메트리 수집의 표준으로 여겨집니다.

뉴렐릭 플랫폼은 전체 분산 시스템에서 스펠의 100%를 관찰 및 분석하고, 오류 또는 비정상적인 레이턴시를 추적할 수 있는 시각화를 제공하며, 완전하게 관리되는 데일 기반 샘플링 서비스를 제공하므로 팀은 신속하게 문제를 식별하고 해결할 수 있습니다.

플랫폼은 모든 스펠을 관찰하고 메트릭, 오류 데이터 및 필수 트레이스를 단일 뷰로 제공합니다. 가장 실행 가능한 데이터를 뉴렐릭 플랫폼에 저장하여 중요한 인사이트를 제공합니다. 그 결과, 분산된 시스템에 대한 탁월한 가시성을 확보하여 팀이 다운스트림의 레이턴시나 오류의 영향을 세부적인 메트릭으로 파악한 다음, 저장된 트레이스 데이터를 세부 분석하여 가장 관련성 높은 트레이스를 찾을 수 있습니다.

분산 추적은 뉴렐릭 APM에 포함되어 있으며, 뉴렐릭 에이전트에서 낮은 레이턴시와 낮은 비용의 데이터 전송, 서버리스 함수 내 계측 그리고 서드파티 계측 등의 다른 데이터 소스를 제공합니다.

뉴렐릭은 다음을 지원합니다.

- 온디맨드로 확장되며 완전하게 관리되는 클라우드-로컬 서비스를 제공합니다.
- 분산 시스템 전반에서 트레이스를 100% 관찰하고 분석할 수 있습니다.
- 오류 또는 비정상적인 레이턴시가 포함된 가장 실행 가능한 트레이스를 시각화해줍니다.
- 환경에서 서드파티 게이트웨이 또는 위성을 배포, 관리, 지원 및 확장하는 데 따른 수고를 제거해줍니다.
- 트레이스 텔레메트리를 위한 개방형 계측과 표준을 완전하게 지원합니다.
- 클라우드 워크로드에 근접하여 발생하는 데이터 유출 비용을 절감해줍니다.
- 보다 효율적으로 문제를 해결할 수 있습니다.
- 높은 충실도의 실행 가능한 트레이스를 통해 평균 감지 시간(MTTD)과 평균 해결 시간(MTTR)을 줄여줍니다.
- 엔지니어와 개발자가 새로운 기능 개발 같은 보다 중요한 작업에 집중할 수 있도록 지원합니다.

앞면 또는 뒷면? 동전 던지기로 결정할 필요는 없습니다.

뉴렐릭은 분산 추적을 위한 유연한 옵션을 제공하기 때문에, 팀은 애플리케이션 수준에서 헤드 또는 테일 기반 샘플링 결정을 내릴 수 있습니다. 팀이 모든 트레이스를 관찰하고 분석해야 하는 중요한 애플리케이션의 경우, 샘플링 인프라 관리에 대한 걱정 없이 테일 기반 샘플링을 선택할 수 있습니다.

뉴렐릭은 소프트웨어 팀이 헤드 기반 샘플링이나 완전 관리형 테일 기반 샘플링을 통해 분산 추적을 선택할 수 있는 유연성을 제공하는 유일한 오피저빌리티 공급업체입니다. 관리가 간단할 수록 혁신과 경쟁 우위를 확보할 시간을 더 많이 확보할 수 있습니다.



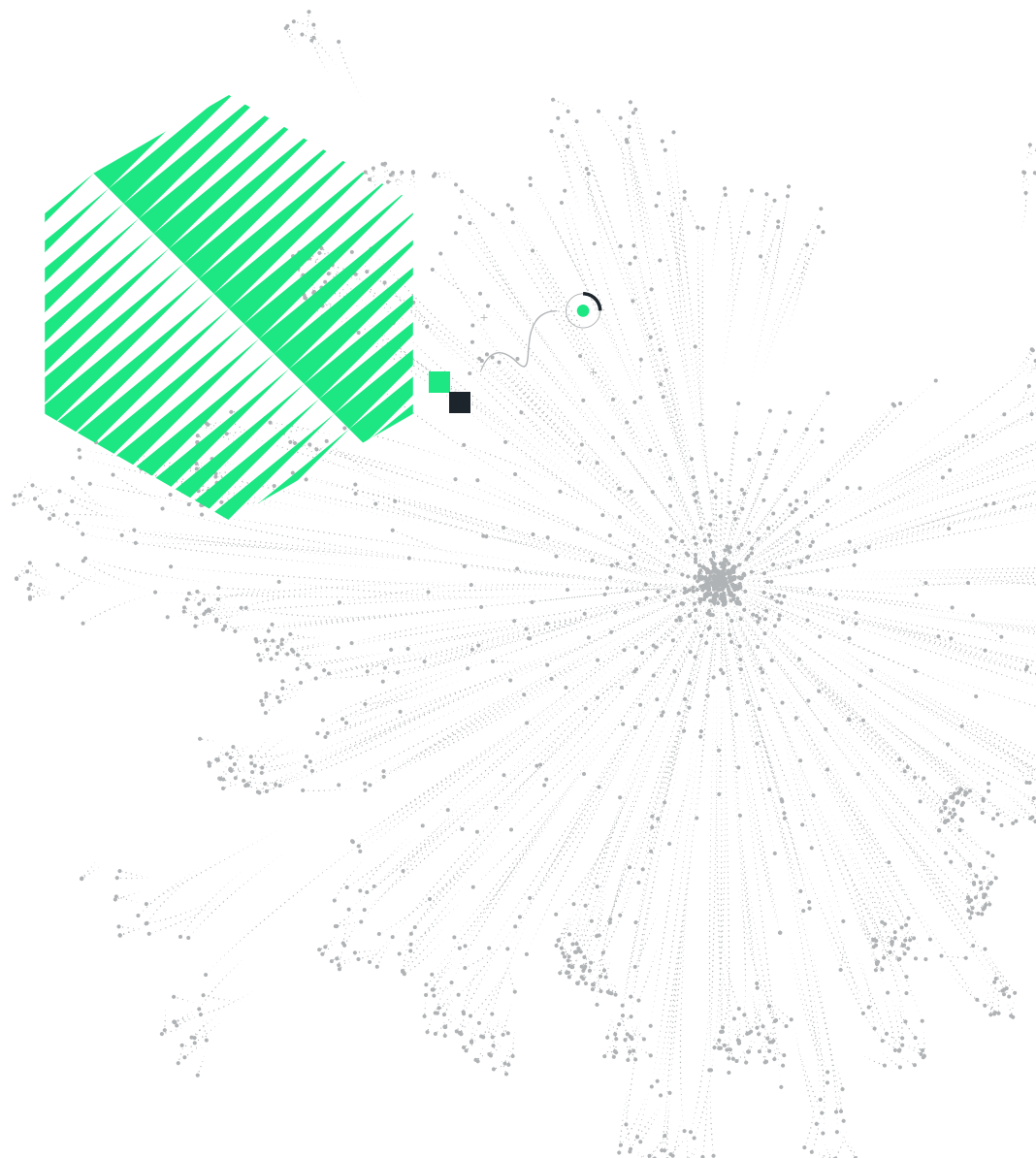
뉴렐릭 오피저빌리티 플랫폼은 로그 관리, APM, 분산 추적, 인프라 모니터링, 서버리스 모니터링, 모바일 모니터링, 브라우저 모니터링, 신세틱 모니터링, 쿠버네티스 모니터링 등을 통합합니다.

다음 단계

분산 추적과 함께 뉴렐릭 APM을 사용하려면 지금 무료 계정을 신청하십시오. 무료 계정에는 매월 100GB의 무료 데이터 수집과 1명의 전체 플랫폼 사용자 및 무제한 무료 기본 사용자가 포함됩니다.

뉴렐릭 계정이 있으신가요? 뉴렐릭 APM 분산 추적은 쉽게 시작할 수 있습니다. 최신 APM 에이전트를 사용하면 됩니다. 분산 추적 설정 옵션에 대해 알아보십시오.

지금 시작하기



뉴렐릭 소개

뉴렐릭은 오피버빌리티 분야의 리더로, 엔지니어들에게 탁월한 소프트웨어를 계획, 구축, 배포 및 실행할 수 있는 데이터 중심 접근 방식을 제공합니다. 뉴렐릭은 메트릭, 이벤트, 로그, 트레이스 등 모든 텔레메트리 데이터를 갖춘 유일한 통합 데이터 플랫폼과 강력한 풀스택 분석 툴을 결합하여 엔지니어들이 의견이 아닌 데이터에 기반해 작업을 최대한 효과적으로 수행할 수 있도록 지원합니다.

업계에서 유일하게 직관적이고 예측 가능한 사용량 기반 요금제를 제공하는 뉴렐릭은 계획 주기 시간, 변경 실패율, 릴리스 빈도, MTTR을 개선하여 엔지니어가 비용 대비 더 많은 가치를 실현할 수 있도록 지원합니다. 이러한 요금제는 세계 유수의 브랜드와 고성능 스타트업들이 업타임, 안정성 및 운영 효율성을 개선하고 혁신과 성장을 촉진하는 탁월한 고객 경험을 제공할 수 있도록 지원합니다.

