Domande Cloud Computing

Domande Vallati

1. Ceph

- Qual'è il vantaggio di Ceph rispetto alle soluzioni che offrono un approccio diverso (Data replica, non c'è una dipendenza con la lookup table) (soluzioni ad es file system distribuito) (poche cose, secondo te)
- RAID
- Supponiamo di avere 3 macchine con CEPH installato sopra ed ogni macchina ha 5 dischi. Abiliteresti RAID in aggiunta a Ceph e se sì quale? Raid 0
  Ho 5 dischi sulla macchina, se metto raid li faccio diventare uno solo, e me li gestisce lui levo overhead id avere 5 osd deamons su una singola macchina
  - Come funziona il processo di scrittura di un dato? Risposta: algoritmo crush
  - vantaggio Ceph rispetto ad altre soluzioni (es FS distribuito, GlusterFS, NFS)? Risposta: principalmente, la replicazione dei dati

2. DevOps

- Cos'è il concetto di DevOps? Pro/Contro?
- Service broker
- Kubernetes (architettura)
- cos'è un pod (insieme di container su stessa macchina ...)
- come app di kubernetes vengono esposte all esterno
- per quale motivo un app non cloud migrata sul cloud non puo sfruttare al massimo i vantaggi di essere sul cloud? (scalabilità verticale è limitata e costosa)

3. OpenStack

- dove abbiamo visto la tecnologia gre oltre a qui e come veniva utilizzata? (neutron controller neutron agent, permetteva isolamento delle vm - In OpenStack per far creare un tunnel agli agent, creare isolamento delle VM ...) gestione virtual networks su OpenStack)
- what is openstack (softw to create cloud platform), architettura di openstack
- Il workflow per creare una nuova VM su Openstack
- Tecnologie di virtualizzazione di rete (OpenStack)
- Quali sono le metodologie per virtualizzare una rete (VLAN, VXLAN, GRE TUNNELING)
- Come funziona la gestione della rete di Openstack (Neutron)
- Architettura di OpenStack
- what is a cloud platform

4. General architecture of global application

- Come si realizzano delle applicazioni su scala globale (ad es. motore di ricerca)? risposta: global e local load balancers
- Applicazioni eventually consistent, com'è che viene implementato il sistema (Bayou architecture) e perchè (applicazioni su larga scala, gestione delle partizioni di rete, performance)
- perche c'è bisogno del tunnel nel load balancer?

5. Strategie di data replication

- A grandi linee, come si fa comunicazione frontend backend (scambio di messaggi)
- Sistemi Msg-queue
- opzioni che sviluppatore ha per implementare comunicazione tra moduli di una cloud application(paradigma di comunicazione->produttore e consumatore, quali sono vantaggi, implementazione di broker che abbiamo visto->RabbitMQ)

6. vantaggi applicazioni basate su scambio messaggi (SOA)

- Modo con cui si creano le interfacce verso gli utenti di un'interfaccia cloud (Soap, rest e indirect communication)
- paradigmi di comunicazione verso esterno o verso applicazione (rest o soap)
- Principali differenze tra SOAP e REST (WSDL)
- Quale protocollo utilizzare per un'applicazione tipo candy crush, o di un servizio bancario (1- Rest, 2- Soap)
- Protocolli di comunicazione fra moduli? SOAP, REST
- Come si creano le interfacce verso i Client in un applicazione Cloud? Risposta: SOAP, REST, Code, Msg broker
- Differenza tra questi approcci?

7. Virtualization

- Lightweight virtualization options
- (differences and advantages/disadvantages compared to full virtualization) hypervisor
- full virtualization when some instructions cannot run in vm
- cosa sono i container
- vantaggi svantaggi container e full virtualized environment
- tipi di servizio che un cloud provider puo fornire ai clienti
- differenza public cloud e private cloud (e altri tipi)
- quale tipo di infrastruttura (private,public, ma anche iaas e saas) nel caso di una startup con ricette di cucina , e nel caso di una multinazionale che vende apparecchi biomedici
- struttura applicazioni cloud computing anche in base agli obiettivi che si vogliono raggiungere
- cosa è hypervisor, tipologie disponibili
- meccanismi di virtualizzazione, meccanismi per creare una rete virtuale
- vantaggi virtualizzazione

- spiegare meccanismo multi tenancy
- processo della cloudification, cos è, vantaggi svantaggi
- trap e emulate in virtualizzazione cosa è, è ancora utilizzato frequentemente?
- la para virtualization caratteristiche, come si ovvia al problema di dover fare trap e emulate (hypercalls)
- trategie di data replication
- come funziona gossip
- principali vantaggi di adottare cloud computing (da punto di vista stratup e grande azienda e aziende che hanno bisogno di potenza computazionale con spike e periodi morti)
- applicazioni eventually consistent, perche e quando la adotto
- cos è harware pass through, vantaggi e svantaggi
- sistemi di scambio messaggi basati su code
- differenze tra full e para virtualization
- Vantaggi e svantaggi dei container comparato a full virtualization
- Come avviene l'invio dei pacchetti in questi contesti? (incapsulamento, tunnel)
- Perché abbiamo bisogno del tunnel in questo caso? Il load balancer non potrebbe mandare i pacchetti direttamente senza tunnel? (Per avere come destinazione il local load balancer)
- Quali sono i vari tipi di servizio che un Cloud Provider può servire ai propri clienti?
- Supponiamo che tu sia il progettista di una nuova app per smartphone dove utenti inseriscono ricette di cucina e gli altri le ritrovano. Sup/poniamo invece che tu sia il progettista di una multinazionale che opera nel campo biomedico perché produce dispositivi biomedici per il monitoraggio dei pazienti. Quale tipo di infrastruttura e servizio suggeriresti nei due casi? (startup; andare su un cloud public, PaaS o SaaS se esiste; multinazionale andare su private cloud, IaaS)
- Quali sono le strutture cloud che si adottano in base al tipo di obiettivo che si vuole raggiungere? (HA, LB e CI)
- Come integrare i paradigmi dei messaggi nell'architettura cloud modulare
- Hypervisor e tipologie
- Quale tipo di hypervisor scegliere se hai 50000 vm tutte uguali? (Bare metal approach)
- Metodologia di virtualizzazione della CPU
- Vantaggi della virtualizzazione
- Meccanismo della multi-tenancy che vantaggi da?
- Processo della Cloudification e vantaggi
- Meccanismo trap (emulazione di un istruzione privilegiata) e come è stata superata (root-non root mode)
- indirect communication è popolare
- Le para-virtulizzazioni come evitano emulazione? (Hypercalls)
- Strategie di data replication nelle cloud applications (active, passive, gossip)
- I vantaggi delle cloud applications con casi d'uso (start up vs. grandi aziende)
- Service-oriented architecture (perchè offre un packaging dei messaggi standardizzato)
- Struttura generale delle architteture cloud (HA, LB e CI)

- Differenza tra public cloud/private cloud + community
- Sistemi di supporto hardware alla virtualizzazione
- Sistemi di scambio di messaggi basati su code
- Tipologie di servizio erogate da un Cloud Provider
- Struttura (in generale) delle Architetture Cloud
- Caratteristiche & tecnologie del backend. Risposta: SOA
- Quali tecnologie usa la CPU per virtualizzare la stessa CPU e RAM? Risposta: cose su cambio contesto ecc
- Metodologie per creare Virtual Networks? Risposta: Neutron
    - Come fanno piattaforme tipo Xen ad evitare il Trap & emulate? Risposta: hypervisor espone specifiche funzioni choiamate hypercalls che sostituiscono le istruzioni privilegiate
- Cos'è la Sequential Consistency?
- Struttura generaledelle applicazioni Cloud? Risposta: high-avail clusters ecc

8. RAID

- Secondo te, RAID è utilizzabile/consigliato nel realizzare cloud storage economico stile Ceph?
- RAID è infallibile come tecnologia?

Laboratorio Vallati

1. Procedura per creare un nuovo container in Docker e per lanciarlo. Il flusso di lavoro
2. Cos'è libvirt o kvm?
   2.1) Come funziona l'istanziazione di una macchina virtuale utilizzando libvirt/kvm? Che cosa devo fare?
   2.2) Dal punto di vista della rete come funziona? Le macchine virtuali su quali reti vengono inserite?
3. Come funziona il flusso di lavoro con Kubernetes? In generale, cos'è Kubernetes?
   3.1) Come avviene il processo di installazione di OpenStack? Come si installano le componenti di OpenStack sulle varie macchine una volta aggiunte?
   Che macchine ci sono e che ruoli hanno quelle che avete installato voi?
   3.2) Dove si esegue "juju" e a cosa serve? Qual è il flusso di lavoro con cui viene utilizzato?
4. Stessa domanda su Kubernetes perché l'altro non ha saputo rispondere. (non ha risposto nemmeno lui)
5. Cos'è Kubernetes?
   5.1) Qual è il workflow di Kubernetes? Ovvero, uno sviluppatore che sviluppa un'applicazione e vuole farla girare su Kubernetes, che deve fare?
   5.2) Cosa c'è dentro il descrittore di app?
6. Come funziona l'esposizione di un container verso l'esterno su Docker? Cosa vuol dire esporre un container verso l'esterno e cosa comporta questa operazione?
7. Cos'è kvm libvirt?
   7.1) Come accedono le macchine virtuali alla rete?

8. Una delle caratteristiche di Kubernetes è quella di gestire il ciclo di vita delle applicazioni e in particolar modo dei container. Tutto in maniera automatica. Ad esempio c'è la gestione della replica delle istanze di un container in maniera automatica. Come funziona questo meccanismo?
9. Cos'è Kubernetes?
   9.1) Differenze tra Kubernetes e Docker?
10. Cos'è juju?
    10.1) Quali sono i passaggi per installare una componente di Openstack e più in generale in juju?
11. Workflow di Docker
12. Come fanno le macchine virtuali create attraverso libvirt kvm a connettersi ad internet? Quali sono le opzioni che abbiamo visto?
    12.1) Passaggi per creare una macchina virtuale con libvirt
13. In Kubernetes che cos'è un POD?

Tonellotto

- Come gestisce la fault-tolerance Spark?
- Metodo cache() di Spark è una trasformazione o azione? (r: azione)
- Cos'è un partitioner? Come funziona il partitioner di Hadoop?
- Perchè nei paradigmi visti a lezione per MapReduce bisogna che le funzioni che passiamo siano "stateless"? Le cossiddette higher-order function
- Ruolo del partitioner in MapReduce
- Descrivere approccio stripes vs paired
- Cos'è il Lineage? (Spark)
- Differenza tra trasformazioni wide e narrow?
- Definizione formale di paradigma MapReduce
- Cos'è HDFS? Responsabilità namenode/datanodes
- Differenze tra Trasformazioni e Azioni (Spark)
- Come implementeresti una join a tuple in MapReduce Hadoop?
- Come mai bisogna tenere d'occhio l'occupazione della memoria in Mapreduce in Hadoop?
- Cos'è lo shadow namenode?
- A cosa serve il Combiner?
- Moltiplicazione vettore x matrice in mapreduce

- Architettura di Spark
- Cos'è l'in-memory combining?
- Cos'è la rack topology?
- funzione broadcast in Spark
- accumulator in Spark
- Cosa sono gli struggler? (r: worker particolarmente lenti in hadoop mapred)
- Scheduling in YARN
- Spiegare codice progetto(getConfiguration.setInt-> set shared variable read-only, perché implements Writable,Comparable)
- resource manager
- fare map reduce per comprimere un miliardo di pagine web(dove si fa la compressione->in mapper perché si possono avere quanti mapper ci pare)
- Definizione formale di paradigma di programmazione MapReduce
- Cosa é un input split
- Cos'è HDFS? Che architettura viene utilizzata in HDFS? Parla dell'architettura Master Slave.
- Differenza trasformazioni e azioni in Spark
- Un'operazione di join tra tuple in MapReduce con Hadoop, come la implementeresti? (visto a lezione)
- Perché bisogna controllare l'occupazione della memoria nei programmi MapReduce, in particolare su Hadoop? (Evitare che il GarbageCollector sia saturo)
- Cos'è lo shadow name node? (componente di Hadoop che non ha a che fare con i job ma con HDFS)
- A cosa servono i Combiner?
- Cos'è YARN, qual è la sua architettura e quali sono i suoi componenti? Cos'è un Container?
- Come si implementa la moltiplicazione di una matrice per un vettore in MapReduce?
- Architettura del framework Spark (Architettura Master Slave, Driver ed Esecutori)
- Cosa è un partitioner
- Cos'è inmemorry mapper
- rack topology o network topology
- differenza tra pairing e qualcosaltro

- Cosa è lineage
- Perche l rdd è immutabile
- definizione di paradigna di prog mapreduce (no hadop)
- cosè input split
- caratteristiche file sistem distribuito: high availability e affidabilita ottenuta con meccanismo di replicazione attraverso i vari datanode (hdsf sa dove sono i nodi) la prima copia gestita direttamente localmente in hdfs, la seconda replica in un altro rack in un nodo casuale, la terza allocata in un altro nodo dello stesso rack il secondo
- fault tolerance il name node responsabile della gestione dei dati decide dove archiviare i dati, il client contatta il name che dice al client dove archiviar ei blocchi tutto gestito dal namenode è un spof quindi, deve esserci active standby in caso di fail del namenode sositturlo prontamente , le strutture vanno replicate o storage condviso o database, l stato va replicato cosi ogni macchina nel momento in cui viene richiesto di essere attiva deve essere pronta.
- data locality si costriusce il codice
- che tipo di op si svolgono tipicamente (accesso sequenziale) non ho bisogno di accesso casuale come avviene nei normali os
- differenza tra trasformazione e azione
- implementazione di un operazione di join in hadoop
- come mai bisogna tenere sotto controllo l'occupazione della memoria nei programmi map reduce
- cos'è lo shadow namenode
- che succedeva se nn usavo il combiner? cosa avrei visto
- moltiplicazione tra matrici: la matrice viene suddivisa per righe e ogni mapper fara al sua molt riga per colonna che è il vettore che è in memoria e in uscita da l indice del vettor edi uscita in cui salvare il risultato prodotto riga colonna fatto dal mapper il reducer ha l identita
- architettura di spark framework : MASTERSLAVE driver e esecutori
- l esempio del mapreduce immagini cosa prende in input
- accumulator
- a che serve il partitioner serve quando la distribuzione dei valori tra lo spazio intermedio delle chiavi
- Qual è la problematica dei grafi per gli algoritmi in map reduce risposta bisogna mantenere la struttura del grafico tra un iterazione e l'altra e quindi bisogna propagarla