AWS 기반 빅 데이터 분석 옵션

2016년 1월

고지 사항

이 문서는 정보 제공 목적으로만 제공됩니다. 본 문서의 발행일 기준으로 AWS 제품 및 실행방법을 설명하며, 예고 없이 변경될 수 있습니다. 고객은 본 문서에 포함된 정보나 AWS 제품 또는 서비스의 사용을 독립적으로 평가할 책임이 있으며, 각 정보 및 제품은 명시적이든 묵시적이든 어떠한 종류의 보증 없이 "있는 그대로" 제공됩니다. 본 문서는 AWS, 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 보증, 표현, 계약 약속, 조건 또는 보증을 구성하지 않습니다. 고객에 대한 AWS의 책임 및 의무는 AWS 계약에 준거합니다. 본 문서는 AWS와 고객 간의 어떠한 계약도 구성하지 않으며 이를 변경하지도 않습니다.

목차

요약

이 백서는 다음 정보와 함께 서비스 개요를 제공하여 아키텍트, 데이터 과학자, 개발자가 AWS 클라우드에서 사용할 수 있는 빅 데이터 분석 옵션을 이해하는 데 도움을 줍니다.

1. 적합한 사용 패턴
2. 비용 모델
3. 성능
4. 지속성과 가용성
5. 확장성과 탄력성
6. 인터페이스
7. 안티 패턴

이 백서는 사용 중인 분석 옵션을 소개하는 시나리오와 AWS에서 빅 데이터 분석을 시작하기 위한 추가 리소스로 끝맺습니다.

서론

디지털 사회가 되면서 생성 및 수집되는 데이터의 양이 크게 증가하고 증가 속도도 빨라지고 있습니다. 갈수록 늘어나는 데이터의 분석은 전통적인 분석 도구로는 벅찬 일이 되었습니다. 생성되는 데이터와 효과적으로 분석 가능한 데이터 사이의 간극을 메우기 위한 혁신이 필요합니다.

빅 데이터 분석 도구와 기술은 고객 선호도 이해 제고, 시장에서의 경쟁 우위 확보, 비즈니스 성장을 위해 효율적으로 데이터를 분석할 수 있는 기회와 도전을 제공합니다. 데이터 관리 아키텍처는 전통적인 데이터 웨어하우징 모델에서 실시간 및 일괄 처리, 정형 및 비정형 데이터, 고속 트랜잭션 등 더 많은 요구 사항을 처리하는 보다 복잡한 아키텍처로 진화했습니다.

Amazon Web Services(AWS)는 종단 간 빅 데이터 애플리케이션을 빠르고 쉽게 구축 및 보호하고 완벽하게 확장할 수 있는 폭넓은 관리형 서비스 플랫폼을 제공합니다. 애플리케이션에 필요한 것이 실시간 스트리밍이든 데이터 일괄 처리든 상관없이 AWS는 향후 빅 데이터 프로젝트를 위한 인프라와 도구를 제공합니다. 빅 데이터를 수집, 저장, 처리 및 분석하기 위해 하드웨어를 구입할 필요도, 인프라를 관리하고 확장할 필요도 없습니다. AWS는 급속도로 늘어나는 데이터를 처리하여 귀사에 필요한 직관력을 제공할 수 있도록 특별히 설계된 통합 분석 솔루션을 갖추고 있습니다.

빅 데이터 분석에서 AWS의 장점

대규모 데이터 세트를 분석하려면 상당한 컴퓨팅 파워이 필요하며, 그 크기는 입력 데이터의 양과 분석 유형에 따라 달라질 수 있습니다. 빅 데이터 워크로드의 이러한 특징은 수요에 따라 애플리케이션이 쉽게 확장 및 축소될 수 있는 종량 요금제 클라우드 컴퓨팅 모델에 가장 적합합니다. 요구 사항이 바뀜에 따라 필요에 맞게 AWS에서 환경의 크기를 쉽게 조정(수직적으로 또는 수평적으로)할 수 있으며, 추가 하드웨어를 기다리거나 충분한 용량을 프로비저닝하기 위한 과도한 투자를 할 필요가 없습니다.

보다 전통적인 인프라에서는 미션 크리티컬 애플리케이션의 경우, 비즈니스 요구 증가로 인한 추가 데이터 급증을 시스템이 처리할 수 있어야 하므로 시스템 설계자로서는 과다 프로비저닝 외에는 선택의 여지가 없습니다. 반면 AWS에서는 몇 분 안에 더 많은 용량과 컴퓨팅을 프로비저닝할 수 있어 수요에 따라 빅 데이터 애플리케이션이 확장 및 축소되고 시스템은 거의 최대 효율로 작동합니다.

뿐만 아니라 AWS가 제공하는 여러 다양한 지리적 리전 에 액세스하여 글로벌 인프라에서 유연한 컴퓨팅이 가능하며, 그 밖의 확장 가능한 부가 서비스를 사용하여 정교한 빅 데이터 애플리케이션을 구축할 수 있습니다.[1] 이러한 그 밖의 서비스에는 데이터 저장을 위한 Amazon Simple Storage Service(Amazon S3) 및 이러한 데이터를 쉽게 이동하고 변환하는 작업을 조율하는 AWS Data Pipeline 이 포함됩니다.[2] AWS IoT는 연결된 디바이스가 클라우드 애플리케이션 및 다른 디바이스와 상호 작용할 수 있게 해 줍니다.[3]

그 밖에도 AWS는 페타바이트 규모의 데이터 전송을 가속화하는 AWS Import/Export Snowball 및 스트리밍 데이터를 로드하는 Amazon Kinesis Firehose 같은 보안이 확실한 디바이스와 AWS Direct Connect를 통한 확장 가능한 프라이빗 연결 등 데이터를 클라우드에 올리도록 돕는 여러 옵션을 갖추고 있습니다.[4] 모바일 사용이 계속 빠르게 늘어남에 따라 AWS Mobile Hub 내의 서비스를 사용하여 앱 사용량 및 데이터를 수집 및 측정하거나 이러한 데이터를 추가 사용자 지정 분석을 위해 다른 서비스로 내보낼 수 있습니다.[5]

이와 같은 기능 때문에 AWS 플랫폼은 빅 데이터 문제 해결에 안성맞춤이며, 많은 고객들은 AWS에서 성공적으로 빅 데이터 분석 워크로드를 구현했습니다. 사례 연구에 대한 자세한 내용은 AWS 클라우드 기반 빅 데이터 및 HPC를 참조하십시오.[6]

다음 서비스들은 빅 데이터 수집, 처리, 저장, 분석 순서로 설명됩니다.

- Amazon Kinesis Streams
- AWS Lambda
- Amazon Elastic MapReduce
- Amazon Machine Learning
- Amazon DynamoDB
- Amazon Redshift
- Amazon Elasticsearch Service
- Amazon QuickSight

또한 자체 관리형 빅 데이터 애플리케이션에 Amazon EC2 인스턴스를 사용할 수 있습니다.

Amazon Kinesis Streams

Amazon Kinesis Streams를 사용하면 실시간으로 스트리밍 데이터를 처리 또는 분석하는 사용자 지정 애플리케이션을 구축할 수 있습니다. Amazon Kinesis Streams는 웹사이트 클릭스트림, 금융 거래, 소셜 미디어 피드, IT 로그 및 위치 추적 이벤트와 같은 수십만 개의 소스에서 시간당 테라바이트급의 데이터를 지속적으로 캡처 및 저장할 수 있습니다.[7]

Amazon Kinesis Client Library(KCL)를 사용하여 Amazon Kinesis 애플리케이션을 구축하고 스트리밍 데이터를 사용하여 실시간 대시보드를 지원하고, 알림을 생성하고, 동적 요금 및 광고를 구현할 수 있습니다. 또한 Amazon Kinesis Streams에서 Amazon Simple Storage Service(Amazon S3), Amazon Redshift, Amazon Elastic MapReduce(Amazon EMR), AWS Lambda 같은 다른 AWS 서비스로 데이터를 내보낼 수 있습니다.

AWS Management Console, API,[11] 또는 SDK를 사용하여 데이터 스트림에 필요한 수준의 입출력을 초당 1메가바이트(MB/sec)의 블록으로 프로비저닝하십시오.[8] 스트림의 크기는 스트림을 다시 시작할 필요 없이 데이터를 스트림에 푸시하는 데이터 소스에 영향을 주지 않고 언제든 확장 또는 축소할 수 있습니다. 몇 초 안에 스트림에 포함된 데이터를 분석에 사용할 수 있습니다.

스트림 데이터는 리전의 여러 가용 영역에 24시간 동안 저장됩니다. 이 기간 동안 데이터를 읽기, 다시 읽기, 채움, 분석에 사용하거나 장기 스토리지(예: Amazon S3 또는 Amazon Redshift)로 이동할 수 있습니다. KCL을 통해 개발자는 비즈니스 애플리케이션 작성에 집중하고 스트리밍 데이터 로드 밸런싱, 분산 서비스 조정, 내결함성 데이터 처리에 수반되는 차별화되지 않는 과중한 업무를 제거할 수 있습니다.

적합한 사용 패턴

생산자(데이터 소스)로부터 신속히 데이터를 이동해 지속적으로 처리할 필요가 있을 때마다 Amazon Kinesis Streams가 유용합니다. 이러한 처리는 다른 데이터 스토어로 내보내기 전 데이터 변환, 실시간 측정치 및 분석 구동 또는 다중 스트림 추출 및 보다 복잡한 스트림으로 집계 또는 다운스트림 처리가 될 수 있습니다. 분석에 Amazon Kinesis Streams를 사용하는 일반적인 시나리오는 다음과 같습니다.

- **실시간 데이터 분석** - Amazon Kinesis Streams에서는 웹 사이트 클릭스트림 데이터 분석 및 고객 참여 분석 등 스트리밍 데이터에 대한 데이터 분석을 실시간으로 수행할 수 있습니다.
- **로그 및 데이터 피드 인테이크 및 처리** - Amazon Kinesis Streams를 사용하면 생산자를 통해 Amazon Kinesis Streams으로 직접 데이터를 푸시할 수 있습니다. 예를 들어 몇 초 안에 시스템 및 애플리케이션 로그를 Amazon Kinesis Streams로 전송하고 스트림에 액세스하여 처리할 수 있습니다. 이렇게 하면 프런트 엔드 또는 애플리케이션 서버에 오류가 발생하더라도 로그 데이터가 손실되지 않으며 소스에서 로컬 로그 스토리지가 줄어듭니다. Amazon Kinesis Streams는 인테이크를 위해 데이터를 전송하기 전에 서버의 데이터를 일괄 처리하지 않기 때문에 가속화된 데이터 피드 인테이크를 제공합니다.
- **실시간 측정치 및 보고** - Amazon Kinesis Streams에 수집된 데이터를 실시간 속도로 측정치 추출 및 파워 보고서에 대한 KPI와 대시보드 생성에 사용할 수 있습니다. 따라서 데이터가 지속적으로 스트리밍되므로 데이터 배치가 도달할 때까지 기다리지 않고 데이터 처리 애플리케이션 논리를 데이터에 사용할 수 있습니다.

비용 모델

Amazon Kinesis Streams는 선결제 비용이나 최소 요금이 없는 간단한 종량 요금제이며 사용하는 리소스에 대해서만 비용을 지불합니다. Amazon Kinesis Streams은 하나 이상의 샤드로 구성되며, 각 샤드가 제공하는 용량은 초당 5회 읽기 트랜잭션으로 최대 초당 2MB의 데이터 읽기가 가능합니다. 각 샤드는 초당 최대 1000회 쓰기 트랜잭션 지원이 가능하고 초당 최대 총 1MB의 데이터를 쓸 수 있습니다.

스트림의 데이터 용량은 스트림에 대해 지정하는 샤드 수의 함수입니다. 스트림의 총 용량은 각 샤드 용량의 합계입니다. 요금 구성 요소는 샤드당 시간제 요금과 100만 PUT 트랜잭션당 요금 단 두 가지입니다. 자세한 내용은 Amazon Kinesis Streams 요금을 참조하십시오.[9] Amazon EC2에서 실행되고 Amazon Kinesis Streams을 처리하는 애플리케이션도 표준 Amazon EC2 요금이 발생합니다.

성능

Amazon Kinesis Streams에서는 샤드에 필요한 처리 용량을 선택할 수 있습니다. Amazon Kinesis Streams의 각 샤드는 초당 최대 1000회 쓰기 트랜잭션으로 초당 최대 1MB의 데이터를 캡처할 수 있습니다. Amazon Kinesis 애플리케이션은 초당 최대 2MB의 속도로 각 샤드에서 데이터를 읽을 수 있습니다. 필요한 만큼 샤드를 프로비저닝하여 원하는 처리 용량을 얻을 수 있습니다. 예를 들어 초당 1GB의 데이터 스트림에는 1,024개의 샤드가 필요합니다.

지속성과 가용성

Amazon Kinesis Streams는 AWS 리전의 3개 가용 영역에 데이터를 동기 복제하여 고가용성과 데이터 지속성을 제공합니다.

그 외에도 DynamoDB에 커서를 저장하여 Amazon Kinesis Streams에서 읽은 데이터를 지속적으로 추적할 수 있습니다. 스트림에서 데이터를 읽는 도중 애플리케이션에 장애가 발생하는 경우, 애플리케이션을 다시 시작하고 커서를 사용하여 장애가 발생한 애플리케이션이 중단된 정확한 지점부터 다시 읽을 수 있습니다.

확장성과 탄력성

진행 중인 스트림 처리의 중단 없이 비즈니스 또는 업무상 필요에 따라 언제든 스트림 용량을 늘리거나 줄일 수 있습니다. API 호출 또는 개발 도구를 사용하여 수요에 맞춰 Amazon Kinesis Streams 환경 조정을 자동화하고 필요한 용량만큼만 비용을 지불할 수 있습니다.

인터페이스

Amazon Kinesis Streams에는 두 가지 인터페이스가 있습니다. 데이터 생산자가 데이터를 Amazon Kinesis Streams에 넣는 데 사용하는 입력과 들어오는 데이터를 처리하고 분석하기 위한 출력이 그것입니다. 생산자는 Amazon Kinesis PUT API, AWS 소프트웨어 개발 키트(SDK) 또는 도구 키트 추상화, Amazon Kinesis Producer Library(KPL) 또는 Amazon Kinesis Agent를 사용하여 데이터를 쓸 수 있습니다.[10]

이미 Amazon Kinesis Streams에 통합된 데이터 처리의 경우, 실시간 스트리밍 데이터 처리 애플리케이션을 구축하고 작동하기 위한 클라이언트 라이브러리가 제공됩니다. KCL[17]은 Amazon Kinesis Streams와 특정 처리 논리가 포함된 비즈니스 애플리케이션 사이의 매개체 역할을 합니다.[11] Amazon Kinesis Storm Spout를 통해 Amazon Kinesis Streams에서 Apache Storm으로 읽기 통합도 이루어집니다.[12]

안티 패턴

Amazon Kinesis Streams에는 다음과 같은 안티 패턴이 있습니다.

- **소규모의 일정한 처리량** – Amazon Kinesis Streams는 200KB/sec 이하의 스트리밍 데이터에 작동하지만 이보다 큰 데이터 처리량을 감안하여 설계 및 최적화되어 있습니다.
- **장기 데이터 스토리지 및 분석** – Amazon Kinesis Streams는 장기 데이터 스토리지에 적합하지 않습니다. 기본적으로 데이터는 24시간 동안 보존되며, 7일까지 보존 기간을 연장할 수 있습니다. 7일 이상 저장해야 하는 데이터는 Amazon S3, Amazon Glacier, Amazon Redshift 또는 DynamoDB 같은 다른 장기 스토리지 서비스로 이동할 수 있습니다.

AWS Lambda

AWS Lambda를 사용하면 서버를 프로비저닝하거나 관리할 필요 없이 코드를 실행할 수 있습니다.[13] 사용한 컴퓨팅 시간에 대해서만 요금을 지불하면 되고 코드가 실행되지 않을 때는 요금이 부과되지 않습니다. Lambda를 사용하면 사실상 모든 유형의 애플리케이션 또는 백엔드 서비스를 실행할 수 있으며 이를 관리할 필요는 전혀 없습니다. 코드를 업로드하기만 하면 고가용성을 유지한 채로 코드를 실행하고 확장하는 데 필요한 모든 것을 Lambda가 알아서 처리해 줍니다. 코드가 다른 AWS 서비스에서 자동으로 트리거되도록 설정하거나 어떤 웹 또는 모바일 앱에서도 코드를 직접 호출할 수 있습니다.

적합한 사용 패턴

Lambda를 통해 데이터 변경, 시스템 상태 변화 또는 사용자 동작 같은 트리거에 응답하여 코드를 실행할 수 있습니다. Lambda는 Amazon S3, DynamoDB, Amazon Kinesis Streams, Amazon Simple Notification Service(Amazon SNS), Amazon CloudWatch 같은 AWS 서비스에 의해 직접 트리거되어 다양한 실시간 데이터 처리 시스템을 구축할 수 있습니다.

- **실시간 파일 처리** – 파일이 Amazon S3에 업로드되거나 수정된 프로세스를 호출하도록 Lambda를 트리거할 수 있습니다. 예를 들어 이미지가 Amazon S3에 업로드된 후 컬러를 회색조로 변경하려는 경우가 그렇습니다.
- **실시간 스트림 처리** - Amazon Kinesis Streams와 Lambda를 사용하여 클릭스트림 분석, 로그 필터링, 소셜 미디어 분석을 위해 스트리밍 데이터를 처리할 수 있습니다.
- **추출, 변환, 로드** - Lambda를 사용하여 데이터를 변환하고 한 데이터 리포지토리에서 다른 리포지토리로 데이터를 로드하는 작업을 실행할 수 있습니다.
- **cron 대체** – EC2 인스턴스에서 cron을 실행하는 것보다 저렴하고 가용성 높은 솔루션으로 일정 표현식을 사용하여 일정한 시간 간격으로 Lambda 함수를 실행하십시오.
- **AWS 이벤트 처리** – AWS CloudTrail 같은 다른 많은 서비스는 단순히 Amazon S3에 로그인하고 S3 버킷 알림을 사용하여 Lambda 함수를 호출함으로써 이벤트 소스 역할을 할 수 있습니다.

비용 모델

Lambda에서는 사용한 만큼만 지불하면 됩니다. 요금은 함수 요청 건수와 코드 실행 시간을 기준으로 부과됩니다. Lambda 프리 티어에는 매월 100만 건의 무료 요청과 매월 400,000GB-초의 컴퓨팅 시간이 포함됩니다. 그 다음부터는 100만 요청당 0.20달러(요청당 0.0000002달러)가 부과됩니다. 또한 코드 실행 지속 시간에 대해서는 할당된 메모리를 기준으로 요금이 부과됩니다. 사용하는 GB-초마다 0.00001667달러가 부과됩니다. 자세한 내용은 AWS Lambda 요금을 참조하십시오.[14]

성능

코드를 처음으로 Lambda에 배포한 후 함수는 일반적으로 업로드한 지 몇 초 안에 준비됩니다. Lambda는 밀리초 이내에 이벤트를 처리하도록 설계되었습니다. Lambda 함수가 생성 및 업데이트된 직후 또는 최근에 Lambda 함수를 사용한 적이 없는 경우에는 지연 시간이 늘어납니다.

지속성과 가용성

Lambda는 복제와 중복을 사용하여 서비스 자체 및 서비스가 작동하는 Lambda 함수에 고가용성을 제공하도록 설계되었습니다. 어느 경우든 유지 관리 기간 또는 예정된 가동 중단 시간이 없습니다. 장애 발생 시 호출되는 Lambda 함수는 동기식으로 예외와 함께 응답합니다. 비동기식으로 호출되는 Lambda 함수는 3회 이상 재시도되며, 그 이후에는 이벤트가 거부될 수 있습니다.

확장성과 탄력성

실행할 수 있는 Lambda 함수의 수에는 제한이 없습니다. 다만 Lambda에는 리전마다 계정당 동시 실행 100건이라는 기본 안전 조절이 있습니다. AWS 지원 팀 팀원이 이 한도를 늘릴 수 있습니다.

Lambda는 사용자를 대신하여 자동으로 확장하도록 설계되었습니다. 함수 확장에는 근본적으로 제한이 없습니다. Lambda는 수신 이벤트 속도에 맞게 용량을 동적으로 할당합니다.

인터페이스

Lambda 함수는 다양한 방법으로 관리할 수 있습니다. Lambda 콘솔에서 대시보드를 사용하여 Lambda 함수를 간편하게 나열, 삭제, 모니터링할 수 있습니다. 또한 AWS CLI와 AWS SDK를 사용하여 Lambda 함수를 관리할 수도 있습니다.

Amazon S3 버킷 알림, DynamoDB Streams, CloudWatch 로그, Amazon SES, Amazon Kinesis Streams, Amazon SNS, Amazon Cognito 등과 같은 AWS 이벤트에서 Lambda 함수를 트리거할 수 있습니다. CloudTrail을 지원하는 모든 서비스의 API 호출은 CloudTrail 감사 로그에 응답하여 Lambda에서 이벤트로 처리될 수 있습니다. 이벤트 소스에 대한 자세한 내용은 [핵심 구성 요소: AWS Lambda 함수 및 이벤트 소스]를 참조하십시오.[15]

Lambda는 Java, Node.js, Python 같은 프로그래밍 언어를 지원합니다. 코드는 기존 라이브러리를 포함할 수 있으며, 심지어 기본 라이브러리도 포함할 수 있습니다. Lambda 함수는 Bash, Go, Ruby를 비롯해 [Amazon Linux AMI]가 지원하는 언어를 사용하여 손쉽게 프로세스를 시작할 수 있습니다.[16] 자세한 내용은 [Node.js], [Python], [Java] 설명서를 참조하십시오.[17]

안티 패턴

Lambda에는 다음과 같은 안티 패턴이 있습니다.

- **장기 실행 애플리케이션** – 각각의 Lambda 함수는 300초 이내에 완료되어야 합니다. 작업이 5분 이상 실행되어야 하는 장기 실행 애플리케이션의 경우, Amazon EC2를 권장합니다. 또는 함수 1이 함수 2를 호출하고, 함수 2가 함수 3을 호출하며, 프로세스가 완료될 때까지 이런 방식이 계속되는 Lambda 함수 체인을 생성하십시오.
- **동적 웹사이트** – Lambda를 사용하여 정적 웹사이트를 실행하는 것도 가능하지만 고도로 동적인 대량 웹사이트를 실행하면 성능이 저하될 수 있습니다. Amazon EC2 및 Amazon CloudFront를 사용하는 것이 권장 사용 사례입니다.
- **상태 저장 애플리케이션** –Lambda 코드는 "상태 비저장" 스타일로 작성되어야 합니다. 즉, 기본 컴퓨팅 인프라에 선호도가 없다고 가정해야 합니다. 로컬 파일 시스템 액세스, 하위 프로세스 및 비슷한 아티팩트는 요청의 수명 주기 이상으로 연장될 수 없으며, 모든 지속적 상태는 Amazon S3, DynamoDB 또는 인터넷 사용이 가능한 다른 스토리지 서비스에 저장해야 합니다.

Amazon EMR

Amazon EMR은[25] 비용 효율적으로 쉽고 빠르게 데이터를 처리하고 저장할 수 있는 고도로 분산된 컴퓨팅 프레임워크입니다.[18] Amazon EMR은 오픈 소스 프레임워크인 Apache Hadoop을 사용하여 크기 조정이 가능한 Amazon EC2 인스턴스에 데이터 및 처리를 분산시키고, Hive, Pig, Spark 등 가장 일반적인 하둡 도구를 사용할 수 있게 해 줍니다. 하둡은 빅 데이터 처리 및 분석을 실행할 프레임워크를 제공하고, 하둡 클러스터의 인프라 및 소프트웨어 프로비저닝, 관리, 유지 관리에 수반되는 모든 과중한 업무는 Amazon EMR이 수행합니다.

적합한 사용 패턴

Amazon EMR의 유연한 프레임워크는 대규모 처리 문제 및 데이터 세트를 보다 작은 작업들로 줄여 하둡 클러스터의 여러 컴퓨팅 노드에 분산시킵니다. 이 기능은 빅 데이터 분석이 포함된 많은 사용 사례에 적합합니다. 다음은 몇 가지 예입니다.

- 로그 처리 및 분석
- 대규모 추출, 변환, 로드(ETL) 데이터 이동

- 위험 모델링 및 위협 분석
- 광고 타겟 설정 및 클릭 스트림 분석
- 유전체학
- 예측 분석
- 애드혹 데이터 마이닝 및 분석

자세한 내용은 Amazon EMR 모범 사례 백서를 참조하십시오.[19]

비용 모델

Amazon EMR을 사용하여 무기한 유지되는 지속적 클러스터 또는 분석 완료 후 종료되는 임시 클러스터를 시작할 수 있습니다. 어느 시나리오건 클러스터 작동 시간에 대해서만 요금을 지불합니다.

Amazon EMR은 다양한 Amazon EC2 인스턴스 유형(표준, 고CPU, 고메모리, 고I/O 등)과 모든 Amazon EC2 요금 옵션(On-Demand, Reserved, Spot)을 지원합니다. Amazon EMR 클러스터("작업 흐름"이라고도 함)를 시작할 때 프로비저닝할 Amazon EC2 인스턴스의 수와 유형을 선택합니다. Amazon EMR 요금은 Amazon EC2 요금에 추가됩니다. 자세한 내용은 Amazon EMR 요금을 참조하십시오.[20]

성능

Amazon EMR 성능은 클러스터 실행을 위해 선택하는 EC2 인스턴스의 유형과 분석 실행을 위해 얼마나 많은 EC2 인스턴스를 선택했는지에 따라 결정됩니다. 메모리, 스토리지 및 처리 성능이 충분하고 처리 요구 사항에 적합한 인스턴스 유형을 선택해야 합니다. EC2 인스턴스 사양에 대한 자세한 내용은 Amazon EC2 인스턴스 유형을 참조하십시오.[21]

지속성과 가용성

기본적으로 Amazon EMR은 코어 노드 장애에 대한 내결함성을 갖추고 있으며, 슬레이브 노드가 고장 나더라도 작업 실행을 계속합니다. 현재 Amazon EMR은 자동으로 다른 노드를 프로비저닝하여 장애가 발생한 슬레이브를 대체하지는 않지만 고객은 CloudWatch를 사용하여 노드 상태를 모니터링하고 장애가 발생한 노드를 대체할 수 있습니다.

가능성은 적지만 마스터 노드 장애가 발생할 경우에 대비하여 Amazon S3 같은 영구 저장소에 데이터를 백업하는 것이 좋습니다. 또는 자동 장애 조치 및 대체를 통해 여러 건의 동시 장애를 견딜 수 있는 no-NameNode 아키텍처를 제공하는 Amazon EMR with the MapR distribution을 실행할 수 있습니다.[22] 메타데이터는 데이터와 마찬가지로 분산되고 복제됩니다. no-NameNode 아키텍처의 경우, 저장할 수 있는 파일 수에 사실상 제한이 없고 외부 네트워크에 연결된 스토리지에 대한 종속성도 없습니다.

확장성과 탄력성

Amazon EMR에서는 실행 중인 클러스터 크기 조정이 쉽습니다.[23] 언제든 하둡 분산 파일 시스템(HDFS)이 있는 코어 노드를 추가해 처리 성능을 높이고 HDFS 스토리지 용량(및 처리량)을 높일 수 있습니다. 그 밖에도 Amazon S3를 기본으로 사용하거나 메모리 및 컴퓨팅을 스토리지에서 분리할 수 있도록 해 주는 로컬 HDFS와 함께(또는 대신) EMFS를 사용하여 유연성과 비용 효율성을 높일 수 있습니다.

또한 하둡 작업을 처리할 수 있지만 HDFS를 유지하지는 않는 작업 노드를 언제든 추가하고 제거할 수 있습니다. 일부 고객은 일괄 처리가 이루어질 때 클러스터에 수백 개의 인스턴스를 추가하고 처리가 완료되면 추가 인스턴스를 제거합니다. 예를 들어 6개월 안에 클러스터가 처리하게 될 데이터가 얼마나 될지 알지 못하거나 처리 수요가 급증할 수도 있습니다.

Amazon EMR을 사용하면 언제든 용량을 쉽게 추가 또는 제거할 수 있으므로 장래의 요구 사항을 추측하거나 피크 수요에 대응해 프로비저닝할 필요가 없습니다.

뿐만 아니라 콘솔에서 클릭 몇 번으로 또는 프로그래밍 방식 API 호출을 사용하여 언제든 다양한 크기의 새 클러스터를 추가하고 제거할 수 있습니다.[24]

인터페이스

Amazon EMR은 빅 데이터 분석에 사용할 수 있고 각각 자체 인터페이스를 갖춘, 하둡 기반의 여러 도구를 지원합니다. 다음은 가장 인기 있는 옵션의 간단한 요약입니다.

Hive

Hive는 하둡을 기반으로 실행되는 오픈 소스 데이터 웨어하우스 및 분석 패키지입니다. Hive는 사용자들이 데이터를 구성, 요약, 쿼리하도록 해 주는 SQL 기반 언어인 Hive QL에 의해 작동됩니다. Hive QL은 표준 SQL을 뛰어넘어 map/reduce 함수와 JSON 및 Thrift 같은 복잡하고 확장 가능한 사용자 정의 데이터 형식에 대한 최고 수준의 지원을 추가합니다. 이 기능을 통해 텍스트 문서와 로그 파일 같은 복잡한 비정형 데이터 소스 처리가 가능합니다.

Hive에서는 Java로 작성된 사용자 정의 함수를 통한 사용자 확장이 가능합니다. Amazon EMR에서는 DynamoDB 및 Amazon S3와의 직접 통합을 비롯하여 대단히 많은 Hive 개선이 이루어졌습니다. 예를 들어 Amazon EMR을 사용하면 Amazon S3에서 자동으로 테이블 파티션을 로드할 수 있고, 임시 파일을 사용하지 않고 Amazon S3의 테이블에 데이터를 쓸 수 있으며, 사용자 지정 map 및/또는 reduce 연산을 위한 스크립트와 추가 라이브러리 같은 Amazon S3의 리소스에 액세스할 수 있습니다. 자세한 내용은 *Amazon EMR 릴리스 안내서*의 Apache Hive를 참조하십시오.[25]

Pig

Pig는 하둡을 기반으로 실행되는 오픈 소스 분석 패키지입니다. Pig는 사용자들이 데이터를 구성, 요약, 쿼리하도록 해 주는 유사 SQL 언어인 Pig Latin에 의해 작동됩니다. Pig Latin은 유사 SQL 연산뿐 아니라 map 및 reduce 함수와 복잡하고 확장 가능한 사용자 정의 데이터 형식에 대한 최고 수준의 지원도 추가합니다. 이 기능을 통해 텍스트 문서와 로그 파일 같은 복잡한 비정형 데이터 소스 처리가 가능합니다.

Pig에서는 Java로 작성된 사용자 정의 함수를 통한 사용자 확장이 가능합니다. Amazon EMR에서는 여러 파일 시스템을 사용할 수 있는 기능(일반적으로 Pig는 하나의 원격 파일 시스템에만 액세스할 수 있습니다), Amazon S3에서 고객 JAR 및 스크립트를 로드할 수 있는 기능(예: “REGISTER s3://my-bucket/piggybank.jar”), String 및 DateTime 처리를 위한 추가 기능 등 대단히 많은 Pig 개선이 이루어졌습니다. 자세한 내용은 *Amazon EMR 릴리스 안내서*의 Apache Pig를[33] 참조하십시오.

Spark

Spark는 인메모리 MapReduce용 기초를 포함하여 하둡에 구축된 오픈 소스 데이터 분석 엔진입니다. Spark는 일부 분석에 추가 속도를 제공하며, Shark(SQL 구동 데이터 웨어하우징), Spark Streaming(스트리밍 애플리케이션), GraphX(그래프 시스템), MLlib(기계 학습) 같은 다른 파워 도구의 기초입니다. 자세한 내용은 Amazon EMR 클러스터에 Apache Spark 설치 블로그 게시물을 참조하십시오.[26]

HBase

HBase는 Google의 BigTable을 모델로 하는 오픈 소스 비관계형 분산 데이터베이스입니다. 이는 Apache Software Foundation의 하둡 프로젝트의 일부로 개발되었으며, 하둡용 BigTable 유사 기능을 제공하기 위해 하둡 분산 파일 시스템(HDFS)을 기반으로 실행됩니다. HBase는 컬럼 기반 압축 및 저장을 사용하여 다량의 스파스 데이터를 결함 없이 효율적으로 저장할 수 있도록 지원합니다. 또한 데이터가 디스크 대신 메모리에 저장되므로 데이터를 빠르게 조회할 수 있습니다.

HBase는 순차 쓰기 작업에 최적화되어 있으며, 배치 삽입, 업데이트 및 삭제에 매우 효율적입니다. HBase는 하둡과 완벽하게 연동되어 파일 시스템을 공유하고 하둡 작업에 대한 직접 입출력 역할을 합니다. HBase는 또한 Apache Hive와 통합되어 HBase 테이블을 통한 SQL 유사 쿼리, Hive 기반 테이블과의 조인 및 Java Database Connectivity(JDBC)에 대한 지원이 가능합니다. Amazon EMR을 사용하여 HBase를 Amazon S3에 백업(전체 또는 증분, 수동 또는 자동)할 수 있고, 이전에 생성한 백업에서 복원할 수 있습니다. 자세한 내용은 *Amazon EMR 개발자 안내서*의 HBase 및 EMR을 참조하십시오.[27]

Impala

Impala는 SQL 구문을 사용하는 대화형 임시 쿼리를 위한 하둡 에코시스템의 오픈 소스 도구입니다. Impala는 MapReduce를 사용하지 않고 전통적 RDBMS(관계형 데이터베이스 관리 시스템)의 엔진과 비슷한 MPP(대량 병렬 프로세싱) 엔진을 사용합니다. 이 아키텍처를 통해 사용자는 HDFS 또는 HBase 테이블의 데이터를 아주 빠르게 쿼리할 수 있으며, 실행 시간에 스키마를 제공하고 다양한 데이터 유형을 처리하는 하둡의 기능을 활용할 수 있습니다. 이런 이유로 Impala는 지연 시간이 짧은 대화형 분석을 수행하기에 좋은 도구입니다.

Impala에는 Java 및 C++ 사용자 정의 함수도 있으며, ODBC 및 JDBC 드라이버를 통해 BI 도구에 연결할 수 있습니다. Impala는 Hive 메타스토어를 사용하여 파티션 이름 및 데이터 유형과 같은 입력 데이터에 대한 정보를 보관합니다. 자세한 내용은 *Amazon EMR 개발자 안내서*의 Impala 및 EMR을 참조하십시오.[28]

Hunk

Hunk는 모두가 액세스하고 사용할 수 있으며 가치 있는 머신 데이터를 위해 Splunk가 개발했습니다. Hunk를 사용하면 Amazon EMR과 Amazon S3에 저장된 데이터를 대화식으로 탐색, 분석, 시각화하여 하둡에서 Splunk 분석을 활용할 수 있습니다. 자세한 내용은 Amazon EMR with Hunk: Splunk Analytics for Hadoop and NoSQL을 참조하십시오.[37]

Presto

Presto는 지연 시간이 짧은 애드혹 데이터 분석에 최적화된 오픈 소스 분산 SQL 쿼리 엔진입니다. Presto는 복잡한 쿼리, 집계, 조인, 창 함수 등 ANSI SQL 표준을 지원합니다. Presto는 하둡 분산 파일 시스템(HDFS)과 Amazon S3를 비롯한 여러 데이터 소스의 데이터를 처리할 수 있습니다.

기타 타사 도구

Amazon EMR은 R(통계), Mahout(기계 학습), Ganglia(모니터링), Accumulo(보안 NoSQL 데이터베이스), Hue(하둡 데이터 분석을 위한 사용자 인터페이스), Sqoop(관계형 데이터베이스 커넥터), HCatalog(테이블 및 스토리지 관리) 등 하둡 에코 시스템에 있는 그 밖의 다양한 인기 애플리케이션과 도구도 지원합니다.

뿐만 아니라 Amazon EMR을 기반으로 비즈니스 요구 해결에 도움이 되는 자체 소프트웨어를 설치할 수 있습니다. AWS는 MapReduce를 사용하여 대량의 데이터를 효율적으로 이동하는 오픈 소스 도구 DistCp의 확장인 Amazon EMR의 S3DistCp를 사용하여 Amazon S3에서 HDFS로, HDFS에서 Amazon S3로, Amazon S3 버킷 사이에서 대량의 데이터를 신속하게 이동하는 기능을 제공합니다.[29]

필요한 경우, Amazon EMR 클러스터가 Amazon S3에 데이터를 저장할 수 있게 해 주는 HDFS 구현인 EMR 파일 시스템(EMRFS)을 사용할 수 있습니다. Amazon S3 서버 측 및 클라이언트 측 암호화와 EMRFS 일관성 보기를 활성화할 수 있습니다. EMRFS를 사용하면 Amazon S3와의 상호 작용 관리를 돕는 메타데이터 스토어가 DynamoDB에 투명하게 구축되어 여러 EMR 클러스터가 Amazon S3에서 같은 EMRFS 메타데이터 및 스토리지를 쉽게 사용하도록 할 수 있습니다.

안티 패턴

Amazon EMR에는 다음과 같은 안티 패턴이 있습니다.

- **소규모 데이터 세트** – Amazon EMR은 대량 병렬 처리를 위해 만들어졌습니다. 데이터 세트가 단일 머신과 단일 스레드에서 빠르게 실행될 만큼 충분히 작은 경우, map 및 reduce 작업에 추가되는 오버헤드는 단일 시스템의 메모리에서 쉽게 처리할 수 있는 작은 데이터 세트에는 이점이 없을 수 있습니다.
- **ACID 트랜잭션 요구 사항** – 하둡에서 ACID(원자성, 일관성, 격리성, 내구성) 또는 제한된 ACID를 달성하는 방법도 있지만 요구 사항이 매우 엄격한 워크로드에는 Amazon RDS 또는 Amazon EC2에서 실행되는 관계형 데이터베이스 등 다른 데이터베이스가 보다 나은 옵션이 될 수 있습니다.

Amazon Machine Learning

Amazon ML은 누구나 예측 분석 및 기계 학습 기술을 손쉽게 사용할 수 있게 해 주는 서비스입니다.[30] Amazon ML이 제공하는 시각화 도구 및 마법사가 기계 학습(ML) 모델 작성 프로세스를 안내하므로 복잡한 ML 알고리즘과 기술을 배우지 않아도 됩니다. 모델이 준비된 후 Amazon ML이 API 작업을 사용하여 애플리케이션에 대한 예측을 쉽게 얻도록 지원하므로 사용자 지정 예측 생성 코드를 실행하거나 인프라를 관리할 필요가 없습니다.

Amazon ML은 Amazon S3, Amazon Redshift 또는 Amazon RDS에 저장된 데이터를 기반으로 ML 모델을 생성할 수 있습니다. 내장 마법사는 대화식으로 데이터를 탐색하는 단계를 통해 ML 모델 교육, 모델 품질 평가, 비즈니스 목표에 일치하는 출력 조정을 안내합니다. 모델이 준비된 후 배치 단위로 또는 지연 시간이 짧은 실시간 API를 사용하여 예측을 요청할 수 있습니다.

적합한 사용 패턴

Amazon ML은 데이터의 패턴을 발견하고 이러한 패턴을 사용하여 새롭고 본 적 없는 데이터 지점에서 예측을 생성할 수 있는 ML 모델 작성에 적합합니다. 예를 들어 다음을 수행할 수 있습니다.

- **의심되는 트랜잭션에 애플리케이션이 플래그 표시할 수 있도록 함** – 새 트랜잭션이 합법적인지 사기성인지 예측하는 ML 모델을 구축합니다.
- **제품 수요 예측** – 과거 주문 정보를 입력하여 장래의 주문 수량을 예측합니다.

- **애플리케이션 콘텐츠 개인화** – 사용자가 어떤 항목에 가장 관심을 가질지 예측하고 이러한 예측을 실시간으로 애플리케이션에서 검색합니다.
- **사용자 활동 예측** – 사용자 행동을 분석하여 웹사이트를 맞춤 구성하고 보다 나은 사용자 경험을 제공합니다.
- **소셜 미디어 수신** – 비즈니스 결정에 잠재적으로 영향을 미치는 소셜 미디어 피드를 수집하고 분석합니다.

비용 모델

Amazon ML에서는 사용한 만큼만 비용을 지불하며, 최소 요금 및 선수금은 없습니다. Amazon ML은 예측 모델 구축에 사용된 컴퓨팅 시간에 대해 시간당 요금을 부과하며, 사용자는 애플리케이션을 위해 생성한 예측 수에 대해 비용을 지불합니다. 실시간 예측의 경우, 모델 실행에 필요한 메모리 양을 기준으로 예약된 용량에 대한 시간당 요금도 지불합니다.

데이터 분석, 모델 교육, 평가 요금은 이를 수행하는 데 필요한 컴퓨팅 시간을 기준으로 하며, 입력 데이터의 크기, 데이터 내의 속성 수, 적용된 변환의 수와 유형에 따라 달라집니다. 데이터 분석 및 모델 구축 요금은 시간당 0.42달러입니다. 예측 요금은 배치 단위와 실시간 단위로 구분됩니다. 배치 예측은 예측 1,000건당 0.10달러(다음 1,000건으로 반올림)이며, 실시간 예측은 예측당 0.0001달러(근사값으로 반올림)입니다. 실시간 예측의 경우, 모델에 프로비저닝하는 메모리 10MB마다 시간당 예약 용량 요금 0.001달러도 적용됩니다.

모델 생성 중에 각 모델의 최대 메모리 크기를 지정하여 비용을 관리하고 예측 성능을 제어하십시오. 예약 용량 요금은 모델에서 실시간 예측이 활성화된 경우에만 지불합니다. Amazon S3, Amazon RDS 또는 Amazon Redshift에 저장된 데이터에 대한 요금은 별도로 청구됩니다. 자세한 내용은 Amazon Machine Learning 요금을 참조하십시오.[31]

성능

모델을 작성하거나 이러한 모델에서 배치 예측을 요청하는 데 소요되는 시간은 입력 데이터 레코드 수, 이러한 레코드 내 속성의 유형과 분포, 지정하는 데이터 처리 "레시피"의 복잡성에 따라 달라집니다.

대부분의 실시간 예측 요청은 100ms 내에 응답을 반환하므로 대화식 웹, 모바일 또는 데스크톱 애플리케이션에는 충분히 빠릅니다. 실시간 API가 예측을 생성하는 데 걸리는 정확한 시간은 예측을 생성하는 ML 모델에 연결된 입력 데이터 레코드의 크기와 데이터 처리 "레시피"의 복잡성에 따라 달라집니다.[32] 실시간 예측이 활성화된 각각의 ML 모델을 사용하여 기본적으로 초당 최대 200건의 트랜잭션을 요청할 수 있으며, 고객 지원에 연락하면 이 숫자를 늘릴 수 있습니다. CloudWatch 측정치를 사용하여 ML 모델이 요청하는 예측 수를 모니터링할 수 있습니다.

지속성과 가용성

Amazon ML은 고가용성을 위해 설계되었습니다. 유지 관리 기간 또는 예정에 따른 가동 중단 시간이 없습니다. 서비스는 Amazon의 입증된 고가용성 데이터 센터에서 각 AWS 리전의 3개 시설에 걸쳐 구성된 서비스 스택 복제를 사용하여 실행되므로 서버 장애 또는 가용 영역 중단 시 내결함성을 제공합니다.

확장성과 탄력성

최대 100GB 크기의 데이터 세트를 처리하여 ML 모델을 작성하거나 배치 예측을 요청할 수 있습니다. 대량의 배치 예측의 경우, 입력 데이터 레코드를 별도의 조각으로 나누면 보다 많은 예측 데이터를 처리할 수 있습니다.

기본적으로 최대 5개의 동시 작업을 실행할 수 있으며, 고객 서비스에 연락하면 이 한도를 높일 수 있습니다. Amazon ML은 관리형 서비스이므로 프로비저닝할 서버가 없습니다. 따라서 과다 프로비저닝하거나 사용하지 않는 리소스에 요금을 지불할 필요 없이 애플리케이션 확대에 따른 확장이 가능합니다.

인터페이스

데이터 소스 생성은 데이터를 Amazon S3에 추가하는 것만큼 간단합니다. 또는 Amazon Redshift 혹은 Amazon RDS가 관리하는 MySQL 데이터베이스에서 직접 데이터를 가져올 수 있습니다. 데이터 소스가 정의된 후 콘솔을 사용하여 Amazon ML과 상호 작용할 수 있습니다. Amazon ML에 대한 프로그래밍 방식 액세스는 AWS SDK와 Amazon ML API를 통해 가능합니다.[33] Windows, Mac, Linux/UNIX 시스템에서 사용할 수 있는 AWS CLI를 사용하여 Amazon ML 엔터티를 생성 및 관리할 수도 있습니다.

안티 패턴

Amazon ML에는 다음과 같은 안티 패턴이 있습니다.

- **매우 큰 데이터 세트** – Amazon ML은 최대 100GB의 데이터를 지원할 수 있지만 테라바이트 규모의 데이터 수집은 현재 지원하지 않습니다. 이러한 사용 사례의 경우, Amazon EMR을 사용한 Spark Machine Learning Library(MLlib) 실행이 일반적 도구입니다.
- **지원되지 않는 학습 작업** – Amazon ML을 사용하여 이진수 분류(두 가지 선택 중 하나를 선택하고 신뢰도를 제공), 멀티클래스 분류(두 가지 옵션 이상으로 선택을 확장) 또는 수치 회귀(하나의 숫자를 직접 예측)를 수행하는 ML 모델을 작성할 수 있습니다. 시퀀스 예측 또는 비감독 클러스터링 같은 지원되지 않는 ML 작업은 Amazon EMR을 사용하여 Spark 및 MLlib를 실행하는 방식으로 접근할 수 있습니다.

Amazon DynamoDB

Amazon DynamoDB는 빠르고 완벽하게 관리되는 NoSQL 데이터베이스 서비스로서 간단하고 비용 효율적인 방법으로 원하는 양의 데이터를 저장 및 검색하고 원하는 수준의 요청 트래픽을 처리합니다.[34] DynamoDB는 고가용성 분산 데이터베이스 클러스터를 운영하고 조정해야 하는 관리 부담을 덜어줍니다. 이 스토리지 대안은 한 자릿수 밀리초의 지연 시간과 완벽한 처리량 및 스토리지 확장성을 갖춘 예측 가능한 성능을 제공하여 고도로 까다로운 애플리케이션의 지연 시간 및 처리량 요구 사항을 충족합니다.

DynamoDB는 기본 키로 인덱싱되는 구조적 데이터를 테이블에 저장하고, 1바이트부터 400KB까지의 항목에 대한 지연 시간이 짧은 읽기 및 쓰기 액세스를 허용합니다. DynamoDB는 스칼라 및 다중 값 세트에서 세 가지 데이터 형식(숫자, 문자열, 이진수)을 지원합니다. DynamoDB는 이러한 데이터 형식으로 된 JSON, XML 또는 HTML 같은 문서 저장소를 지원합니다. 테이블에는 고정된 스키마가 없기 때문에 데이터 항목마다 속성의 수가 다를 수 있습니다. 기본 키는 단일 속성 해시 키 또는 복합 해시 범위 키일 수 있습니다.

DynamoDB는 전역 보조 인덱스 및 로컬 보조 인덱스를 제공하여 기본 키 외의 속성에 대한 쿼리가 가능한 추가적인 유연성을 제공합니다. DynamoDB는 최종적 일관된 읽기(기본값)와 강력한 일관된 읽기(선택 사항)를 모두 제공하며, 항목 넣기, 업데이트, 삭제, 조건부 작업, 증가/감소를 위한 묵시적 항목 수준 트랜잭션도 지원합니다.

DynamoDB는 Amazon EMR, Amazon Redshift, AWS Delta Pipeline, Amazon S3 같은 다른 서비스와 통합되어 분석, 데이터 웨어하우스, 데이터 가져오기/내보내기, 백업, 아카이브를 제공합니다.

적합한 사용 패턴

DynamoDB는 읽기 및 쓰기 지연 시간이 짧은 유연한 NoSQL 데이터베이스와 코드 변경이나 가동 중지 없이 필요에 따라 스토리지와 처리량을 확장 또는 축소할 수 있는 기능이 필요한 기존 애플리케이션이나 새 애플리케이션에 적합합니다.

일반 사용 사례는 다음과 같습니다.

- 모바일 앱
- 게임
- 디지털 광고 서비스
- 라이브 투표
- 라이브 이벤트를 위한 청중 상호 작용
- 센서 네트워크
- 로그 수집
- 웹 기반 콘텐츠 액세스 제어
- Amazon S3 객체용 메타데이터 스토리지
- 전자 상거래 장바구니
- 웹 세션 관리

이러한 사용 사례 상당수는 가동 중지 또는 성능 저하가 조직의 비즈니스에 즉각 악영향을 미치기 때문에 고가용성과 확장성을 갖춘 데이터베이스가 필요합니다.

비용 모델

DynamoDB에서는 사용하는 만큼만 요금을 지불하며 최소 요금이 없습니다. DynamoDB의 요금 구성 요소는 프로비저닝된 처리 능력(시간당), 인덱싱된 데이터 스토리지(매월 GB당), 양방향 데이터 전송(매월 GB당) 등 세 가지입니다. 신규 고객은 AWS 프리 티어의 일환으로 DynamoDB 사용을 무료로 시작할 수 있습니다.[35] 자세한 내용은 Amazon DynamoDB 요금을 참조하십시오.[36]

성능

SSD와 속성 인덱싱 제한은 높은 처리량과 짧은 지연 시간을 제공하며 읽기 및 쓰기 작업 비용을 대폭 줄여 줍니다.[37] 데이터세트가 커짐에 따라 워크로드의 짧은 지연 시간을 유지할 수 있는 예측 가능한 성능이 필요합니다. 이러한 예측 가능한 성능은 주어진 테이블에 필요한 프로비저닝된 처리 능력을 정의하여 달성할 수 있습니다.

백그라운드에서 서비스가 리소스 프로비저닝을 처리하여 요청된 처리 속도를 달성합니다. 사용자는 인스턴스, 하드웨어, 메모리 및 애플리케이션의 처리 속도에 영향을 미칠 수 있는 다른 요인을 생각할 필요가 없습니다. 프로비저닝된 처리량 예약은 탄력적이며, 수요에 따라 늘리거나 줄일 수 있습니다.

지속성과 가용성

DynamoDB는 고가용성을 보장하고 개별 머신의 장애, 나아가 시설 장애로부터 데이터를 보호할 수 있도록 리전의 3개 데이터 센터에 자동으로 데이터를 동기 복제하는 내결함성을 기본적으로 갖추고 있습니다.

DynamoDB Streams 는 테이블에서 이루어지는 모든 데이터 활동을 캡처하며, 다른 지리적 리전으로의 리전 복제가 가능하므로 훨씬 뛰어난 가용성을 제공합니다.[38]

확장성과 탄력성

DynamoDB는 확장성과 탄력성이 모두 뛰어납니다. DynamoDB 테이블에 저장할 수 있는 데이터 양에는 제한이 없습니다. DynamoDB 쓰기 API 연산을 사용하여 더 많은 데이터를 저장하면 서비스에서 더 많은 스토리지를 자동으로 할당합니다. 데이터는 필요에 따라 자동으로 분할 및 재분할되며, SSD 사용을 통해 어떤 규모에서도 예측 가능하고 지연 시간이 짧은 응답 시간을 제공합니다. 이 서비스는 또 요구 변화에 따라 테이블의 읽기 및 쓰기 용량을 간단히 "다이얼 업" 또는 "다이얼 다운" 할 수 있다는 점에서 탄력적입니다.[39]

인터페이스

DynamoDB는 하위 수준 REST API는 물론 하위 수준 REST API를 래핑하고 일부 객체 관계형 매핑(ORM) 함수를 제공하는 보다 상위 수준의 Java, ET, PHP용 SDK도 제공합니다. 이러한 API는 DynamoDB의 관리 및 데이터 인터페이스를 제공합니다. 현재 이 API는 테이블 관리(메타데이터 생성, 나열, 삭제, 획득) 및 속성 작업(속성 가져오기, 쓰기, 삭제 및 인덱스를 사용한 쿼리, 전체 스캔)이 가능한 연산을 제공합니다.

표준 SQL은 사용할 수 없지만 DynamoDB select 연산을 사용하여 사용자가 제공하는 기준에 따라 속성 세트를 검색하는 SQL 유사 쿼리를 생성할 수 있습니다. 콘솔을 사용하여 DynamoDB 작업을 수행할 수도 있습니다.

안티 패턴

DynamoDB에는 다음과 같은 안티 패턴이 있습니다.

- **기존 관계형 데이터베이스에 연계된 미리 작성된 애플리케이션** – 기존 애플리케이션을 AWS 클라우드로 포팅하려 하고 관계형 데이터베이스를 계속 사용해야 하는 경우, Amazon RDS(Amazon Aurora, MySQL, PostgreSQL, Oracle 또는 SQL Server) 또는 미리 구성된 여러 Amazon EC2 데이터베이스 AMI를 사용할 수 있습니다. 관리하는 EC2 인스턴스에 원하는 데이터베이스 소프트웨어를 설치할 수도 있습니다.
- **조인 또는 복잡한 트랜잭션** – 많은 솔루션이 DynamoDB를 활용하여 사용자를 지원할 수 있지만 조인, 복잡한 트랜잭션 및 기존 데이터베이스 플랫폼이 제공하는 그 밖의 관계형 인프라가 애플리케이션에 필요할 수 있습니다. 이런 경우, Amazon Redshift, Amazon RDS 또는 자체 관리형 데이터베이스가 있는 Amazon EC2에 대해 알아보는 것이 좋습니다.
- **Binary Large Object(BLOB) 데이터** – 디지털 비디오, 이미지, 음악 등 큰(400KB 이상) BLOB 데이터를 저장하려는 경우, Amazon S3를 고려하는 것이 좋습니다. 하지만 이 시나리오에서도 바이너리 객체에 대한 메타데이터(예: 항목 이름, 크기, 생성일, 소유자, 위치 등)를 기록하기 위한 DynamoDB 나름의 역할이 있습니다.
- **I/O 속도가 낮은 대규모 데이터** – DynamoDB는 SSD 드라이브를 사용하며, 저장되는 GB당 I/O 속도가 높은 워크로드에 최적화되어 있습니다. 자주 액세스하지 않는 매우 많은 양의 데이터를 저장할 경우에는 Amazon S3 등 다른 스토리지 옵션을 선택하는 것이 더 좋을 수 있습니다.

Amazon Redshift

Amazon Redshift는 빠르고 완벽하게 관리되는 페타바이트 규모의 데이터 웨어하우스 서비스로서 간편하고 비용 효율적으로 기존 비즈니스 인텔리전스 도구를 사용하여 모든 데이터를 효율적으로 분석할 수 있습니다.[40] Amazon Redshift는 수백 기가바이트부터 페타바이트 이상까지의 데이터 세트에 최적화되어 있으며, 대부분의 기존 데이터 웨어하우징 솔루션 비용의 10분의 1만 들도록 설계되었습니다.

Amazon Redshift는 컬럼 방식의 스토리지 기술을 사용하면서 여러 노드에서 쿼리를 병렬로 실행하고 분산함으로써 데이터 세트 크기에 상관없이 빠른 쿼리 및 I/O 성능을 제공합니다. 또한 데이터 웨어하우스에 대한 프로비저닝, 구성, 모니터링, 백업 및 보안과 관련된 일반적인 관리 작업을 대부분 자동화하므로 유지 및 관리가 간편하면서도 비용이 적게 듭니다. 이러한 자동화로 기존의 온프레미스 구현 시 몇 주에서 몇 달까지 걸리던 페타바이트 규모의 데이터 웨어하우스를 단 몇 분 만에 구축할 수 있습니다.

적합한 사용 패턴

Amazon Redshift는 기존 비즈니스 인텔리전스 도구를 사용한 OLAP(Online Analytical Processing)에 적합합니다. 조직에서는 Amazon Redshift를 사용하여 다음과 같은 작업을 수행하고 있습니다.

- 여러 제품에 대한 전체 판매 데이터 분석
- 주식 거래 데이터 저장
- 광고 노출 수 및 클릭 횟수 분석
- 게임 데이터 집계
- 소셜 트렌드 분석
- 의료 분야의 임상 품질, 작업 효율, 재무 성과 측정

비용 모델

Amazon Redshift 데이터 웨어하우스 클러스터는 장기 약정이나 선납 요금이 필요 없습니다. 따라서 자본 비용이 들지 않으며 데이터 웨어하우스 용량을 사전에 복잡하게 계획하고 구입할 필요도 없습니다. 요금은 클러스터의 노드 크기와 수에 따라 청구됩니다.

프로비저닝한 스토리지의 100%까지 추가 비용 없이 백업 스토리지를 제공합니다. 예를 들어 총 4TB 스토리지에 대해 XL 노드가 2개인 활성 클러스터가 하나 있을 경우, AWS는 추가 비용 없이 Amazon S3에서 최대 4TB의 백업 스토리지를 제공합니다. 프로비저닝한 스토리지 크기를 초과하는 백업 스토리지와 클러스터가 종료된 후에 저장된 백업에 대해서는 표준 Amazon S3 요율이 적용됩니다.[41] Amazon S3와 Amazon Redshift 간의 통신에 대해서는 데이터 전송 요금이 부과되지 않습니다. 자세한 내용은 Amazon Redshift 요금을 참조하십시오.[42]

성능

Amazon Redshift는 다음과 같은 다양한 혁신을 통해 수백 기가바이트에서 페타바이트 이상에 이르는 크기의 데이터 세트에 대해 매우 높은 성능을 제공합니다. Amazon Redshift는 컬럼 방식 스토리지, 데이터 압축 및 영역 매핑을 사용하여 쿼리 수행에 필요한 I/O 수를 줄입니다.

Amazon Redshift는 대량 병렬 처리(MPP) 아키텍처를 사용하므로 SQL 작업을 병렬 처리하고 분산하여 사용 가능한 리소스를 모두 활용할 수 있습니다. 기반 하드웨어는 CPU와 드라이브 간의 처리량을 최대화하는 로컬 연결 스토리지와, 노드 간의 처리량을 최대화하는 10GigE 메시 네트워크를 사용하여 고성능 데이터 처리에 맞게 설계되었습니다. 데이터 웨어하우징 필요에 따라 성능을 튜닝할 수 있습니다. AWS는 SSD가 포함된 고밀도 컴퓨팅(DC)과 고밀도 스토리지(DS) 옵션을 제공합니다.

지속성과 가용성

Amazon Redshift는 데이터 웨어하우스 클러스터에서 장애가 발생한 노드를 자동으로 검출하여 교체합니다. 데이터 웨어하우스 클러스터는 대체 노드가 프로비저닝되어 DB에 추가될 때까지 읽기 전용 모드이며, 일반적으로 이러한 재구축 작업은 단 몇 분밖에 걸리지 않습니다. Amazon Redshift를 통해 대체 노드는 바로 사용할 수 있으며 Amazon S3에서 가장 자주 액세스하는 데이터가 제일 먼저 로드되기 때문에 최대한 빠르게 데이터 쿼리를 재개할 수 있습니다.

또한 드라이브에 장애가 발생하더라도 데이터 웨어하우스는 계속 사용할 수 있습니다. Amazon Redshift는 클러스터 전체에서 데이터를 미러링하기 때문에 다른 노드의 데이터를 사용하여 장애가 발생한 드라이브를 재구축합니다. Amazon Redshift 클러스터는 하나의 가용 영역에 상주하지만 Amazon Redshift에 Multi-AZ를 설정하려는 경우, 미러를 설정한 다음 복제 및 장애 조치를 스스로 관리할 수 있습니다.[43]

확장성과 탄력성

성능이나 용량에 대한 요구 사항이 변함에 따라 콘솔에서 간단한 작업을 수행하거나 API 호출을 사용하여 데이터 웨어하우스의 노드 수와 유형을 손쉽게 변경할 수 있습니다.[44] Amazon Redshift를 사용하면 최소 한 개의 160GB 노드로 시작한 후 많은 노드를 사용하는 1페타바이트 이상의 압축된 사용자 데이터까지 확장할 수 있습니다. 자세한 내용은 *Amazon Redshift 관리 안내서*의 Amazon Redshift 클러스터 주제에서 클러스터 및 노드에 대하여 섹션을 참조하십시오.[45]

규모 조정 과정에서 Amazon Redshift는 기존의 클러스터를 읽기 전용 모드로 설정하고 선택한 크기의 새 클러스터를 프로비저닝한 후 구 클러스터의 데이터를 새 클러스터로 병렬로 복사합니다. 이 프로세스 도중에는 활성 Amazon Redshift 클러스터에 대해서만 요금을 지불합니다. 새 클러스터가 프로비저닝되는 동안에는 구 클러스터에서 계속 쿼리를 실행할 수 있습니다. 데이터가 새 클러스터로 모두 복사되면 Amazon Redshift는 자동으로 쿼리를 새 클러스터로 리디렉션하고 구 클러스터를 제거합니다.

인터페이스

Amazon Redshift에는 사용자 지정 JDBC 드라이버와 ODBC 드라이버가 있으며, 이러한 드라이버는 콘솔의 Connect Client 탭에서 다운로드할 수 있습니다. 이처럼 널리 사용되는 다양한 SQL 클라이언트를 사용할 수 있습니다. 또한 표준 PostgreSQL JDBC 드라이버와 ODBC 드라이버를 사용할 수도 있습니다. Amazon Redshift 드라이버에 대한 자세한 내용은 Amazon Redshift 및 PostgreSQL을 참조하십시오.[46]

인기 있는 BI 및 ETL 벤더를 통해 검증된 통합 사례는 무수히 많습니다.[47] 각각의 컴퓨팅 노드로 로드 및 언로드가 병렬로 시도되므로 데이터를 데이터 웨어하우스 클러스터로 수집하는 속도와 Amazon S3 및 DynamoDB와 데이터를 주고받는 속도가 극대화됩니다. Amazon Kinesis Firehose를 사용하여 Amazon Redshift로 스트리밍 데이터를 쉽게 로드함으로써 현재 이미 사용 중인 기존의 비즈니스 인텔리전스 도구 및 대시보드에서 실시간에 가까운 분석이 가능합니다. 콘솔이나 CloudWatch API 연산을 사용하여 컴퓨팅 사용률, 메모리 사용률, 스토리지 사용률 및 Amazon Redshift 데이터 웨어하우스 클러스터에 대한 읽기/쓰기 트래픽 측정치를 무료로 사용할 수 있습니다.

안티 패턴

Amazon Redshift에는 다음과 같은 안티 패턴이 있습니다.

- **소규모 데이터 세트** – Amazon Redshift는 클러스터에서의 병렬 처리를 위해 만들어졌습니다. 데이터 세트가 100기가바이트 미만일 경우, Amazon Redshift가 제공하는 모든 기능을 활용할 수 없으며 Amazon RDS가 더 적합한 솔루션이 될 수 있습니다.

- **On-Line Transaction Processing(OLTP)** – Amazon Redshift는 경제적이면서도 속도가 뛰어난 분석 기능을 생성해야 하는 데이터 웨어하우스 워크로드용으로 설계되었습니다. 속도가 빠른 트랜잭션 시스템이 필요할 경우, Amazon RDS 기반의 기존 관계형 데이터베이스 시스템이나 DynamoDB와 같은 NoSQL 데이터베이스를 선택하는 것이 좋습니다.
- **비정형 데이터** – Amazon Redshift의 데이터는 행마다 임의의 스키마 구조를 지원하는 것이 아니라 정의된 스키마를 통해 정형화되어야 합니다. 데이터가 비정형 상태일 경우, Amazon EMR에서 추출, 변환 및 로드(ETL)를 수행하여 Amazon Redshift에 로드할 수 있는 데이터를 얻을 수 있습니다.
- **BLOB 데이터** – 디지털 비디오, 이미지, 음악 등 큰 바이너리 파일을 저장하려는 경우, 데이터를 Amazon S3에 저장한 후 Amazon Redshift에서 그 위치를 참조하는 방법을 고려해 볼 수 있습니다. 이 시나리오에서 Amazon Redshift는 바이너리 객체에 대한 메타데이터(항목 이름, 크기, 생성일, 소유자, 위치 등)를 기록하지만 큰 객체 자체는 Amazon S3에 저장됩니다.

Amazon Elasticsearch Service

Amazon ES는 AWS 클라우드에서 Elasticsearch를 쉽게 배포, 운영 및 조정할 수 있는 관리형 서비스입니다.[48] Elasticsearch는 실시간 분산 검색 및 분석 엔진입니다. Elasticsearch를 통해 전에는 불가능했던 속도와 규모로 데이터를 탐색할 수 있습니다. Elasticsearch는 전체 텍스트 검색, 구조적 검색, 분석 및 이 세 가지의 조합에 사용됩니다.

콘솔을 사용하여 몇 분 만에 Amazon ES 클러스터를 설정 및 구성할 수 있습니다. Amazon ES는 요청하는 인프라 용량 프로비저닝부터 Elasticsearch 소프트웨어 설치까지 도메인 설정과 관련된 작업을 관리합니다.

도메인이 가동된 후 Amazon ES는 백업 수행, 인스턴스 모니터링 및 Amazon ES 인스턴스를 구동하는 소프트웨어 패치 적용 등 공통 관리 작업을 자동화합니다. Amazon ES는 장애가 발생한 Elasticsearch 노드를 자동으로 감지하고 교체해 자체 관리형 인프라 및 Elasticsearch 소프트웨어와 관련된 오버헤드를 줄입니다. 이 서비스를 통해 API를 한 번만 호출하거나 콘솔에서 몇 번만 클릭하여 클러스터를 쉽게 조정할 수 있습니다.

Amazon ES를 사용하면 Elasticsearch 오픈 소스 API에 바로 액세스하여 기존 Elasticsearch 환경에서 이미 사용 중인 코드와 애플리케이션을 원활하게 사용할 수 있습니다. 로그 및 기타 이벤트 데이터 처리를 돕는 오픈 소스 데이터 파이프라인인 Logstash와의 통합도 지원합니다. 또한 데이터에 대한 이해를 높여 주는 오픈 소스 분석 및 시각화 플랫폼인 Kibana에 대한 지원도 포함되어 있습니다.

적합한 사용 패턴

Amazon ES는 대량의 데이터 쿼리 및 검색에 적합합니다. 조직에서는 Amazon ES를 사용하여 다음과 같은 작업을 수행할 수 있습니다.

- 고객용 애플리케이션 또는 웹사이트의 로그 등 활동 로그 분석
- Elasticsearch로 CloudWatch 로그 분석
- 다양한 서비스와 시스템에서 오는 제품 사용 데이터 분석
- 소셜 미디어 감성과 CRM 데이터 분석 및 브랜드와 제품의 트렌드 발견
- Amazon Kinesis Streams와 DynamoDB 같은
- 다른 AWS 서비스의 데이터 스트림 업데이트 분석
- 고객에게 풍부한 검색 및 탐색 환경 제공
- 모바일 애플리케이션 사용 모니터링

비용 모델

Amazon ES에서는 사용한 컴퓨팅 및 스토리지 리소스에 대해서만 요금을 지불합니다. 최소 요금이나 선수금은 없습니다. Amazon ES 인스턴스 시간당 요금, Amazon EBS 스토리지 요금(이 옵션을 선택한 경우), 표준 데이터 전송 요금이 청구됩니다.[49]

스토리지로 EBS 볼륨을 사용하는 경우, Amazon ES에서는 볼륨 유형을 선택할 수 있습니다. Provisioned IOPS(SSD) 스토리지를 선택하는 경우, 스토리지뿐 아니라 프로비저닝하는 처리량에 대해서도 요금이 청구됩니다.[50] 그러나 사용하는 I/O에 대해서는 요금이 청구되지 않습니다. 또한 도메인의 데이터 노드에 연결된 EBS 볼륨의 누적 크기에 따라 추가 스토리지에 대한 비용을 지불하도록 선택할 수 있습니다.

Amazon ES는 Amazon ES 도메인마다 무료로 자동 스냅샷용 스토리지 공간을 제공합니다. 수동 스냅샷은 Amazon S3 스토리지 요율에 따라 요금이 청구됩니다. 자세한 내용은 Amazon Elasticsearch Service 요금을 참조하십시오.[51]

성능

Amazon ES의 성능은 인스턴스 유형, 워크로드, 인덱스, 사용되는 샤드 수, 읽기 전용 복제본, 스토리지 구성(범용 SSD 등 인스턴스 스토리지 또는 EBS 스토리지)을 비롯한 여러 요인에 따라 달라집니다. 인덱스는 여러 가용 영역의 서로 다른 인스턴스에 분산시킬 수 있는 데이터의 샤드로 구성됩니다.

영역 인식 확인란을 선택한 경우, Amazon ES는 서로 다른 가용 영역에서 샤드의 읽기 전용 복제본을 유지합니다. Amazon ES는 인덱스 저장용 고속 SSD 인스턴스 스토리지 또는 EBS 볼륨을 사용할 수 있습니다. 검색 엔진은 스토리지 디바이스를 집중 사용하여 디스크의 쿼리 및 검색 성능을 높입니다.

지속성과 가용성

도메인 생성 시에 또는 라이브 도메인 수정을 통해 영역 인식을 활성화하면 Amazon ES 도메인을 고가용성으로 구성할 수 있습니다. 영역 인식이 활성화되면 Amazon ES는 도메인을 지원하는 인스턴스를 2개의 서로 다른 가용 영역에 분산시킵니다. 그런 다음 Elasticsearch에서 복제본을 활성화하면 교차 영역 복제가 되도록 인스턴스가 자동으로 분산됩니다.

자동 및 수동 스냅샷을 통해 Amazon ES 도메인의 데이터 지속성을 구축할 수 있습니다. 스냅샷을 사용하여 미리 로드된 데이터로 도메인을 복구하거나 미리 로드된 데이터로 새 도메인을 만들 수 있습니다. 스냅샷은 안전하고 지속성 및 확장성이 뛰어난 객체 스토리지인 Amazon S3에 저장됩니다. 기본적으로 Amazon ES는 각 도메인의 일일 스냅샷을 자동으로 만듭니다. 또한 Amazon ES 스냅샷 API를 사용하여 추가 수동 스냅샷을 만들 수 있습니다. 수동 스냅샷은 Amazon S3에 저장됩니다. 수동 스냅샷은 교차 영역 재해 복구 및 추가 지속성 제공에 사용할 수 있습니다.

확장성과 탄력성

인스턴스를 추가 또는 삭제하고 Amazon EBS 볼륨을 쉽게 수정하여 데이터 증가를 수용할 수 있습니다. CloudWatch 측정치를 통해 도메인 상태를 모니터링하는 몇 줄의 코드를 작성할 수 있고, Amazon ES API를 호출하여 설정된 임계값에 따라 도메인을 확장 또는 축소할 수 있습니다. 서비스는 가동 중단 없이 조정을 실행합니다.

Amazon ES는 클러스터에 연결된 인스턴스당 1개의 EBS 볼륨(최대 크기 512GB)을 지원합니다. Amazon ES 클러스터당 최대 10개의 인스턴스를 사용하면 단일 Amazon ES 도메인에 약 5TB의 스토리지를 할당할 수 있습니다.

인터페이스

Amazon ES는 Elasticsearch API를 지원하므로 기존 Elasticsearch 환경에서 이미 사용 중인 코드, 애플리케이션 및 인기 도구도 완벽하게 작동합니다.[52] AWS SDK에서는 모든 Amazon ES API 연산을 지원하므로 선호하는 기술을 사용하여 도메인을 쉽게 관리하고 상호 작용할 수 있습니다. AWS CLI 또는 콘솔을 도메인 생성 및 관리에 사용할 수도 있습니다.

Amazon ES는 Amazon S3, Amazon Kinesis Streams, DynamoDB Streams의 스트리밍 데이터 등 일부 AWS 서비스와의 통합을 지원합니다. 이 통합이 클라우드상에서 이벤트 핸들러로 사용하는 Lambda 함수는 새 데이터에 응답하여 데이터를 처리하고 Amazon ES 도메인으로 스트리밍합니다. Amazon ES는 CloudWatch와도 통합되어 Amazon ES 도메인 측정치 및 CloudTrail 모니터링을 통해 Amazon ES 도메인에 대한 구성 API 호출을 감사합니다.

Amazon ES에는 오픈 소스 분석 및 시각화 플랫폼인 Kibana에 대한 지원이 포함되어 있으며, 로그 및 기타 이벤트 데이터 처리를 돕는 오픈 소스 데이터 파이프라인인 Logstach와의 통합을 지원합니다. Logstash 구현을 통해 들어오는 모든 로그의 백엔드 스토어로 Amazon ES 도메인을 설정하면 다양한 소스의 정형 및 비정형 데이터를 쉽게 수집할 수 있습니다.

안티 패턴

Amazon ES에는 다음과 같은 안티 패턴이 있습니다.

- **온라인 트랜잭션 처리(OLTP)**– Amazon ES는 실시간 분산 검색 및 분석 엔진입니다. 데이터 조작에서의 트랜잭션 또는 처리에 대한 지원은 없습니다. 속도가 빠른 트랜잭션 시스템이 필요할 경우, Amazon RDS 기반의 기존 관계형 데이터베이스 시스템이나 DynamoDB와 같은 기능을 제공하는 NoSQL 데이터베이스를 선택하는 것이 더 좋습니다.
- **페타바이트 스토리지** – Amazon ES 클러스터당 최대 10개의 인스턴스를 사용하면 단일 Amazon ES 도메인에 약 5TB의 스토리지를 할당할 수 있습니다. 워크로드가 이보다 큰 경우, Amazon EC2에서 자체 관리형 Elasticsearch를 사용하는 것을 고려해 보십시오.

Amazon QuickSight

2015년 10월, AWS는 Amazon QuickSight 프리뷰를 선보였습니다. Amazon QuickSight는 간편하게 가상화를 구축하고 애드혹 분석을 수행하고 데이터로부터 신속하게 비즈니스 통찰을 이끌어낼 수 있는 빠른 클라우드 기반 비즈니스 인텔리전스(BI) 서비스입니다.

QuickSight는 새로운 초고속 병렬 인 메모리 계산 엔진(SPICE)을 사용하여 고급 계산을 수행하고 시각화를 빠르게 렌더링합니다. QuickSight는 AWS 데이터 서비스와 자동으로 통합되어 조직이 수십만 명의 사용자에 맞게 조정할 수 있도록 하며, SPICE의 검색 엔진을 통해 속도와 응답성이 뛰어난 쿼리 성능을 제공합니다. 기존 솔루션의 1/10 비용에 불과한 QuickStart는 조직 내 모든 이에게 합리적 가격으로 BI 기능을 제공할 수 있습니다. 프리뷰에 대해 자세히 알아보고 가입하려면 Amazon QuickSight를 참조하십시오.[53]

Amazon EC2

AWS 가상 머신 역할을 하는 인스턴스가 포함된 Amazon EC2는 AWS 인프라에서 자체 관리형 빅 데이터 분석 애플리케이션을 운영하기에 적합한 플랫폼을 제공합니다.[54] Linux 또는 Windows 가상 환경에 설치할 수 있는 거의 모든 소프트웨어는 Amazon EC2에서 실행 가능하며, 종량 요금제 모델을 사용할 수 있습니다. 이 백서에서 언급된 다른 서비스에서 제공하는 애플리케이션 수준 관리 서비스는 제공되지 않습니다. 다음은 자체 관리형 빅 데이터 분석의 여러 옵션 중 일부입니다.

- MongoDB 같은 NoSQL 제품
- 데이터 웨어하우스 또는 Vertica 같은 컬럼 형식 스토어
- 하둡 클러스터
- Apache Storm 클러스터
- Apache Kafka 환경

적합한 사용 패턴

- **특수 환경** – 사용자 지정 애플리케이션, 표준 하둡 세트의 변형 또는 AWS의 다른 제품 중 하나에 포함되지 않는 애플리케이션을 실행할 때 Amazon EC2는 컴퓨팅 요구를 충족할 수 있는 유연성과 확장성을 제공합니다.

- **규정 준수 요구 사항** – 일부 규정 준수 요구 사항의 경우, 관리형 서비스 제품 대신 Amazon EC2에서 애플리케이션을 직접 실행해야 할 수 있습니다.

비용 모델

Amazon EC2는 수많은 인스턴스 패밀리의 다양한 인스턴스 유형(표준, 고CPU, 고메모리, 고I/O 등)과 다양한 요금 옵션(On-Demand, Reserved, Spot)을 갖추고 있습니다. 애플리케이션 요구 사항에 따라서는 직접 연결된 영구 스토리지로 Amazon Elastic Block Store(Amazon EBS), 또는 지속적 객체 스토어로 Amazon S3 등의 추가 서비스를 Amazon EC2와 함께 사용하는 것이 좋습니다. 각각의 서비스에는 자체 요금 모델이 적용됩니다. Amazon EC2에서 자체 빅 데이터 애플리케이션을 실행하는 경우, 자체 데이터 센터에서와 마찬가지로 라이선스 수수료는 사용자가 부담합니다. AWS Marketplace는 간단히 버튼 클릭으로 시작할 수 있게 미리 구성된 다양한 타사 빅 데이터 소프트웨어 패키지를 다수 제공합니다.[55]

성능

Amazon EC2에서의 성능은 빅 데이터 플랫폼에 대해 선택하는 인스턴스 유형이 결정합니다. 인스턴스 유형마다 CPU, RAM, 스토리지, IOPS 양과 네트워킹 용량이 다르므로 애플리케이션 요구 사항에 적합한 성능 수준을 선택할 수 있습니다.

지속성과 가용성

인스턴스 또는 데이터 센터 장애가 애플리케이션 사용자에게 영향을 미치지 않도록 중요한 애플리케이션은 AWS 리전 내의 여러 가용 영역에 걸쳐 있는 클러스터에서 실행되어야 합니다. 가동 시간이 중요하지 않은 애플리케이션의 경우, 애플리케이션을 Amazon S3에 백업했다가 인스턴스 또는 영역 장애가 발생하는 경우, 리전 내 어느 가용 영역으로도 복원할 수 있습니다. 실행 중인 애플리케이션과 요구 사항에 따라 애플리케이션 미러링과 같은 다른 옵션도 있습니다.

확장성과 탄력성

Auto Scaling은 사용자가 정의하는 조건에 따라 Amazon EC2 용량을 자동으로 확장하거나 축소할 수 있는 서비스입니다.[56] Auto Scaling을 사용하면 수요 급증 시 사용 중인 Amazon EC2 인스턴스 수를 확장하여 성능을 유지하고 수요가 줄 때는 인스턴스 수를 자동으로 축소하여 비용을 최소화합니다. Auto Scaling은 사용량이 시간, 일, 주 단위로 바뀌는 애플리케이션에 특히 적합합니다. Auto Scaling은 CloudWatch를 통해 활성화되며, CloudWatch 요금 외에 추가 비용이 발생하지 않습니다.

인터페이스

Amazon EC2는 API, SDK 또는 콘솔을 통해 프로그래밍 방식으로 접속할 수 있습니다. 콘솔이나 CloudWatch API 연산을 통한 컴퓨팅 사용률, 메모리 사용률, 스토리지 사용률, 네트워크 소비 및 인스턴스에 대한 읽기/쓰기 트래픽 측정치는 무료입니다.

Amazon EC2를 기반으로 실행하는 빅 데이터 분석 소프트웨어의 인터페이스는 선택하는 소프트웨어의 특징에 따라 달라집니다.

안티 패턴

Amazon EC2에는 다음과 같은 안티 패턴이 있습니다.

- **관리형 서비스** – 빅 데이터 분석에서 인프라 계층과 관리를 추상화하는 관리형 서비스 제품이 필요한 경우, Amazon EC2에서 자체 분석 소프트웨어를 관리하는 이 "DIY(Do It Yourself)" 모델은 적절한 선택이 아닐 수 있습니다.
- **전문성 또는 리소스 부족** – 조직에 해당 시스템을 위한 고가용성 서비스를 설치하고 관리할 리소스 또는 전문성이 없거나 관련 비용을 지출할 의사가 없는 경우, Amazon EMR, DynamoDB, Amazon Kinesis Streams 또는 Amazon Redshift 같은 AWS의 유사 서비스 사용을 고려해야 합니다.

AWS에서 빅 데이터 문제 해결

이 백서에서는 AWS에서 빅 데이터 분석에 사용할 수 있는 몇 가지 도구를 살펴봤습니다. 이는 빅 데이터 애플리케이션 설계를 시작할 때 좋은 기준이 됩니다. 하지만 구체적인 사용 사례에 적합한 도구를 선택할 때 추가로 고려해야 할 측면이 있습니다. 일반적으로 분석 워크로드마다 다음과 같은 일정한 특징과 요구 사항이 있으며, 이것이 사용할 도구를 결정합니다.

- 분석 결과가 얼마나 빨리 필요한가, 즉 실시간, 몇 초 또는 한 시간이 적절한가?
- 이러한 분석이 조직에 얼마나 많은 가치를 제공하며 존재하는 예산 제약은 무엇인가?
- 데이터가 얼마나 크며, 증가 속도는 어느 정도인가?
- 데이터는 어떻게 정형화되어 있는가?
- 생산자와 소비자가 가지고 있는 통합 기능은 무엇인가?

- 생산자와 소비자 사이에서 허용되는 지연 시간은 얼마인가?
- 가동 중단의 비용은 얼마인가 또는 솔루션의 가용성과 지속성은 어느 정도인가?
- 분석 워크로드가 일정한가 탄력적인가?

이러한 특징 또는 요구 사항 하나하나는 사용할 도구에 대한 올바른 방향을 잡는 데 도움이 됩니다. 일부 경우, 일련의 요구 사항을 기초로 빅 데이터 분석 워크로드를 서비스 중 하나에 단순히 매핑할 수 있습니다. 하지만 현실의 빅 데이터 분석 워크로드 대부분은 동일한 데이터 세트에서도 상이하고 때로는 상충하는 특징과 요구 사항이 많습니다.

예를 들어 어떤 결과 세트는 사용자가 시스템과 상호 작용하는 실시간이어야 하는 반면 일괄 처리하여 일일 단위로 실행할 수 있는 분석도 있습니다. 동일한 데이터 세트에서 이렇게 서로 다른 요구 사항은 분리하여 둘 이상의 도구를 사용해 해결해야 합니다. 같은 도구 세트에서 위의 두 가지 예를 모두 해결하려 한다면 과다 프로비저닝으로 인해 불필요한 응답 시간에 과도하게 비용을 지불하게 되거나 사용자에게 실시간으로 응답할 만큼 빠르지 않은 솔루션을 얻게 됩니다. 각각의 개별적 분석 문제 세트에 가장 적합한 도구를 짝지으면 컴퓨팅 및 스토리지 리소스를 가장 비용 효율적으로 사용할 수 있습니다.

빅 데이터가 꼭 "큰 비용"을 뜻하지는 않습니다. 따라서 애플리케이션을 설계할 때는 비용 효율적으로 설계하는 것이 중요합니다. 만약 다른 대안과 비교해 비용 효율적이지 않다면 적합한 선택이 아닌 것입니다. 또 다른 일반적인 착각은 빅 데이터 문제 해결을 위해 여러 도구 세트를 두는 것이 한 가지 큰 도구를 두는 것보다 비용이 더 많이 들거나 관리하기 힘들다는 것입니다. 같은 데이터 세트에서의 요구 사항이 서로 다른 위의 예에서 실시간 요청은 CPU 요구 사항은 낮지만 I/O 요구 사항은 높은 반면 보다 느린 처리 요청은 컴퓨팅 리소스를 아주 많이 사용할 수 있습니다. 이 두 가지를 분리하면 비용이 훨씬 적게 들고 관리하기가 쉬워지는데 정확한 사양에 맞게 각 도구를 설계할 수 있고 과다 프로비저닝을 피할 수 있기 때문입니다. AWS의 종량 과금제 및 IaaS(Infrastructure-as-a-Service) 모델 사용에 대해서만 비용을 지불하는 방식을 활용하면 이는 훨씬 큰 가치로 연결됩니다. 단 한 시간 안에 배치 분석을 실행할 수 있어서 해당 시간의 컴퓨팅 리소스에 대해서만 요금을 지불하면 되기 때문입니다. 또 이 방식은 모든 요구 사항을 충족하려 하는 단일 시스템을 활용하는 것보다 관리하기도 더 쉽습니다. 하나의 도구로 서로 다른 요구 사항을 해결하려는 것은 둥근 구멍(큰 데이터 웨어하우스)에 네모난 말뚝(실시간 요청)을 박는 격입니다.

AWS 플랫폼은 서로 다른 도구로 같은 데이터 세트를 분석함으로써 쉽게 아키텍처를 분리할 수 있습니다. AWS 서비스에는 병렬화를 사용하여 데이터 하위 집합을 한 도구에서 다른 도구로 아주 쉽게 이동할 수 있는 통합이 기본적으로 포함되어 있습니다. 이제 몇 가지 현실적인 빅 데이터 분석 시나리오와 AWS의 아키텍처 솔루션을 살펴봄으로써 이를 실행해 보겠습니다.

예 1: 엔터프라이즈 데이터 웨어하우스

매장이 천 곳이 넘고, 백화점과 할인 매장을 통해 특정 의류 상품을 판매하며, 온라인 매장을 가지고 있는 다국적 의류 기업이 있습니다.

현재 기술적 관점에서 이러한 세 가지 채널은 독립적으로 운영되며, 관리진, POS(Point-of-Sale) 시스템, 회계 부서도 각각 다릅니다. 이런 데이터 세트를 모두 취합하여 CEO에게 전체 사업에 대한 포괄적인 분석 자료를 제공하는 시스템은 없습니다. CEO는 회사의 채널을 전사적인 관점에서 파악하고 필요할 경우, 다음과 같은 특별 분석을 수행할 수 있기를 바랍니다. 이 회사가 원하는 예시 분석은 다음과 같습니다.

- 전체 유통 채널의 추세는 어떠한가?
- 전체 채널에서 어느 지역이 실적이 좋은가?
- 회사의 광고와 쿠폰이 얼마나 효과가 있는가?
- 각 의류 라인의 추세는 어떠한가?
- 회사의 매출에 영향을 미칠 수 있는 외부적인 요인은 무엇인가(예: 실업률 또는 기후 조건)?
- 매장의 특성이 판매에 어떤 영향을 주는가(예: 직원/관리진의 근속 기간, 길가의 개방형 매장과 건물 내 매장의 비교, 매장 내 상품의 위치, 프로모션, 진열대 및 전시 공간, 판매 권유, 매장 내 디스플레이 등)?

엔터프라이즈 데이터 웨어하우스는 이 문제를 해결하는 훌륭한 방법입니다. 이 데이터 웨어하우스는 3개 채널의 다양한 각각의 시스템과 날씨 및 경제 데이터와 같은 공개 기록으로부터 데이터를 수집해야 합니다. 각 데이터 소스는 데이터 웨어하우스에서 사용하는 데이터를 매일 전송합니다. 각 데이터 소스의 구조가 서로 다를 수 있으므로 추출, 변환 및 로드(ETL) 프로세스를 수행하여 데이터를 공통된 구조로 다시 정형화합니다. 그런 다음 모든 소스의 데이터에 대해 동시에 분석을 수행할 수 있습니다. 이 프로세스를 위해 다음 예시에 나온 데이터 흐름 아키텍처가 사용됩니다.

엔터프라이즈 데이터 웨어하우스

1. 이 프로세스의 첫 단계는 다양한 여러 소스의 데이터를 Amazon S3로 모으는 것입니다. Amazon S3가 선택된 이유는 다양한 소스의 데이터를 매우 경제적인 비용으로 동시에 쓸 수 있는, 지속성이 우수하고 경제적이면서도 확장 가능한 스토리지 플랫폼이기 때문입니다.
2. Amazon EMR은 소스 형식의 데이터를 대상 및 형식으로 변환하고 정리하는 데 사용됩니다. Amazon EMR에는 Amazon S3이 통합되어 있어서 클러스터의 각 노드에서 Amazon S3와 주고받는 작업을 병렬 스레드로 동시에 처리합니다. 일반적으로 데이터 웨어하우스는 야간에 여러 다양한 소스에서 새 데이터를 받습니다. 한밤중에는 이러한 분석 기능이 필요하지 않기 때문에 이 변환 프로세스의 유일한 요구 사항은 CEO와 다른 업무 관계자들이 결과를 필요로 하는 아침까지 완료하는 것입니다. 이 요구 사항은 곧 Amazon EC2 스팟 시장을 활용하여 변환 비용을 더욱 줄일 수 있다는 뜻이 됩니다.[57] 좋은 스팟 전략은 자정에 매우 낮은 가격으로 입찰을 시작한 후 용량이 부여될 때까지 점차적으로 가격을 올리는 것일 수 있습니다. 마감 시간이 다가오는데 스팟 입찰이 아직 낙찰되지 않았을 경우, 온디맨드 가격으로 후퇴하여 완료 시간 요건을 맞출 수 있습니다. 각 소스는 Amazon EMR에서 서로 다른 변환 프로세스를 가질 수 있지만, AWS 선불형 종량 요금제를 사용하면 각 변환에 대해 별도의 Amazon EMR 클러스터를 생성한 후 정확한 성능이 되도록 조정하여 다른 작업의 리소스와 경합 없이 가능한 최저 가격으로 모든 데이터 변환 작업을 완료할 수 있습니다.
3. 그런 다음 각 변환 작업은 형식을 변환하여 정리한 데이터를 Amazon S3에 로드합니다. Amazon S3를 여기서 다시 사용하는 이유는 Amazon Redshift가 각 노드의 여러 스레드에서 병렬로 이 데이터를 사용할 수 있기 때문입니다. Amazon S3의 이 위치는 기록 레코드 역할도 하며, 시스템 사이의 형식이 지정된 SOT(Source of Truth)입니다. 시간이 지남에 따라 추가적인 요구 사항이 발생할 경우 Amazon S3의 데이터를 다른 분석용 도구에서 사용할 수 있습니다.

4. Amazon Redshift는 데이터를 테이블에 로드하고, 정렬 및 분산, 압축하여 분석 쿼리가 효율적으로 동시에 실행될 수 있도록 합니다. Amazon Redshift는 데이터 웨어하우스 워크로드를 위해 만들어졌으며, 시간이 지남에 따라 데이터 크기가 커지고 비즈니스가 확장되면 다른 노드를 추가하여 손쉽게 확장할 수 있습니다.
5. 분석을 시각화하려면 Amazon QuickSight 또는 OBDC나 JDBC를 사용하여 Amazon Redshift에 연결하는 파트너 사의 다양한 시각화 플랫폼 중 하나를 사용하면 됩니다. 이 시점에서 CEO와 경영진은 보고서와 그래프를 볼 수 있습니다. 이제 경영진은 이 데이터를 통해 회사 리소스에 대해 보다 나은 의사결정을 할 수 있으며, 더 나아가서 수익과 주주 가치를 향상시킬 수 있습니다.

이 아키텍처는 매우 유연하며, 사업이 확장되거나 데이터 소스가 추가되거나 신규 채널이 개설되거나 고객용 모바일 애플리케이션이 출시될 경우, 손쉽게 확장할 수 있습니다. 언제든 Amazon Redshift 클러스터에서 노드 수를 늘려 몇 번의 클릭으로 추가 도구를 통합하고 웨어하우스를 조정할 수 있습니다.

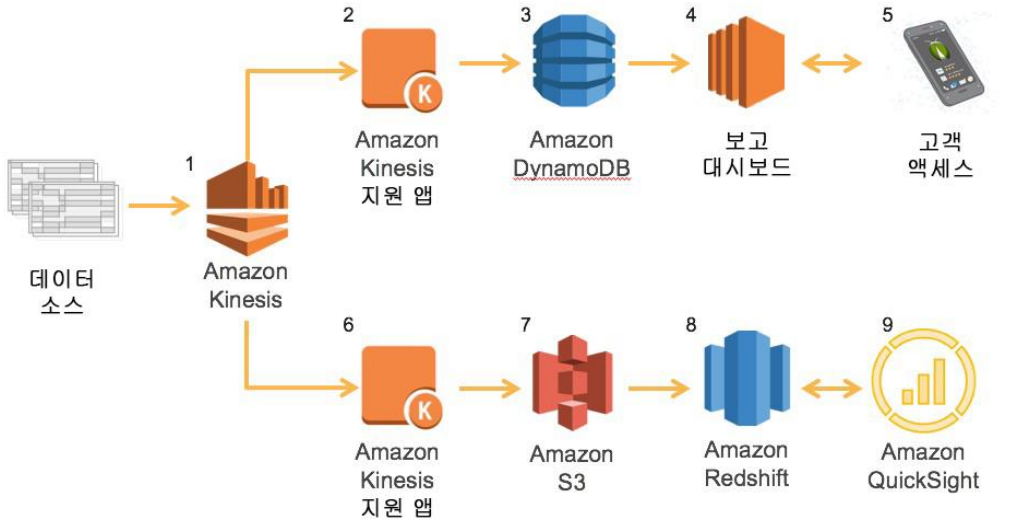
예 2: 센서 데이터 캡처 및 분석

다양한 기업을 상대로 대형 에어컨을 대량 판매하는 국제적인 에어컨 제조업체가 있습니다. 이 회사는 에어컨 장치만 판매하는 것이 아니라 경쟁사보다 앞서기 위해 모바일 애플리케이션이나 웹 브라우저에서 실시간 대시보드를 볼 수 있는 애드온 서비스도 제공합니다. 각 장치가 전송하는 센서 정보가 처리 및 분석됩니다. 이 데이터는 제조업체와 고객이 사용합니다. 이 기능을 통해 제조업체는 데이터세트를 시각화하고 추세를 파악할 수 있습니다.

현재 이 회사는 이 기능을 갖춘 장치를 수천 대 사전 판매했습니다. 회사는 2개월 안에 제품을 고객들에게 전달할 계획이며, 얼마 안 가 전 세계적으로 수천 대의 장치가 이 플랫폼을 사용하기를 바라고 있습니다. 만약 성공한다면 이 제품을 소비자 라인에까지 확장할 생각이며, 그렇게 된다면 매출과 시장 점유율은 훨씬 높아질 것입니다. 이 솔루션은 대량의 데이터를 처리할 수 있어야 하며 비즈니스 성장에 따라 중단 없이 확장될 수 있어야 합니다. 이러한 시스템은 어떻게 설계해야 할까요? 일단 같은 데이터에서 비롯되는 다음과 같은 두 개의 작업 흐름으로 나눠야 합니다.

- 실시간에 가까운 요구 사항과 정보를 소비하는 수많은 고객이 포함된 A/C 장치의 현재 정보.
- 내부용 추세 파악 및 분석 실행을 위한 A/C 장치에 대한 모든 과거 정보.

다음 예시의 데이터 흐름 아키텍처는 이 빅 데이터 문제를 해결하는 방법을 보여 줍니다.



센서 데이터 캡처 및 분석

1. 이 프로세스는 Amazon Kinesis Streams에 일정한 데이터 스트림을 제공하는 각각의 에어컨 장치에서 시작됩니다. 이는 더 많은 A/C 장치가 판매되고 작동됨에 따라 원활하게 확장될 수 있는, 장치와 대화가 가능한 탄력적이고 지속적인 인터페이스를 제공합니다.
2. Kinesis Client Library 또는 SDK와 같은 Amazon Kinesis Streams가 제공하는 도구를 사용하면 Amazon Kineis Streams에서 들어오는 데이터를 읽고 분석하여 데이터를 실시간 대시보드에 업데이트할지 결정하는 간단한 애플리케이션이 Amazon EC2에 구축됩니다. 이 애플리케이션은 시스템 작동 변화, 온도 변동 및 장치의 오류를 파악합니다.
3. 장치에 문제가 있을 경우, 고객과 유지 관리 팀이 최대한 빨리 알 수 있도록 이 데이터 흐름은 거의 실시간으로 이루어져야 합니다. 대시보드의 데이터에는 일부 집계된 추세 정보가 있기는 하지만 현재 상태 및 시스템 오류가 주를 이룹니다. 따라서 대시보드를 채울 데이터는 상대적으로 적습니다. 또한 다음과 같은 소스에서 이 데이터에 대한 많은 잠재적 액세스가 이루어집니다.
    - 모바일 디바이스나 브라우저를 통해 시스템을 확인하는 고객
    - 제품군 전체의 상태를 확인하는 유지 관리 팀

- 데이터 및 인텔리전스 알고리즘과 보고 플랫폼의 분석은 예컨대 A/C 팬이 비정상적으로 오래 작동하는데도 건물 온도가 내려가지 않는 경우, 경보로 전송할 수 있는 추세를 파악합니다.

이러한 거의 실시간에 가까운 데이터 세트 저장을 위해 DynamoDB가 선택된 것은 가용성과 확장성이 모두 뛰어나 플랫폼 채택과 사용이 늘어남에 따라 이 데이터에 대한 처리량을 쉽게 확장하거나 축소하여 소비자의 요구를 충족할 수 있기 때문입니다.

4. 보고 대시보드는 이 데이터 세트를 기반으로 구축되어 Amazon EC2에서 실행되는 사용자 지정 웹 애플리케이션입니다. 이 대시보드는 시스템 상태 및 추세에 기반하여 콘텐츠를 제공하고 장치에 발생할 수 있는 문제를 고객 및 유지 관리 담당자에게 알립니다.
5. 고객은 모바일 디바이스나 웹 브라우저에서 데이터에 액세스하여 시스템의 현재 상태를 파악하고 과거 추세를 시각화합니다.

정보를 인간 소비자에게 거의 실시간으로 보고하기 위해 방금 설명한 데이터 흐름(2-5단계)이 구축됩니다. 이 데이터 흐름은 짧은 지연 시간을 목표로 구축 및 설계되었으며, 요구에 맞춰 아주 빠르게 조정할 수 있습니다. 다이어그램의 아랫부분에 나온 데이터 흐름(6-9단계)에는 이와 같은 엄격한 속도 및 지연 시간 요구 사항이 없습니다. 이에 따라 아키텍트는 정보 바이트당 훨씬 적은 비용으로 보다 많은 데이터를 보관할 수 있는 다른 솔루션을 설계하고 보다 저렴한 컴퓨팅 및 스토리지 리소스를 선택할 수 있습니다.

6. Amazon Kinesis Streams에서 데이터를 읽기 위해 보다 조정 속도가 느린 보다 작은 EC2 인스턴스에서 실행되는 별도의 Amazon Kinesis 지원 애플리케이션이 있습니다. 이 애플리케이션은 위쪽의 데이터 흐름과 동일한 데이터 세트를 분석하지만 이 데이터의 궁극적 용도는 장기 레코드로 저장하고 데이터 웨어하우스에서 데이터 세트를 호스팅하는 것입니다. 이 데이터 세트의 모든 데이터는 시스템에서 전송되어 실시간에 가까운 요구 사항 없이 훨씬 광범위한 분석을 수행할 수 있도록 합니다.
7. 데이터는 Amazon Kinesis 지원 애플리케이션에 의해 장기 스토리지, 데이터 웨어하우스로의 로드, Amazon S3에의 저장에 적합한 형식으로 변환됩니다. Amazon S3의 데이터는 Amazon Redshift에 대한 병렬 수집 지점 역할을 할 뿐 아니라 이 시스템을 통해 실행되는 모든 데이터를 보관하는 장기 스토리지이므로 단일 SOT(Source of Truth)가 될 수 있습니다. 추가적인 요구 사항이 발생할 경우, 다른 분석 도구를 로드하는 데도 사용할 수 있습니다. Amazon S3는 데이터를 장기 저비용 콜드 스토리지로 전환해야 하는 경우, Amazon Glacier와의 기본 통합도 제공합니다.

8. 보다 큰 데이터 세트의 경우, Amazon Redshift가 데이터 웨어하우스로 다시 사용됩니다. 데이터 세트가 더 커지면 클러스터에 다른 노드를 추가하여 쉽게 확장할 수 있습니다.
9. 분석을 시각화하려면 OBDC/JDBC와 Amazon Redshift의 연결을 통해 파트너 사의 다양한 시각화 플랫폼 중 하나를 사용하면 됩니다. 여기서 보고서, 그래프, 애드혹 분석을 데이터 세트에서 수행하여 A/C 장치의 성능 저하나 고장을 초래할 수 있는 일부 변수와 추세를 찾을 수 있습니다.

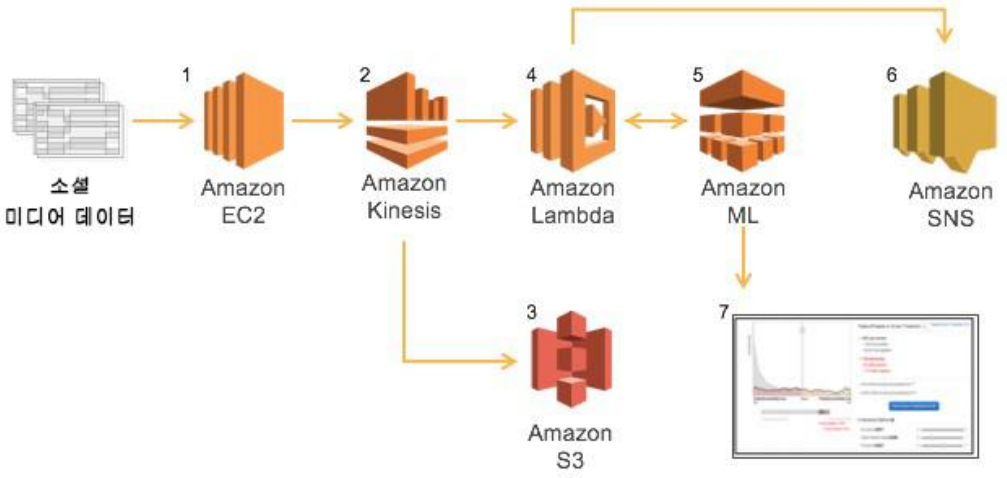
이 아키텍처는 소규모로 시작하여 필요에 따라 확장할 수 있습니다. 그뿐 아니라 2개의 작업 흐름을 서로 분리하면 사전 약정 비용 없이 필요에 따라 각 작업 흐름이 고유의 속도로 확장할 수 있으므로 제조업체는 대규모 투자 없이도 이 새 제품의 성패를 평가할 수 있습니다. Amazon ML 같은 서비스를 추가하면 A/C 장치의 사용 연한을 정확히 예측하고 이 예측 알고리즘을 기반으로 미리 유지 관리 팀을 보내 고객들에게 최상의 서비스와 경험을 제공할 수 있다는 것도 쉽게 상상할 수 있습니다. 이런 수준의 서비스는 경쟁사와 차별화하는 요소이며 장래 매출 증가로 이어집니다.

예 3: 소셜 미디어의 감성 분석

급성장 중인 대형 장난감 제조업체가 제품 라인을 확장하고 있습니다. 새 장난감이 출시될 때마다 이 회사는 제품에 대한 고객들의 반응이 궁금합니다. 또 자사 제품을 통해 소비자들이 좋은 경험을 하기를 바랍니다. 장난감 에코시스템이 커지면서 이 회사는 자사 제품이 고객들에게 여전히 유의미한지, 고객 피드백을 기초로 장래 로드맵을 계획할 수 있는지 확인하고 싶습니다. 이 회사가 소셜 미디어에서 포착하려는 것은 다음과 같습니다.

- 소비자들이 자사 제품을 사용하는 방식 이해
- 고객 만족도 확인
- 장래 로드맵 계획

다양한 소셜 네트워크에서 데이터를 캡처하는 것은 비교적 쉽지만 프로그래밍 방식으로 인텔리전스를 구축하기는 쉽지 않습니다. 이 회사는 데이터가 수집된 후 비용 효율적인 프로그래밍 방식으로 이 데이터를 분석 및 분류할 수 있기를 바랍니다. 이를 위해 다음 예시의 아키텍처를 사용할 수 있습니다.



소셜 미디어 감성 분석

제일 먼저 할 일은 어느 소셜 미디어를 청취할지 결정하는 것입니다. 그런 다음 해당 API를 통해 이 사이트들을 폴링하는 애플리케이션을 만들어 Amazon EC2에서 실행합니다.

다음으로 Amazon Kinesis Streams이 만들어지는데, Twitter, Tumblr 등등 데이터 소스가 많을 수 있기 때문입니다. 이렇게 하면 새 데이터 소스가 추가될 때마다 새 스트림을 만들 수 있고, 기존 애플리케이션 코드와 아키텍처를 사용할 수 있습니다. 또한 이 예에서는 원시 데이터를 Amazon S3로 복사하기 위해 새 Amazon Kinesis Streams이 생성됩니다.

원시 데이터는 보관, 장기 분석, 기록 참조를 위해 Amazon S3에 저장됩니다. Amazon S3에 위치한 데이터에서 추가적인 Amazon ML 배치 모델을 실행하여 예측 분석을 수행하고 소비자 구매 추세를 추적할 수 있습니다.

아키텍처 다이어그램에서 언급했듯이 데이터 처리 및 정규화와 Amazon ML로부터의 예측 요청에는 Lambda가 사용됩니다. Amazon ML 예측이 반환된 후 Lambda 함수는 예측을 기반으로 작업을 수행할 수 있습니다(예: 추가 검토를 위해 소셜 미디어 게시물을 고객 서비스 팀으로 라우팅).

Amazon ML은 입력 데이터에서 예측을 하는 데 사용됩니다. 예를 들어 ML 모델을 구축해 소셜 미디어 댓글 분석을 통해 고객이 제품에 대해 부정적 감정을 표현했는지 파악할 수 있습니다. Amazon ML을 사용하여 정확한 예측을 얻으려면 교육 데이터부터 시작하여 ML 모델이 제대로 작동하는지 확인하십시오. ML 모델을 처음 만드는 경우, 자습서: Amazon ML을 사용한 마케팅 반응 예측을 참조하십시오.[58] 앞서 언급했듯이 여러 소셜 네트워크 데이터 소스를 사용하는 경우에는 예측 정확도를 위해 각 데이터 소스마다 다른 ML 모델을 추천합니다.

끝으로 Lambda를 사용하여 실행 가능한 데이터를 Amazon SNS로 전송하고 텍스트 또는 이메일을 통해 적절한 리소스로 전달해 추가 조사합니다.

감성 분석의 일환으로 정기적으로 업데이트되는 Amazon ML 모델 생성은 정확한 결과를 얻는 데 매우 중요합니다. 정확도, 오탐지율, 정밀도, 리콜 등 특정 모델에 대한 추가 측정치는 콘솔을 통해 그래픽으로 표시할 수 있습니다. 자세한 내용은 4단계: ML 모델 예측 성능 검토 및 구분 기준 설정을 참조하십시오.[59]

Amazon Kinesis Streams, Lambda, Amazon ML, Amazon SES의 조합을 사용하여 확장 가능하고 쉽게 사용자 지정할 수 있는 소셜 리스닝 플랫폼을 만들었습니다. 이 그림은 ML 모델 생성 작업을 설명하지 않는다는 점에 유의하십시오. 이 작업은 한 번 이상 수행되지만 보통은 모델을 최신으로 유지하기 위해 정기적으로 수행됩니다. 새 모델 생성 빈도는 워크로드에 따라 달라지며, 실제로는 상황이 바뀔 때 모델의 정확도를 높이기 위해서만 생성됩니다.

결론

생성되고 수집되는 데이터가 늘어날수록 적시에 통찰을 제공할 수 있는 확장 가능하고 유연하며 성능이 뛰어난 도구가 데이터 분석에 필요합니다. 하지만 조직들은 새로운 도구가 등장했다가 금방 "사라지는", 갈수록 커지는 빅 데이터 에코시스템과 마주하고 있습니다. 따라서 뒤처지지 않으면서 올바른 도구를 선택하는 일은 매우 어려울 수 있습니다.

이 백서는 이런 문제의 해결을 돕는 첫 단계를 제공합니다. AWS 플랫폼은 빅 데이터를 수집, 처리, 분석하는 광범위한 관리형 서비스를 통해 빅 데이터 애플리케이션을 보다 쉽게 구축, 배포, 조정할 수 있으므로 사용자는 이러한 도구의 업데이트와 관리보다는 비즈니스 문제에 집중할 수 있습니다.

AWS는 빅 데이터 분석 요구 사항을 해결할 수 있는 여러 솔루션을 제공합니다. 대부분의 빅 데이터 아키텍처 솔루션은 여러 AWS 도구를 사용하여 완전한 솔루션을 구축합니다. 이는 비용 최적화, 성능, 복원력을 극대화하여 가장 엄격한 비즈니스 요구를 충족하는 데 도움이 될 수 있습니다. 따라서 AWS의 글로벌 인프라에서 비즈니스와 함께 확장할 수 있는 유연한 빅 데이터 아키텍처를 구축할 수 있습니다.

기고자

다음은 이 문서의 작성에 도움을 준 개인 및 조직입니다.

- Erik Swensson, Amazon Web Services 솔루션 아키텍처 매니저
- Erik Dame, Amazon Web Services 솔루션 아키텍트
- Shree Kenghe, Amazon Web Services 솔루션 아키텍트

참고 문헌

다음은 AWS에서 빅 데이터 분석을 시작하는 데 도움이 되는 리소스입니다.

- AWS 기반 빅 데이터[60]

  종합적인 빅 데이터 서비스 포트폴리오와 AWS 빅 데이터 파트너, 자습서, 기사 및 AW Marketplace 빅 데이터 솔루션 제품 링크를 확인해 보십시오. 도움이 필요한 경우, 문의하십시오.[61]

- AWS 빅 데이터 블로그를 읽어 보십시오.[62]

  이 블로그에서는 빅 데이터를 수집, 저장, 정리, 처리 및 시각화하는 데 도움이 되는 현실적 사례와 아이디어가 주기적으로 업데이트됩니다.

- 빅 데이터 테스트 드라이브 중 하나를 시험해 보십시오.[63]

  AWS를 사용하여 빅 데이터 문제를 해결하도록 설계된 풍부한 제품 에코시스템을 살펴보십시오. 테스트 드라이브는 AWS 파트너 네트워크(APN) 컨설팅 및 기술 파트너가 개발한 것으로서 교육, 데모, 평가 목적으로 무료 제공됩니다.

- AWS 교육 과정을 수강해 보십시오.[64]

AWS 기반 빅 데이터 과정에서는 클라우드 기반 빅 데이터 솔루션과 Amazon EMR을 소개합니다. Amazon EMR을 통해 Pig와 Hive 등 광범위한 하둡 도구 에코시스템을 사용하여 데이터를 처리하는 방법을 살펴봅니다. 또 빅 데이터 환경을 구축하고, DynamoDB와 Amazon Redshift 작업을 하며, Amazon Kinesis Streams의 장점을 이해하고, 빅 데이터 환경 설계 모범 사례를 활용하여 보안 및 비용 효율성을 높이는 방법도 배웁니다.

- 빅 데이터 고객 사례 연구를 살펴보십시오.[65]

  AWS 클라우드에서 강력하고 성공적인 빅 데이터 플랫폼을 구축한 다른 고객의 경험을 배워 보십시오.

문서 수정

날짜	설명
2016년 1월	Amazon Machine Learning, AWS Lambda, Amazon Elasticsearch Service에 대한 정보를 추가하여 개정, 일반 업데이트.
2014년 12월	최초 발행

참고

[1] http://aws.amazon.com/about-aws/globalinfrastructure/

[2] https://aws.amazon.com/s3/

http://aws.amazon.com/datapipeline/

[3] https://aws.amazon.com/iot/

[4] https://aws.amazon.com/importexport/

http://aws.amazon.com/kinesis/firehose

https://aws.amazon.com/directconnect/

[5] https://aws.amazon.com/iot/

[6] http://aws.amazon.com/solutions/case-studies/big-data/

[7] https://aws.amazon.com/kinesis/streams

[8] http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html

http://docs.aws.amazon.com/aws-sdk-php/v2/guide/service-kinesis.html

[9] http://aws.amazon.com/kinesis/pricing/

[10] http://aws.amazon.com/tools/

http://docs.aws.amazon.com/kinesis/latest/dev/developing-producers-with-kpl.html

http://docs.aws.amazon.com/kinesis/latest/dev/writing-with-agents.html

[11] https://github.com/awslabs/amazon-kinesis-client

[12] https://github.com/awslabs/kinesis-storm-spout

[13] https://aws.amazon.com/lambda/

[14] https://aws.amazon.com/lambda/pricing

[15] http://docs.aws.amazon.com/lambda/latest/dg/intro-core-components.html

[16] https://aws.amazon.com/amazon-linux-ami/

[17] http://docs.aws.amazon.com/lambda/latest/dg/nodejs-create-deployment-pkg.html

http://docs.aws.amazon.com/lambda/latest/dg/lambda-python-how-to-create-deployment-package.html

http://docs.aws.amazon.com/lambda/latest/dg/lambda-java-how-to-create-deployment-package.html

[18] http://aws.amazon.com/elasticmapreduce/

[19]

https://media.amazonwebservices.com/AWS_Amazon_EMR_Best_Practices.pdf

[20] http://aws.amazon.com/elasticmapreduce/pricing/

[21] http://aws.amazon.com/ec2/instance-types/

[22] http://aws.amazon.com/elasticmapreduce/mapr/

[23]

http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-manage-resize.html

[24] http://docs.aws.amazon.com/ElasticMapReduce/latest/API/Welcome.html

[25] http://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/emr-hive.html

[26] http://blogs.aws.amazon.com/bigdata/post/Tx15AY5C50K70RV/Installing-Apache-Spark-on-an-Amazon-EMR-Cluster

[27] http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-hbase.html

[28] http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-impala.html

[29] http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/UsingEMR_s3distcp.html

[30] https://aws.amazon.com/machine-learning/

[31] https://aws.amazon.com/machine-learning/pricing/

[32] http://docs.aws.amazon.com/machine-learning/latest/dg/suggested-recipes.html

[33] http://docs.aws.amazon.com/machine-learning/latest/APIReference/Welcome.html

[34] https://aws.amazon.com/dynamodb

[35] http://aws.amazon.com/free/

[36] http://aws.amazon.com/dynamodb/pricing/

[37] 서버 측 평균 응답 시간은 보통 10 밀리초 미만입니다.

[38] http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html

[39] DynamoDB 를 통해 UpdateTable API 연산 호출 한 번으로 프로비저닝된 처리량 수준을 100%까지 변경할 수 있습니다. 처리량을 100% 이상 늘리려면 UpdateTable 을 다시 호출하십시오. 프로비저닝된 처리량은 원할 때마다 늘릴 수 있지만 줄이는 것은 하루에 두 번으로 제한됩니다.

[40] http://aws.amazon.com/redshift/

[41] http://aws.amazon.com/s3/pricing/

[42] http://aws.amazon.com/redshift/pricing/

[43] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

[44] http://docs.aws.amazon.com/redshift/latest/APIReference/Welcome.html

[45] http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html#rs-about-clusters-and-nodes

[46] http://docs.aws.amazon.com/redshift/latest/dg/c_redshift-and-postgres-sql.html

[47] http://aws.amazon.com/redshift/partners/

[48] https://aws.amazon.com/elasticsearch-service/

[49] https://aws.amazon.com/ec2/pricing/

[50] https://aws.amazon.com/ebs/details/

[51] https://aws.amazon.com/elasticsearch-service/pricing/

[52] https://aws.amazon.com/elasticsearch-service/faqs/

[53] https://aws.amazon.com/quicksight

[54] https://aws.amazon.com/ec2/

[55] https://aws.amazon.com/marketplace

[56] http://aws.amazon.com/autoscaling/

[57] http://aws.amazon.com/ec2/spot/

[58] http://docs.aws.amazon.com/machine-learning/latest/dg/tutorial.html

[59] http://docs.aws.amazon.com/machine-learning/latest/dg/step-4-review-the-ml-model-predictive-performance-and-set-a-cut-off.html

[60] http://aws.amazon.com/big-data

[61] https://aws.amazon.com/marketplace

http://aws.amazon.com/big-data/contact-us/

[62] http://blogs.aws.amazon.com/bigdata/

[63] https://aws.amazon.com/testdrive/bigdata/

[64] http://aws.amazon.com/training/course-descriptions/bigdata/

[65] http://aws.amazon.com/solutions/case-studies/big-data/