Data warehousing su AWS

Marzo 2016

Note

Il presente documento è fornito a solo scopo informativo. In esso sono illustrate le attuali offerte di prodotti e le prassi di AWS alla data di pubblicazione del documento, offerte che sono soggette a modifica senza preavviso. È responsabilità dei clienti effettuare una propria valutazione indipendente delle informazioni contenute nel presente documento e dell'uso dei prodotti o dei servizi di AWS, ciascuno dei quali viene fornito "così com'è", senza garanzie di alcun tipo, né esplicite né implicite. Il presente documento non dà origine a garanzie, rappresentazioni, impegni contrattuali, condizioni o assicurazioni da parte di AWS, delle sue società affiliate, dei suoi fornitori o dei licenzianti. Le responsabilità di AWS nei confronti dei propri clienti sono definite dai contratti AWS e il presente documento non costituisce parte né modifica qualsivoglia contratto tra AWS e i suoi clienti.

Contenuti

Sintesi

Data engineer, analisti e sviluppatori di aziende di tutto il mondo stanno valutando la possibilità di migrare il data warehousing nel cloud per aumentare le prestazioni e ridurre i costi. Questo whitepaper esamina un approccio moderno nei confronti dell'analisi e dell'architettura del data warehousing, illustra i servizi disponibili in Amazon Web Services (AWS) per l'implementazione di tale architettura e fornisce modelli di progettazione comuni per realizzare soluzioni di data warehousing con l'ausilio di tali servizi.

Introduzione

Nel mondo di oggi, i dati e l'analisi sono elementi indispensabili per il business. Quasi tutte le grandi imprese hanno realizzato strutture di data warehousing per i rapporti e l'analisi, utilizzando i dati provenienti da numerose fonti, compresi i propri sistemi di elaborazione delle transazioni e altri database.

Tuttavia, la realizzazione e la gestione di un data warehouse, ovvero un repository centrale di informazioni provenienti da una o più origini dati, sono sempre state complicate e onerose. La maggior parte dei sistemi di data warehousing è complessa da implementare, costa milioni di dollari in spese iniziali per il software e l'hardware e ci possono volere mesi per i processi di pianificazione, approvvigionamento, implementazione e distribuzione. Dopo l'investimento iniziale e la realizzazione del data warehouse, sarà necessario un team di amministratori di database per continuare a eseguire rapidamente le query e prevenire perdite di dati.

I data warehouse tradizionali, inoltre, hanno una scalabilità limitata. Quando i volumi di dati crescono o occorre rendere analisi e rapporti disponibili per un maggior numero di utenti, si deve scegliere tra un rallentamento nell'elaborazione delle query o un upgrade oneroso in termini di costi, tempo ed energie. Alcuni team IT, di fatto, scoraggiano l'aumento dei dati o l'aggiunta di query a tutela dei contratti sul livello di servizio esistenti. Molte imprese hanno difficoltà a mantenere un sano rapporto con i fornitori di database tradizionali. Spesso sono costrette a effettuare l'upgrade hardware di un sistema gestito oppure ad avviare un lungo ciclo negoziale per una licenza a termine scaduta. Una volta raggiunto il limite di dimensionamento su un motore di data warehousing, sono costrette a migrare a un altro motore dello stesso fornitore con una semantica SQL diversa.

Amazon Redshift ha cambiato il modo in cui le aziende vedono il data warehousing, riducendo drasticamente il costo e l'impegno necessari per la distribuzione di questo tipo di sistemi, senza compromettere caratteristiche o prestazioni. Amazon Redshift è una soluzione rapida e interamente gestita di data warehousing di scala petabyte che consente di analizzare in modo semplice e conveniente grandi volumi di dati grazie agli strumenti di business intelligence (BI) esistenti. Con Amazon Redshift, è possibile ottenere le prestazioni di motori di data warehousing colonnari che eseguono l'elaborazione MPP (Massively Parallel Processing), a un decimo del costo. Si può partire in piccolo con 0,25 dollari all'ora senza impegni e arrivare fino a petabyte di dati al costo di 1.000 dollari per terabyte all'anno.

Dal suo lancio nel febbraio 2013, Amazon Redshift è uno dei servizi AWS a più rapida crescita, con molte migliaia di clienti in diversi settori e aziende di varie dimensioni. Imprese come NTT DOCOMO, FINRA, Johnson & Johnson, Hearst, Amgen e NASDAQ hanno effettuato la migrazione ad Amazon Redshift. Amazon Redshift è stato quindi classificato tra i leader nel rapporto *Forrester Wave: Enterprise Data Warehouse, Q4 2015*.[1]

Questo whitepaper fornisce le informazioni necessarie per sfruttare la transizione strategica che si sta verificando nel data warehousing, con il passaggio da locale a cloud:

- Architettura di analisi moderna
- Scelte tecnologiche per il data warehousing disponibili all'interno di tale architettura
- Approfondimento di Amazon Redshift e delle caratteristiche che lo differenziano
- Un piano per la realizzazione di un sistema completo di data warehousing basato su AWS con Amazon Redshift e altri servizi
- Suggerimenti pratici per la migrazione da altre soluzioni di data warehousing e approfondimento dell'ecosistema dei nostri partner

Architettura di analisi e data warehousing moderna

Abbiamo già detto che un *data warehouse* è un repository centrale di informazioni provenienti da una o più origini dati. Solitamente i dati affluiscono al data warehouse da sistemi transazionali e da altri database relazionali e includono, di norma, dati strutturati, semi strutturati e non strutturati. Questi dati vengono elaborati, trasformati e inseriti a intervalli regolari. Gli utenti, tra cui data scientist, analisti aziendali e responsabili decisionali, accedono ai dati attraverso strumenti di BI, client SQL e fogli di calcolo.

Perché realizzare un data warehouse, perché non eseguire semplicemente le query di analisi direttamente su un database OLTP (Online Transaction Processing), dove vengono registrate le transazioni? Per rispondere a questa domanda, esaminiamo le differenze tra data warehouse e database OLTP. I data warehouse sono ottimizzati per operazioni di scrittura in batch e per la lettura di volumi elevati di dati, mentre i database OLTP sono ottimizzati per operazioni continue di scrittura e volumi elevati di piccole operazioni di lettura. In generale, i data warehouse utilizzano schemi denormalizzati come lo schema a stella e lo schema a fiocco di neve in ragione dei requisiti elevati di throuput di dati, mentre i database OLTP impiegano schemi altamente normalizzati, più adatti ai requisiti elevati di throuput di transazioni. Lo schema a stella è costituito da poche grandi tabelle fattuali che fanno riferimento a una serie di tabelle dimensionali. Lo schema a fiocco di neve (un'estensione dello schema a stella) è costituito da tabelle dimensionali normalizzate in modo ancora più marcato.

Per sfruttare i vantaggi dell'utilizzo di un data warehouse gestito come datastore separato con il proprio database OLTP di origine o un altro sistema di origine, consigliamo di realizzare una pipeline di dati efficiente. Tale pipeline estrae i dati dal sistema di origine, li converte in uno schema idoneo al data warehousing e infine li carica nel data warehouse. Nella sezione successiva esamineremo gli elementi fondamentali di una pipeline di analisi e i diversi servizi AWS utilizzabili per definire l'architettura della pipeline.

Architettura di analisi

Le pipeline di analisi sono progettate per gestire grandi volumi di flussi di dati in ingresso provenienti da sorgenti eterogenee come database, applicazioni e dispositivi.

Una pipeline di analisi è caratterizzata, in genere, dalle fasi seguenti:

1. Raccolta dei dati.
2. Storage dei dati.
3. Elaborazione dei dati.
4. Analisi e visualizzazione dei dati.

La Figura 1, di seguito, fornisce un'illustrazione.

Figura 1: pipeline di analisi

Raccolta dei dati

Nella fase di raccolta dei dati, vi saranno probabilmente tipologie di dati diverse, come i dati transazionali, i dati di log, i dati di streaming e i dati IoT (Internet of Things). AWS offre soluzioni per lo storage dei dati appartenenti a ciascuna di queste tipologie.

Dati transazionali

I dati transazionali, come quelli relativi alle transazioni di acquisto dell'E-commerce e alle transazioni finanziarie, sono archiviati, di norma, in sistemi di gestione di database relazionali (RDBMS) o in sistemi di database NoSQL. La scelta della soluzione di database dipende dalle caratteristiche dei casi d'uso e delle applicazioni. Un database NoSQL è adatto quando i dati non sono correttamente strutturati per adattarsi a uno schema definito o quando lo schema cambia molto spesso. Una soluzione RDBMS, d'altro canto, è adatta quando le transazioni avvengono in più righe di tabella e le query richiedono join complessi. Amazon DynamoDB è un servizio di database NoSQL completamente gestito che può essere utilizzato come store OLTP per le proprie applicazioni. Amazon RDS permette di implementare una soluzione di database relazionale basata su SQL per la propria applicazione.

Dati di log

L'acquisizione affidabile di log generati dal sistema aiuta a risolvere i problemi, eseguire controlli ed effettuare analisi utilizzando le informazioni archiviate nei log. Amazon Simple Storage Service (Amazon S3) è una popolare soluzione di storage per dati non transazionali, come i dati di log, utilizzata a fini di analisi. Dato che offre 11 9 di durabilità (ossia una durabilità pari al 99,999999999%), Amazon S3 è diffusa anche come soluzione di archiviazione.

Dati di streaming

Le applicazioni Web, i dispositivi mobili e molte applicazioni e servizi software possono generare enormi quantità di [dati di streaming], a volte nell'ordine di terabyte all'ora, che devono essere raccolti, archiviati ed elaborati in modo continuo[2]. Con i servizi Amazon Kinesis, è possibile farlo in modo semplice e conveniente.

Dati IoT

In tutto il mondo, dispositivi e sensori inviano continuamente messaggi. Le imprese avvertono sempre più l'esigenza di acquisire tali dati e ricavarne informazioni utili. Con AWS IoT, i dispositivi connessi interagiscono in modo semplice e sicuro con il cloud AWS. AWS IoT consente di usare facilmente servizi AWS come AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Machine Learning e Amazon DynamoDB per creare applicazioni in grado di raccogliere, elaborare, analizzare e agire sulla base dei dati IoT, senza che occorra gestire un'infrastruttura.

Elaborazione dei dati

Il processo di raccolta fornisce dati che potenzialmente contengono informazioni utili. Si ha la possibilità di analizzare le informazioni estratte e utilizzarle per far crescere la propria attività. Tali informazioni potrebbero, ad esempio, aiutare a capire il comportamento degli utenti e la popolarità relativa dei propri prodotti. La best practice per raccogliere tali informazioni è caricare i dati non elaborati in un data warehouse per eseguire l'ulteriore analisi.

A tale scopo, esistono due tipi di flussi di lavoro di elaborazione ossia batch e in tempo reale. Le forme più diffuse di elaborazione, OLAP (Online Analytic Processing) e OLTP, utilizzano ciascuna uno di questi tipi. L'elaborazione OLAP si basa, in genere, su batch. I sistemi OLTP, invece, sono orientati all'elaborazione in tempo reale e non sono adatti, di norma, all'elaborazione basata su batch. Se si dissocia l'elaborazione dei dati dal proprio sistema OLTP, si evita che l'elaborazione dei dati abbia ripercussioni sul carico di lavoro OLTP.

In primo luogo esaminiamo gli elementi costitutivi dell'elaborazione batch.

ETL (Extract Transform Load)

Il processo ETL consiste nell'estrazione dei dati da più origini per caricarli nei sistemi di data warehousing. Il processo ETL è, di norma, un processo continuo, con un flusso di lavoro ben definito. Durante tale processo, i dati vengono inizialmente estratti da una o più origini. I dati estratti sono successivamente puliti, arricchiti, trasformati e caricati in un data warehouse. Strumenti del framework Hadoop come Apache Pig e Apache Hive sono normalmente utilizzati in una pipeline ETL per effettuare trasformazioni di grandi volumi di dati.

ELT (Extract Load Transform)

Il processo ELT è una variante dell'ETL in cui i dati estratti sono caricati innanzi tutto nel sistema di destinazione. Le trasformazioni sono effettuate dopo che i dati sono stati caricati nel data warehouse. Il processo ELT funziona correttamente, in genere, quando il sistema di destinazione è abbastanza potente da gestire le trasformazioni. Amazon Redshift è spesso utilizzato nelle pipeline ELT perché molto efficiente nell'esecuzione di trasformazioni.

OLAP (Online Analytical Processing)

I sistemi OLAP effettuano lo storage di dati storici aggregati in schemi multidimensionali. Utilizzati diffusamente nel data mining, i sistemi OLAP permettono di estrarre dati e individuare tendenze su più dimensioni. Visto che è ottimizzato per join rapidi, Amazon Redshift viene spesso utilizzato per realizzare sistemi OLAP.

Esaminiamo ora gli elementi costitutivi dell'elaborazione in tempo reale dei dati.

Elaborazione in tempo reale

In precedenza abbiamo parlato dei dati di streaming e abbiamo menzionato Amazon Kinesis come soluzione per acquisire e archiviare tali dati. Si possono elaborare tali dati in modo sequenziale e incrementale, record per record oppure su finestre temporali mobili e utilizzare i dati elaborati per una vasta gamma di analisi tra cui correlazioni, aggregazioni, applicazione di filtri e campionamento. Questo tipo di elaborazione è denominato elaborazione in tempo reale. Le informazioni ottenute con l'elaborazione in tempo reale forniscono alle aziende visibilità su molti aspetti della loro attività e dell'attività dei clienti, come l'utilizzo dei servizi (per la lettura o la fatturazione), l'attività dei server, i clic nei siti Web e la geolocalizzazione di dispositivi, persone e beni fisici, oltre a consentire loro di reagire tempestivamente alle situazioni emergenti. L'elaborazione in tempo reale richiede un layer di elaborazione simultaneo e altamente scalabile.

Per elaborare i dati di streaming in tempo reale, è possibile utilizzare AWS Lambda. Lambda è in grado di elaborare i dati direttamente da AWS IoT o Amazon Kinesis Streams. Lambda consente di eseguire il codice senza effettuare il provisioning dei server o senza gestirli.

Amazon Kinesis Client Library (KCL) è un altra soluzione per elaborare i dati provenienti da Amazon Kinesis Streams. KCL offre maggiore flessibilità rispetto ad AWS Lambda nella divisione in batch dei dati in ingresso per l'ulteriore elaborazione. Si può anche utilizzare KCL per applicare ampie trasformazioni e personalizzazioni alla logica di elaborazione.

Amazon Kinesis Firehose è il modo più semplice per caricare i dati di streaming in AWS. È in grado di acquisire i dati di streaming e di caricarli automaticamente in Amazon Redshift, consentendo un'analisi quasi in tempo reale con gli strumenti di BI e i pannelli di controllo che già usi. È possibile definire le proprie regole di batching con Firehose, che si occupa in modo affidabile della ripartizione in batch dei dati e della loro trasmissione ad Amazon Redshift.

Storage dei dati

Si può effettuare lo storage dei dati in un data warehouse o in un data mart, come descritto nel prosieguo.

Data warehouse

Come già detto, un *data warehouse* è un repository centrale di informazioni provenienti da una o più origini dati. Con i data warehouse, si possono eseguire analisi rapide su grandi volumi di dati e far emergere i modelli nascosti nei dati utilizzando gli strumenti di business intelligence. I data scientist effettuano query di data warehouse per eseguire analisi offline e identificare tendenze. Gli utenti dell'organizzazione consumano i dati utilizzando query SQL ad hoc, rapporti periodici e pannelli di controllo, al fine di prendere decisioni business-critical.

Data mart

Un *data mart* è una forma semplice di data warehouse incentrata su una specifica area funzionale o materia. Ad esempio, si possono avere data mart specifici per ogni divisione dell'organizzazione o data mart di segmenti basati sulle regioni. Si possono realizzare data mart da un data warehouse di grandi dimensioni o da store operativi oppure optare per una soluzione ibrida tra le due. I data mart sono semplici da progettare, realizzare e amministrare. Tuttavia, poiché i data mart sono incentrati su specifiche aree funzionali, l'esecuzione di query tra aree funzionali può risultare complessa a causa della distribuzione.

Si può utilizzare Amazon Redshift per creare data mart che vadano ad aggiungersi ai data warehouse.

Analisi e visualizzazione

Dopo avere elaborato i dati e averli resi disponibili per l'ulteriore analisi, è necessario disporre degli strumenti adatti per analizzare e visualizzare i dati elaborati.

In molti casi, si può eseguire l'analisi dei dati con gli stessi strumenti utilizzati per l'elaborazione. È possibile servirsi di strumenti come SQL Workbench per analizzare i dati in Amazon Redshift con ANSI SQL. Amazon Redshift funziona correttamente anche con soluzioni di BI diffuse di terze parti disponibili sul mercato.

Amazon QuickSight è un servizio di business intelligence rapido e nativo sul cloud che consente di creare visualizzazioni in tutta facilità, effettuare analisi ad hoc e ottenere rapidamente informazioni aziendali dai dati. Amazon QuickSight è integrato con Amazon Redshift ed è attualmente in anteprima. La sua disponibilità generale è prevista più avanti nel 2016.

Se si utilizza Amazon S3 come storage principale, un modo diffuso per effettuare l'analisi e la visualizzazione consiste nell'eseguire notebook di Apache Spark su Amazon Elastic MapReduce (Amazon EMR). In questo modo, si dispone della flessibilità di eseguire SQL o un codice personalizzato scritto in linguaggi come Python e Scala.

Se si vuole adottare un altro approccio per la visualizzazione, Apache Zeppelin è una soluzione open source di business intelligence eseguibile su Amazon EMR per visualizzare i dati in Amazon S3 utilizzando Spark SQL. È anche possibile utilizzare Apache Zeppelin per visualizzare i dati in Amazon Redshift.

Pipeline di analisi con i servizi AWS

AWS offre una vasta gamma di servizi per l'implementazione di una piattaforma di analisi end-to-end. La Figura 2 mostra i servizi descritti in precedenza e la loro posizione all'interno della pipeline di analisi.

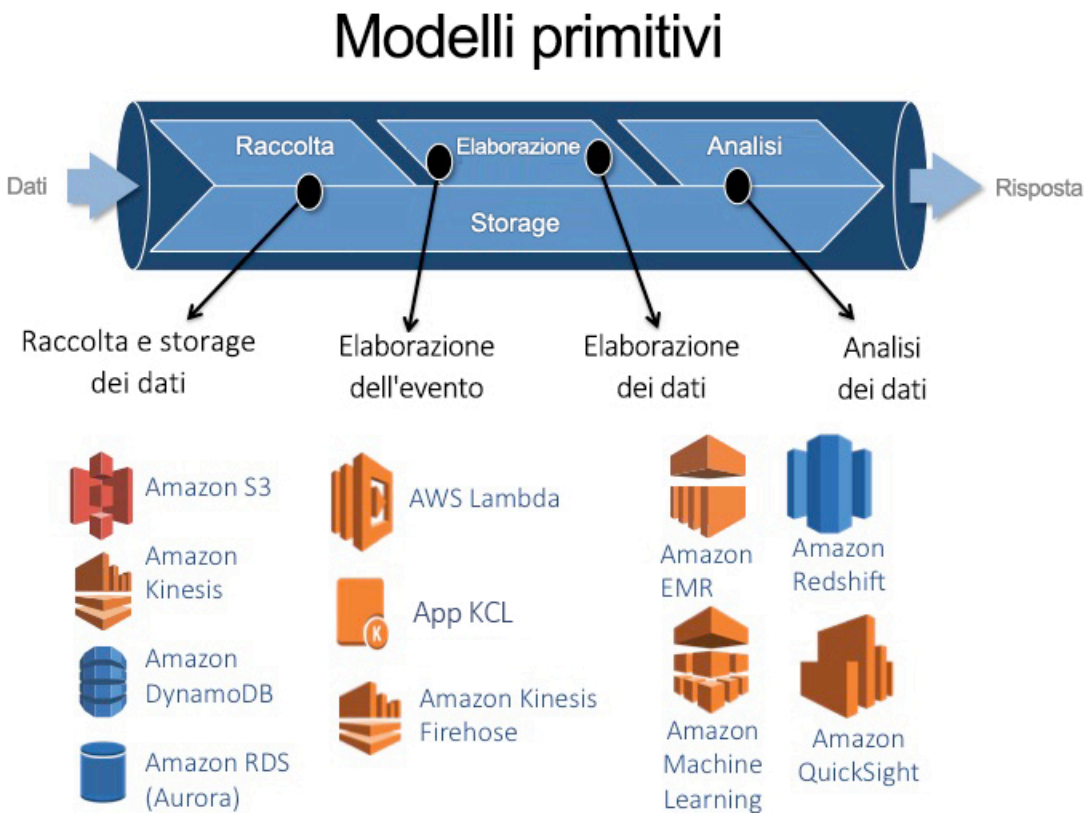


Figura 2: pipeline di analisi con i servizi AWS

Opzioni disponibili nella tecnologia di data warehouse

In questa sezione, esamineremo le opzioni disponibili per la realizzazione di un data warehouse: database orientati alle righe, database orientati alle colonne e architetture MPP (Massively Parallel Processing).

Database orientati alle righe

Solitamente i database orientati alle righe effettuano lo storage di righe intere in un blocco fisico. Si ottengono prestazioni elevate per le operazioni di lettura con indici secondari. Database come Oracle Database Server, Microsoft SQL Server, MySQL e PostgreSQL sono sistemi orientati alle righe. Tali sistemi sono stati utilizzati tradizionalmente per il data warehousing, ma sono più adatti all'elaborazione transazionale (OLTP) che all'analisi.

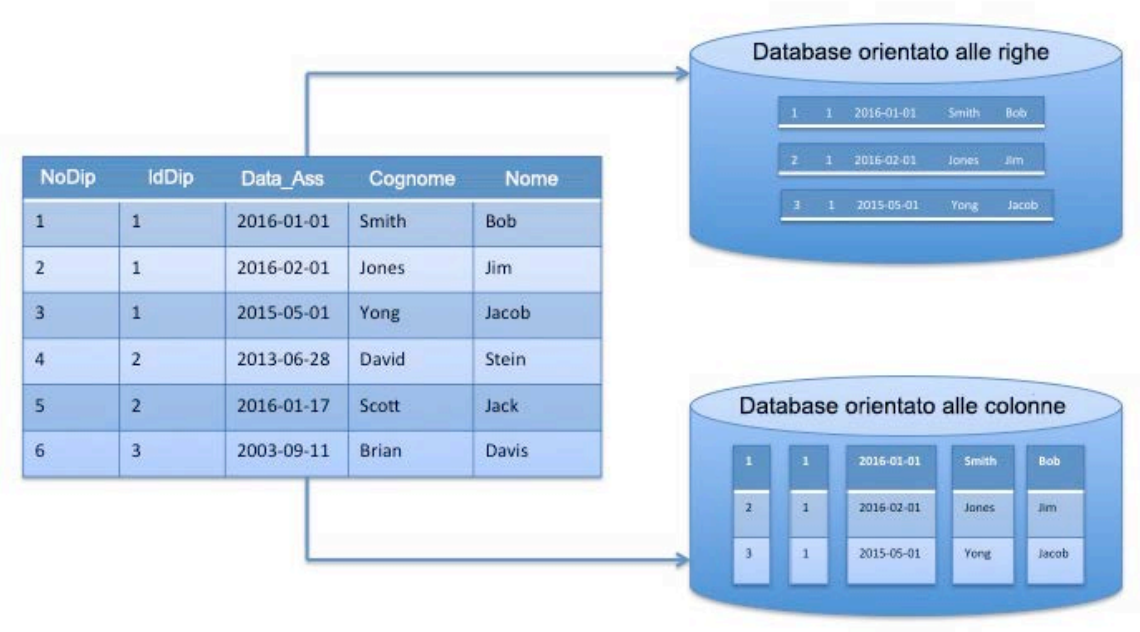
Per ottimizzare le prestazioni di un sistema basato su righe utilizzato come data warehouse, gli sviluppatori utilizzano numerose tecniche, come la realizzazione di viste materializzate, la creazione di tabelle di rollup preaggregate, la realizzazione di indici relativi a ogni possibile combinazione di predicati, l'implementazione di partizioni di dati per sfruttare lo sfoltimento delle partizioni da parte dell'ottimizzatore di query e l'esecuzione di join basati su indici.

I data store tradizionali basati su righe sono limitati dalle risorse disponibili sulla singola macchina. I data mart riducono in parte il problema utilizzando il partizionamento orizzontale funzionale. Si può dividere il data warehouse in più data mart, capaci di soddisfare ciascuno una specifica area funzionale. Tuttavia, quando le dimensioni dei data mart, con il tempo, aumentano, l'elaborazione dei dati rallenta.

In un data warehouse basato su righe, ogni query deve essere letta in tutte le colonne per tutte le righe dei blocchi che soddisfano il predicato della query, comprese le colonne non scelte. Tale approccio crea un collo di bottiglia importante nelle prestazioni dei data warehouse, dove le tabelle hanno più colonne, ma le query ne utilizzano solo alcune.

Database orientati alle colonne

I database orientati alle colonne organizzano ogni colonna in un proprio insieme di blocchi fisici invece di comprimere le righe intere in un blocco. Questa funzionalità permette ai database di essere più efficienti in termini di I/O per le query di sola lettura, dato che devono solo leggere le colonne a cui una query accede da disco (o da memoria). Tale approccio fa dei database orientati alle colonne una scelta migliore per il data warehousing rispetto ai database orientati alle righe.



NoDip	IdDip	Data_Ass	Cognome	Nome
1	1	2016-01-01	Smith	Bob
2	1	2016-02-01	Jones	Jim
3	1	2015-05-01	Yong	Jacob
4	2	2013-06-28	David	Stein
5	2	2016-01-17	Scott	Jack
6	3	2003-09-11	Brian	Davis

Figura 3: confronto tra database orientati alle righe e database orientati alle colonne

La precedente Figura 3 illustra le differenze principali tra database orientati alle righe e database orientati alle colonne. Nel database orientato alle righe, le righe sono compresse nei propri blocchi, mentre le colonne sono compresse nei propri blocchi in un database orientato alle colonne.

Oltre all'I/O più veloce, l'altro importante vantaggio dell'utilizzo di un database orientato alle colonne è il miglioramento della compressione. Dato che ogni colonna è compressa nel proprio insieme di blocchi, ogni blocco fisico contiene lo stesso tipo di dati. Quando tutti i dati appartengono allo stesso tipo, il database è in grado di utilizzare algoritmi di compressione altamente efficienti. Pertanto, occorre uno storage inferiore rispetto al database orientato alle righe. Questo approccio, inoltre, comporta un I/O notevolmente inferiore, dato che gli stessi dati sono archiviati in un minor numero di blocchi.

Alcuni dei database orientati alle colonne utilizzati per il data warehousing sono Amazon Redshift, Vertica, Teradata Aster e Druid.

Architetture MPP (Massively Parallel Processing)

Un'architettura MPP permette di utilizzare tutte le risorse disponibili nel cluster per elaborare i dati, con un netto miglioramento delle performance dei data warehouse nell'ordine di petabyte. I data warehouse MPP permettono di migliorare le prestazioni con la semplice aggiunta di più nodi al cluster. Amazon Redshift, Druid, Vertica, GreenPlum e Teradata Aster sono alcuni dei data warehouse realizzati su un'architettura MPP. L'architettura MPP è supportata anche da framework open source come Hadoop e Spark.

Approfondimento su Amazon Redshift

In quanto tecnologia MPP colonnare, Amazon Redshift offre alcuni vantaggi fondamentali per un data warehousing performante e conveniente tra cui la compressione ottimizzata, un I/O inferiore e la riduzione dei requisiti di storage. Dato che si basa su ANSI SQL, si possono eseguire le query esistenti con poche o nessuna modifica. È quindi diventata una scelta popolare per i data warehouse e i data mart aziendali di oggi. In questa sezione esamineremo Amazon Redshift e le sue funzionalità in modo più approfondito.

Amazon Redshift consente di ottenere query e performance di I/O più rapide pressoché per ogni dimensione di dati, attraverso lo storage colonnare e la parallelizzazione e distribuzione delle query tra più nodi. Automatizza la maggior parte delle attività amministrative comuni associate al provisioning, alla configurazione, al monitoraggio, al backup e alla messa in sicurezza di un data warehouse, con semplificazione della gestione e riduzione dei relativi costi. Grazie all'automazione, si possono realizzare data warehouse nell'ordine di petabyte in pochi minuti, invece delle settimane o dei mesi necessari per le implementazioni locali tradizionali.

Prestazioni

Amazon Redshift utilizza lo storage colonnare, la compressione dei dati e le mappe delle zone per ridurre la quantità di I/O necessaria all'esecuzione delle query. L'ordinamento "interleaved" consente performance rapide senza il sovraccarico dovuto al mantenimento di indici o proiezioni.

Amazon Redshift impiega un'architettura MPP per sfruttare tutte le risorse disponibili attraverso la parallelizzazione e la distribuzione di operazioni SQL. L'hardware sottostante è stato progettato per ottenere prestazioni elevate nell'elaborazione dati, utilizzando lo storage locale collegato per massimizzare il throughput tra CPU e unità, oltre a una rete mesh da 10 GigE per massimizzare il throughput tra nodi. Le performance possono essere ottimizzate sulla base delle proprie esigenze di data warehousing: AWS offre Dense Compute (DC) con unità SSD (Solid State Drive) e opzioni Dense Storage (DS). La distribuzione continua di aggiornamenti software consente di ottenere miglioramenti costanti delle prestazioni senza alcun intervento da parte degli utenti.

Durabilità e disponibilità

Per garantire la massima durabilità e disponibilità dei dati, Amazon Redshift rileva e sostituisce automaticamente gli eventuali nodi non riusciti presenti nel cluster di data warehouse. Rende immediatamente disponibile il nodo sostitutivo e carica per primi i dati a cui accedi più frequentemente, per permetterti di riprendere rapidamente l'esecuzione delle query di dati. Poiché Amazon Redshift replica i dati in tutto il cluster, utilizza i dati di un altro nodo per ricostruire il nodo non riuscito. Il cluster è in modalità di sola lettura fino al provisioning del nodo sostitutivo e alla sua aggiunta al cluster, che in genere richiede solo pochi minuti.

I cluster di Amazon Redshift risiedono in una zona di disponibilità[3]. Se, tuttavia, si preferisce una configurazione Multi-AZ per Amazon Redshift, si può creare un mirror e quindi autogestire la replica e il failover.

Bastano pochi clic nella console di gestione di Amazon Redshift per impostare un ambiente di disaster recovery (DR) efficace con Amazon Redshift. Si possono conservare le copie dei backup in più Regioni AWS. In caso di interruzione del servizio in una Regione AWS, si può ripristinare il cluster da backup di una Regione AWS diversa. È possibile avere accesso in lettura/in scrittura al proprio cluster entro pochi minuti dall'avvio dell'operazione di ripristino.

Scalabilità ed elasticità

Bastano pochi clic nella console o una chiamata API per modificare facilmente il numero e il tipo di nodi del data warehouse quando le esigenze in termini di prestazioni o capacità cambiano[4]. Amazon Redshift consente di iniziare solo con un unico nodo da 160 GB e salire fino a un petabyte o più di dati utente compressi utilizzando molti nodi. Per ulteriori informazioni, consultare About Clusters and Nodes (Informazioni su cluster e nodi) in *Amazon Redshift Cluster Management Guide* (Guida alla gestione dei cluster in Amazon Redshift)[5].

Durante l'adattamento delle dimensioni, Amazon Redshift mette il nodo esistente in modalità di sola lettura, effettua il provisioning di un nuovo cluster delle dimensioni scelte e quindi copia in parallelo i dati dal vecchio cluster a quello nuovo. Nel corso di tale processo, si paga soltanto il cluster Amazon Redshift attivo. Si può continuare a eseguire query nel vecchio cluster mentre viene effettuato il provisioning di quello nuovo. Una volta copiati i dati nel nuovo cluster, Amazon Redshift reindirizza automaticamente le query al nuovo cluster ed elimina il vecchio cluster.

Si possono utilizzare le chiamate API di Amazon Redshift per lanciare sistematicamente cluster, scalare cluster, creare backup, ripristinare i backup e molto altro. Grazie a questo approccio, si possono integrare tali chiamate API nello stack di automazione esistente o realizzare un'automazione personalizzata adatta alle proprie esigenze.

Interfacce

Amazon Redshift dispone di driver personalizzati Java Database Connectivity (JDBC) e Open Database Connectivity (ODBC) che si possono scaricare dalla scheda **Connect Client** della console, per utilizzare una vasta gamma di client SQL familiari. È anche possibile utilizzare driver PostgreSQL JDBC e ODBC standard. Per maggiori informazioni sui driver Amazon Redshift, consulta Amazon Redshift and PostgreSQL in *Amazon Redshift Database Developer Guide (Guida per gli sviluppatori ai database Amazon Redshift)*[6].

Si possono anche trovare numerosi esempi di integrazioni convalidate con molti fornitori BI ed ETL conosciuti[7]. In queste integrazioni, il carico e lo scarico vengono eseguiti in parallelo su ogni nodo di calcolo per massimizzare la velocità di assunzione o esportazione dei dati da e verso più risorse, tra cui Amazon S3, Amazon EMR e Amazon DynamoDB. Si possono caricare facilmente i dati di streaming in Amazon Redshift con Amazon Kinesis Firehose, consentendo un'analisi quasi in tempo reale con gli strumenti di BI e i pannelli di controllo già in uso. Si possono individuare i parametri per l'utilizzo della capacità di calcolo, l'utilizzo della memoria, l'utilizzo dello storage e il traffico in lettura/scrittura relativi al cluster di data warehouse Amazon Redshift utilizzando la console o le operazioni API di Amazon CloudWatch.

Sicurezza

Per rafforzare la sicurezza dei dati, è possibile eseguire Amazon Redshift all'interno di un Virtual Private Cloud basato sul servizio Amazon Virtual Private Cloud (Amazon VPC). Si può utilizzare il modello di networking definito da software di VPC per impostare le regole di firewall che limitano il traffico sulla base delle regole configurate[8]. Amazon Redshift supporta le connessioni abilitate SSL tra l'applicazione client e il cluster di data warehouse Amazon Redshift, consentendo in questo modo la crittografia dei dati in transito.

I nodi di calcolo di Amazon Redshift archiviano i dati, ma è possibile accedere ai dati solo dal nodo principale del cluster. Tale isolamento garantisce un ulteriore layer di sicurezza. Amazon Redshift si integra con AWS CloudTrail per consentire di controllare tutte le chiamate API Amazon Redshift[9]. Per aiutare a garantire la sicurezza dei dati memorizzati, Amazon Redshift effettua la crittografia di ogni blocco con la crittografia AES-256 accelerata via hardware mentre ogni blocco viene scritto su disco. La crittografia avviene a un basso livello del sottosistema I/O, il quale crittografa tutto quello che viene scritto su disco, compresi i risultati intermedi delle query. Il backup dei blocchi viene effettuato "così come sono", il che significa che anche i backup sono crittografati. Per impostazione predefinita, Amazon Redshift si occupa della gestione delle chiavi, ma è possibile scegliere di gestire le chiavi utilizzando i propri moduli di sicurezza hardware (HSM) o di gestire le chiavi tramite AWS Key Management Service[10,11].

Modello di costo

Per Amazon Redshift non occorrono impegni a lungo termine o costi anticipati. Questo approccio tariffario libera dal peso delle spese in conto capitale e dalla complessità di pianificare e acquistare capacità di data warehouse in anticipo per soddisfare le esigenze future. Le spese si basano sulle dimensioni e sul numero di nodi del cluster.

Non sono previsti addebiti aggiuntivi per uno storage di backup fino al 100% dello spazio di storage di cui è stato effettuato il provisioning. Ad esempio, se si dispone di un cluster attivo con due nodi XL per un totale di 4 TB di storage, AWS fornisce fino a 4 TB di storage di backup su Amazon S3 senza costi aggiuntivi. Lo storage di backup che supera le dimensioni dello spazio di storage di cui è stato effettuato il provisioning e i backup archiviati dopo l'eliminazione del cluster sono fatturati secondo le tariffe Amazon S3 standard[12]. Non sono previsti costi di trasferimento dei dati per le comunicazioni tra Amazon S3 e Amazon Redshift. Per ulteriori informazioni, consultare Amazon Redshift Pricing (Prezzi Amazon Redshift)[13].

Modelli di utilizzo ideale

Amazon Redshift è la soluzione ideale per l'elaborazione analitica online (OLAP) con l'ausilio degli strumenti di business intelligence di cui già si dispone. Le organizzazioni utilizzano Amazon Redshift per:

- Eseguire la BI e il reporting aziendali
- Analizzare i dati di vendita globali per più prodotti
- Effettuare lo storage dei dati storici relativi alle negoziazioni
- Analizzare impression pubblicitarie e clic
- Aggregare i dati relativi ai giochi
- Analizzare le tendenze social
- Misurare la qualità clinica, l'efficienza operativa e le performance finanziarie nell'assistenza sanitaria

Modelli non idonei

Amazon Redshift non rappresenta la soluzione ideale per i seguenti modelli di utilizzo:

- **Piccoli set di dati** – Amazon Redshift è stato realizzato per l'elaborazione parallela in un cluster. Se il set di dati è inferiore a 100 gigabyte, non è possibile sfruttare appieno i vantaggi che Amazon Redshift è in grado di offrire. Pertanto Amazon RDS potrebbe rappresentare una soluzione migliore.
- **OLTP** – Amazon Redshift è progettato per carichi di lavoro di data warehousing grazie a capacità analitiche estremamente rapide e convenienti. Se occorre un sistema transazionale rapido, potrebbe essere meglio optare per un sistema tradizionale di database relazionali basato su Amazon RDS o un database NoSQL come Amazon DynamoDB.

- **Dati non strutturati** – I dati di Amazon Redshift devono essere strutturati da uno schema definito. Amazon Redshift non supporta una struttura con schema arbitrario per ogni riga. Se i dati non sono strutturati, è possibile eseguire le operazioni ETL (Extract, Transform and Load, estrazione, trasformazione e caricamento) su Amazon EMR per ottenere i dati pronti per essere caricati in Amazon Redshift. Per i dati JSON, è possibile archiviare le coppie chiave-valore e utilizzare le funzioni JSON native nelle query[14].
- **Dati BLOB** – Se si intende effettuare lo storage di file BLOB (Binary Large Object, oggetto binario di grandi dimensioni) come video digitali, immagini o musica, si potrebbe valutare la possibilità di effettuare lo storage dei dati in Amazon S3 e referenziarne l'ubicazione in Amazon Redshift. In questo scenario, Amazon Redshift tiene traccia dei metadati (come l'item name, le dimensioni, la data creazione, l'ubicazione ecc.) relativi agli oggetti binari, ma gli oggetti di grandi dimensioni sono archiviati in Amazon S3.

Migrazione ad Amazon Redshift

Se si decide di migrare da un data warehouse esistente ad Amazon Redshift, la scelta della strategia di migrazione più adatta dipende da vari fattori:

- Le dimensioni del database e delle sue tabelle
- La larghezza di banda di rete tra il server di origine e AWS
- La migrazione e il passaggio ad AWS sono effettuati in un'unica fase o in una sequenza di fasi nel tempo
- La frequenza di modifica dei dati nel sistema di origine
- Le trasformazioni durante la migrazione
- Lo strumento partner che intendi utilizzare per la migrazione e l'ETL

Migrazione in un'unica fase

La migrazione in un'unica fase è una buona scelta per i database di piccole dimensioni che non devono funzionare in continuo. I clienti possono estrarre i database esistenti come file CSV (Comma-Separated Value, valori delimitati da virgole) quando utilizzano servizi come AWS Import/Export Snowball per trasmettere set di dati ad Amazon S3 per il caricamento in Amazon Redshift. Successivamente i clienti eseguono test sul database Amazon Redshift di destinazione per verificare la coerenza dei dati con l'origine. Una volta superare tutte le convalide, il database passa ad AWS.

Migrazione in due fasi

La migrazione in due fasi viene utilizzata normalmente per database di qualunque dimensione:

1. **Migrazione iniziale dei dati:** I dati sono estratti dal database di origine, preferibilmente non durante i picchi di utilizzo al fine di ridurre al minimo l'impatto. I dati sono quindi migrati ad Amazon Redshift seguendo l'approccio di migrazione in una sola fase descritto in precedenza.
2. **Migrazione dei dati modificati:** I dati che sono stati modificati nel database di origine dopo la migrazione iniziale dei dati sono propagati nella destinazione prima dello switchover. In questa fase i database di origine e di destinazione vengono sincronizzati. Una volta completata la migrazione di tutti i dati modificati, si possono convalidare i dati nel database di destinazione, eseguire i test necessari e, se tutti i test sono stati superati, effettuare il passaggio al data warehouse Amazon Redshift.

Strumenti per la migrazione dei database

Esistono vari strumenti e tecnologie disponibili per la migrazione dei dati. Alcuni di questi strumenti sono intercambiabili, oppure si possono anche utilizzare strumenti di terze parti o open source disponibili sul mercato.

1. AWS Database Migration Service supporta sia il processo di migrazione in una fase sia quello in due fasi descritti in precedenza[15]. Per seguire il processo di migrazione in due fasi, è necessario attivare la registrazione supplementare per acquisire le modifiche al sistema di origine. È possibile attivare la registrazione supplementare a livello di tabella o di database.
2. Gli strumenti aggiuntivi dei partner per l'integrazione dei dati sono i seguenti:
    - Attunity
    - Informatica
    - SnapLogic
    - Talend
    - Bryte

Per ulteriori informazioni sull'integrazione dei dati e i partner con ruolo di consulenti, consultare Amazon Redshift Partners[16].

Progettazione dei flussi di lavoro di data warehousing

Nelle sezioni precedenti abbiamo esaminato le caratteristiche di Amazon Redshift che lo rendono la soluzione ideale per il data warehousing. Per capire come progettare flussi di lavoro di data warehousing con Amazon Redshift, prendiamo ora in esame il modello di progettazione più diffuso insieme a un caso d'uso esemplificativo.

Supponiamo che un produttore multinazionale di capi d'abbigliamento che dispone di più di mille negozi al dettaglio venda determinate linee di abbigliamento attraverso grandi magazzini e discount e che abbia una presenza online. Da un punto di vista tecnico, questi tre canali attualmente operano in maniera indipendente, hanno un management diverso, nonché sistemi POS e uffici contabili differenti. Nessuno di questi sistema unisce tutti i set di dati correlati per fornire al CEO una visione a 360 gradi dell'attività nel suo insieme.

Supponiamo inoltre che il CEO voglia disporre di un quadro d'insieme a livello di società di questi canali ed essere in grado di eseguire analisi ad hoc tra cui:

- Quali tendenze emergono fra canali?
- Quali regioni geografiche ottengono risultati migliori fra canali?
- Quanto sono efficaci le pubblicità e le promozioni della società?
- Quali tendenze esistono fra linee di abbigliamento?
- Quali forze esterne hanno effetti sulle vendite della società, ad esempio il tasso di disoccupazione e le condizioni meteorologiche?
- In che modo le caratteristiche dei negozi influenzano le vendite, ad esempio il contratto dei dipendenti e dei direttori, la posizione del negozio (in un centro commerciale o su una via dello shopping), la posizione dei prodotti nel negozio, le promozioni, gli espositori, i volantini e le vetrine all'interno dei negozi?

Un data warehouse aziendale risolve questo problema. Raccoglie i dati dai vari sistemi di ciascuno dei tre canali di vendita oltre che dati pubblici come i bollettini meteorologici e i rapporti economici. Ogni origine dati invia tutti i giorni i dati per l'utilizzo da parte del data warehouse. Poiché ogni origine dati potrebbe essere strutturata diversamente, viene eseguito un processo ETL per riformattare i dati in una struttura comune. Successivamente è possibile eseguire simultaneamente l'analisi dei dati da tutte le origini. A tale scopo, utilizziamo la seguente architettura di flussi di dati:

Figura 4: flusso di lavoro per il data warehouse aziendale.

1. La prima fase di questo processo consiste nel portare dati con origini diverse in Amazon S3. Amazon S3 offre una piattaforma di storage estremamente durevole, economica e scalabile in cui è possibile scrivere in parallelo da origini diverse a un costo particolarmente basso.
2. Amazon EMR è utilizzato per trasformare e pulire i dati dal formato di origine al formato di destinazione. Amazon EMR è dotato dell'integrazione con Amazon S3, che permette thread paralleli di throughput da ogni nodo del cluster Amazon EMR da e verso Amazon S3.

   In genere, un data warehouse riceve nuovi dati ogni notte. Dato che di notte non occorre eseguire analisi, l'unico requisito di questo processo di trasformazione è che si concluda prima del mattino, quando il CEO e gli altri utenti aziendali devono poter accedere ai rapporti e al pannello di controllo. Si può quindi utilizzare Amazon EC2 Spot Market per ridurre ulteriormente il costo del processo ETL qui[17]. Una buona strategia spot consiste nell'iniziare a fare offerte a un prezzo molto basso a mezzanotte e continuare ad aumentare il prezzo con il passare del tempo finché si ottiene la capacità. Quando si avvicina la scadenza, se le offerte spot non hanno avuto successo, è possibile ritornare ai prezzi on demand per assicurarsi di continuare a soddisfare i requisiti relativi al tempo di completamento. Ogni origine potrebbe essere soggetta a un processo di trasformazione diverso in Amazon EMR, ma con il modello AWS "pay-as-you-go", si può creare un cluster Amazon EMR separato per ogni trasformazione e ottimizzarlo affinché abbia esattamente la capacità necessaria a completare tutte le attività di trasformazione dati senza competere con le risorse delle altre attività.

3. Ogni attività di trasformazione carica dati formattati e puliti in Amazon S3. In questo caso utilizziamo di nuovo Amazon S3 perché Amazon Redshift è in grado di caricare i dati in parallelo da Amazon S3, utilizzando più thread di ciascun nodo di cluster. Amazon S3 offre, inoltre, un record storico e funge da "source of truth" formattata tra i sistemi. I dati su Amazon S3 possono essere utilizzati da altri strumenti a scopo di analisi qualora vi sia l'introduzione successiva di requisiti ulteriori.
4. Amazon Redshift carica, ordina, distribuisce e comprime i dati nelle sue tabelle affinché l'esecuzione delle query analitiche avvenga in modo efficiente e in parallelo. Con l'aumentare della dimensione dei dati nel tempo e l'espansione dell'azienda, si può facilmente incrementare la capacità attraverso l'aggiunta di altri nodi.
5. Per visualizzare l'analisi, è possibile utilizzare Amazon QuickSight o una delle numerose piattaforme di visualizzazione dei partner che si connettono ad Amazon Redshift tramite ODBC o JDBC. Questo è il punto in cui il CEO e il suo personale visualizzano rapporti, pannelli di controllo e grafici. I dirigenti possono ora utilizzare i dati per prendere decisioni migliori riguardo alle risorse aziendali incrementando, in definitiva, gli utili e il valore per gli azionisti.

Si può espandere facilmente questa architettura flessibile di pari passo con l'espansione dell'attività, l'apertura di nuovi canali, il lancio di ulteriori applicazioni mobili specifiche per i clienti e l'impiego di origini dati aggiuntive. Bastano pochi clic nella Console di gestione Amazon Redshift o alcune chiamate API.

Conclusioni

Assistiamo a una transizione strategica nel data warehousing in un momento in cui le imprese effettuano la migrazione di database e soluzioni analitici dal locale al cloud, per sfruttare la semplicità, le prestazioni e la convenienza del cloud. Questo whitepaper fornisce un resoconto completo della situazione attuale del data warehousing in AWS. AWS fornisce un'ampia gamma di servizi e un solido ecosistema di partner che consentono di realizzare e gestire facilmente il data warehousing aziendale nel cloud. Il risultato è un'architettura di analisi estremamente performante e conveniente, in grado di adeguarsi alla crescita del business attraverso l'infrastruttura globale AWS.

Collaboratori

Le persone e le organizzazioni indicate di seguito hanno collaborato alla stesura di questo documento:

- Babu Elumalai, solutions architect, Amazon Web Services
- Greg Khairallah, principal BDM, Amazon Web Services
- Pavan Pothukuchi, principal product manager, Amazon Web Services
- Jim Gutenkauf, senior technical writer, Amazon Web Services
- Melanie Henry, senior technical editor, Amazon Web Services
- Chander Matrubhutam, product marketing, Amazon Web Services

Letture ulteriori

Per ulteriore assistenza, consultare le seguenti fonti:

- Libreria software Apache Hadoop[18]
- Best practice Amazon Redshift[19]
- Architettura Lambda[20]

Note

1 https://www.forrester.com/report/The+Forrester+Wave+Enterprise+Data+Warehouse+Q4+2015/-/E-RES124041

2 http://aws.amazon.com/streaming-data/

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

4 http://docs.aws.amazon.com/redshift/latest/APIReference/Welcome.html

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

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

7 http://aws.amazon.com/redshift/partners/

8 https://aws.amazon.com/vpc/

9 https://aws.amazon.com/cloudtrail/

10 http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-HSM.html

11 https://aws.amazon.com/kms/

12 http://aws.amazon.com/s3/pricing/

13 http://aws.amazon.com/redshift/pricing/

14 http://docs.aws.amazon.com/redshift/latest/dg/json-functions.html

15 https://aws.amazon.com/dms/

16 https://aws.amazon.com/redshift/partners/

17 http://aws.amazon.com/ec2/spot/

18 https://hadoop.apache.org/

19 http://docs.aws.amazon.com/redshift/latest/dg/best-practices.html

20 https://en.wikipedia.org/wiki/Lambda_architecture