

지표 모니터링 및 경고 시스템



인프라의 상태를 선명하게 볼 수 있게 하여 높은 가용성과 안정성을 달성하자

▼ 1 설계 범위 확정



질문 목록

- 시스템의 고객은 누구인가?
 - 대형 IT 업체가 회사 내부에서 사용할 시스템인지
 - SaaS 제품 설계인지 (ex: Datadog)
- 어떤 지표를 수집해야 하는가?
 - 시스템 운영 지표
 - CPU 부하, 메모리 사용률, 디스크 사용량
 - RPS, 웹 서버 프로세스 개수
 - 사업 지표는 포함하지 않음
- 모니터링할 인프라 규모는 얼마인가?
 - DAU 1억 명
 - 1000개의 서버 풀
 - 풀마다 100개의 서버 하드웨어
- 지표 데이터는 얼마나 오래 유지해야 하는가?
 - 1년
- 데이터를 장기 보관 전용 저장소로 옮길 때 해상도 유지 여부
 - 새로 수집한 데이터는 7일 보관
 - 7일 뒤에는 1분 단위 데이터로 30일 보관
 - 그 이후에는 1시간 단위 데이터로 변환해서 보관
- 정보 채널 지원 여부
 - 이메일, 전화, 페이지듀티, 웹훅

- 에러 로그, 액세스 로그 수집 기능 제공 여부
- 분산 시스템 추적 기능 제공 여부

▼ 2 요구사항 정리

기능 요구사항

- 대규모 인프라 모니터링
 - DAU 1억명
 - 서버 풀 1000개, 풀 당 서버 수 100개
 - 서버당 100개의 운영 지표 수집 시 모니터링 해야 하는 지표 수는 1000만 개 수준
 - 데이터 보관 기간 1년
 - 수집한대로 7일 → 1분 단위 30일 → 1시간 단위 1년
- 모니터링 할 지표
 - CPU 사용률
 - 요청 수
 - 메모리 사용량
 - 메시지 큐 내의 메시지 수

비기능 요구사항

- 규모 확장성
- 낮은 응답 지연
- 안정성
- 유연성

▼ 3 개략적 설계

데이터 수집

여러 출처로부터 지표 데이터를 수집한다

데이터 모델

값 집합에 타임스탬프가 붙은 형태로 기록

사례 1) 지표 이름, 레이블, 특정 시각에 측정된 지표 데이터로 구성

사례 2) 지표 이름, 레이블 집합, 지표 값, 타임스탬프 배열

데이터 접근 패턴

- 쓰기 부하 는 막대함) 매일 1000만개 운영 지표 기록
- 읽기 부하 는 일시적으로 치솟았다가 사라짐 (시각화, 경보 서비스)

데이터 저장소

전송되어 오는 데이터를 정리하고 저장한다

RDB?

- 부하 규모에 맞추려면 전문가 수준의 튜닝 필요
- 태그/레이블에 대한 질의 지원 시 태그마다 인덱스를 지정해야 하는 문제
- 쓰기 연산이 지속적으로 발생하는 환경에서 성능 보장 X

NoSQL?

- 시계열 데이터를 처리할만한 카산드라, 빅테이블을 효과적으로 사용하려면 확장이 용이한 스키마 설계 필요
 - 내부 구조에 대한 해박한 지식 필요

→ 둘 다 비추, 시계열 데이터에 최적화된 저장소 시스템을 쓰자 !

가장 인기있는 시계열 데이터베이스는 InfluxDB , 프로메테우스

다량의 시계열 데이터를 저장하고 빠른 실시간 분석 지원

메모리 캐시와 디스크 저장소를 함께 사용

영속성, 높은 성능 요구사항 만족

😓 8CPU 코어, 32GB RAM을 갖춘 InfluxDB 서버 한 대는 초당 25만회 쓰기 연산 처리 가능

레이블 기준으로 막대한 양의 시계열 데이터를 집계하고 분석

레이블 이용 시 데이터베이스 과부하를 피하는 지침 제공

→ 범용 데이터베이스로 지표 데이터를 저장하기 어렵다.

→ 지표 데이터는 본질적으로 시계열 데이터임을 이해하고 시계열 데이터베이스에 저장할 수 있음을 이해하자

설계안

지표 출처

지표 데이터가 만들어지는 곳

지표 수집기

지표 데이터를 수집하고 시계열 데이터에 기록

시계열 데이터베이스

다량의 시계열 데이터를 분석하고 요약하는 데 적합하도록 설계된 질의 인터페이스 제공

질의 서비스

시계열 데이터베이스에 보관된 데이터를 질의하고 가져오는 과정을 돕는 서비스

경보 시스템

경보를 받아야 하는 다양한 대상, 경보 알림 전송 역할

시각화 시스템

지표를 다양한 형태의 그래프/차트로 시각화

▼ 4 상세 설계

지표 수집

플 모델

주기적으로 지표 데이터를 가져오는 지표 수집기가 핵심

지표 수집기가 알아야 할 것

- 데이터를 가져올 서비스 목록
 - 서버 안에 모든 서비스 엔드포인트의 DNS/IP 정보를 담은 파일을 두자

▲ 대규모 운영 환경에서 적용하기 어려움

- 서비스 탐색 기술(etcd, 아파치 주키퍼)을 활용하자

지표 수집 흐름(with 서비스 탐색 기술)

1. 지표 수집기는 서비스 탐색 기술에서 서비스 엔드포인트 설정 메타데이터 목록을 가져온다.
2. 지표 수집기는 사전에 합의된 엔드포인트에서 지표 데이터를 가져온다.
3. 지표 수집기는 서비스 엔드포인트 목록의 변화를 통지받기 위한 변경 이벤트 알림 콜백을 서비스 탐색 컴포넌트에 등록하거나, 주기적으로 엔드포인트 목록을 다시 가져온다.

🤖 지표 수집기 서버 한 대로 가능한가?

🙋 안되세요.

서버를 여러 대 두면 데이터를 중복해서 가져올 가능성이 있으므로 수집 서버 간 중재 메커니즘을 뒤야 한다.

안정 해시 링 을 사용하자!

안정 해시 링

- 해시 링 구간마다 해당 구간에 속한 서버로부터 생산되는 지표의 수집을 담당하는 수집기 서버를 지정
- 특정 서버의 지표 데이터는 항상 하나의 수집 서버가 처리함을 보장

푸시 모델

지표 출처에 해당하는 서버가 직접 지표를 수집기에 전송하는 모델

수집 에이전트

- 모니터링 대상 서버에 설치하면 지표 데이터를 모아 주기적으로 수집기에 전달해줌
- 직접 데이터 집계 작업을 처리하기도 데이터의 양을 줄여줌
- 수집기가 일시적으로 처리하지 못할 경우 에이전트가 보관했다가 나중에 재전송 가능
 - 오토 스케일링이 가능하다면 서버 추가/삭제 과정에서 데이터가 소실될 수 있음

무엇을 쓸까?

정답은 없다. 두 모델의 장단점을 비교하는 능력이 더 중요

- 풀 모델: 프로메테우스
- 푸시 모델: 아마존 클라우드워치, 그래파이트

풀 모델

디버깅이 더 쉬움

상태진단이 더 쉬움

생명 주기가 짧은 일괄 작업 프로세스는
수집기가 지표를 끌어가기 전에 종료될 수
있음

모든 /metrics 엔드포인트를 접근 가능하
도록 구성해야 함

TCP 사용으로 전송 지연 발생

데이터 신빙성이 높음

푸시 모델

상태 진단이 어려움

지표 수집기가 자동 규모 확장 클러스터
형태라면 어디서 오는 지표라도 수집 가능

UDP 사용으로 전송 지연이 더 낮음

아무나 지표 수집기에 데이터를 보낼 수
없음

지표 전송 파이프라인의 규모 확장

시계열 데이터베이스에 장애가 생기면 데이터 손실 발생 가능성이 있다. 큐를 두어 문제를 해소하자!

지표 수집기 - 시계열 데이터베이스 사이에 **카프카**와 소비자를 두어 규모 확장에 대응한다.

- 대역폭 요구사항에 따라 파티션 수를 설정
- 지표 이름에 따라 어떤 지표를 어느 파티션에 배치할지 결정
 - 소비자는 지표 이름에 따라 데이터 집계
- 태그/레이블에 따라 지표 데이터를 더욱 세분화한 파티션으로 나눔
- 중요 지표가 먼저 처리될 수 있도록 지표를 분류하고 우선순위 지정

카프카의 대안

페이스북의 메모리 기반 시계열 데이터베이스 시스템 **고릴라**

- 네트워크 장애 발생에도 높은 수준의 쓰기 연산 가용성 유지
- 카프카와 같은 메시지 큐 없이도 같은 수준의 안정성 제공 가능

데이터 집계 지점

수집 에이전트

복잡한 집계 로직 지원 어
려움

데이터 수집 파이프라인

스트림 프로세싱 엔진 필
요

질의 시

전체 데이터셋을 대상으로
집계 결과 계산으로 속도

늦게 도착하는 지표 데이터의 처리가 어려움
원본 데이터를 보관하지 않아 정밀도, 유연성 측면에서 손해

질의 서비스

시각화 또는 경보 시스템에서 접수된 요청을 시계열 데이터베이스를 통해 처리하는 역할

캐시 계층

시계열 데이터베이스에 대한 질의 부하를 낮추고 질의 서비스의 성능을 높이자

질의 서비스를 두면 곤란한 경우

- 이미 시계열 데이터베이스와의 연동을 처리할 강력한 플러그인이 있는 경우
- 캐시를 도입할 필요가 없는 시계열 데이터베이스가 있는 경우

시계열 데이터베이스 질의어

시계열 데이터베이스 분석에 최적화된 언어 **플렉스**

저장소 계층

저장 용량 최적화

1. 데이터 인코딩 및 압축

타임스탬프 하나를 온전히 표현하는 데는 32비트, 10을 표현하는 데는 4비트가 필요

타임스탬프의 차이를 기준으로 값을 저장하여 저장 용량을 최적화할 수 있다.

2. 다운 샘플링

데이터의 해상도를 낮춰 저장소 요구량을 줄인다.

- 7일 이내: 샘플링 적용 X
- 30일 이내: 1분 해상도로 낮춰 보관
- 1년 이내: 1시간 해상도로 낮춰 보관

3. 냉동 저장소

잘 사용되지 않는 비활성 상태 데이터를 비용이 적게 드는 곳에 보관한다.

경보 시스템

직접 만들기보다 상용품을 쓰는 게 훨씬 낫지만... 알아보자

처리 흐름

1. 설정 파일을 가져와 캐시 서버에 보관한다.
 - a. 경보 규칙은 디스크에 파일 상태로 보관
 - b. 규칙 정의는 YAML 사용
2. 경보 관리자는 경보 설정 내역을 캐시에서 가져온다.
3. 설정된 규칙에 근거하여 경보 관리자는 지정된 시간마다 질의 서비스를 호출한다.
4. 질의 결과가 설정된 임계값을 위반하면 경보 이벤트를 생성한다.
 - a. 짧은 시간동안 같은 인스턴스에서 발생한 경보는 병합
 - b. 특정 경보 관리 작업은 특정한 개인이 수행할 수 있도록 접근 제어
 - c. 경보 관리자는 경보 상태를 확인하고 알림이 최소 한 번 전달됨을 보장
5. 경보 저장소(키-값 저장소)에는 모든 경보의 상태가 보관되고 알림이 적어도 한 번 이상 전달되도록 보장하는 구실을 한다.
6. 경보 이벤트를 카프카에 전달한다.
7. 경보 소비자는 카프카에서 경보 이벤트를 읽는다.
8. 경보 소비자는 카프카에서 읽은 경보 이벤트를 처리하여 이메일, 단문 메시지, 페이지뷰티, HTTP 서비스 엔드포인트 등의 다양한 채널로 알림을 전송한다.

시각화 시스템

직접 만들기보다 상용품을 쓰는 게 훨씬 낫지만... 알아보자

데이터 계층 위에 만들어져 지표 대시보드에는 지표를 다양한 시간 범위로 표시하고 경보 대시보드에는 다양한 경보의 상태를 표시한다.

그라파나 같은 상용품을 구입해서 쓰면 상용 시계열 데이터베이스와의 궁합도 좋다 🍌

▼ 5 마무리

- 지표 데이터 수집은 폴 모델로 할 것인가, 푸시 모델로 할 것인가
- 카프카를 활용해서 어떻게 규모 확장을 지원할 수 있는가

- 좋은 시계열 데이터베이스는 어떻게 고르는가
- 다운샘플링으로 어떻게 데이터 크기를 줄이는가
- 경보/시각화 시스템은 구현할 것인가 구입할 것인가