

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gregor Cimerman

Orodja za napreden nadzor gruče
Hadoop

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTORICA: doc. dr. Mojca Ciglarič

Ljubljana 2013

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00101/2013

Datum: 11.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GREGOR CIMERMAN**

Naslov: **ORODJA ZA NAPREDEN NADZOR GRUČE HADOOP
TOOLS FOR ADVANCED HADOOP CLUSTER CONTROL**

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje


Tematika naloge:

Opišite sistem za hranjenje in obdelavo velikih količin podatkov Hadoop ter vgrajene možnosti za upravljanje in nadzor sistema. Identificirajte slabosti takšnega nadzora in poiščite upravljalska orodja, ki te slabosti zmanjšujejo.

Izberite optimalno konfiguracijo nadzornih orodij in jih uporabite na prototipni postavitvi Hadoopa. Za analizo sistemskih dogodkov uporabite orodje Splunk, ki naj tudi izvaja alarmiranje.


Tako razširjen nadzorni sistem kritično ovrednotite in primerjajte funkcionalnosti z osnovno različico nadzora.

Mentor:


doc. dr. Mojca Ciglarič



Dekan:


prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Gregor Cimerman, z vpisno številko **63090092**, sem avtor diplomskega dela z naslovom:

Orodja za napreden nadzor gruče Hadoop

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mojce Ciglarič,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 23. septembra 2013

Podpis avtorja:

Zahvaljujem se mentorici doc. dr. Mojci Ciglarič za pomoč pri izdelavi diplomskega dela. Zahvale gredo tudi zaposlenim na Arnesu, ki so mi omogočili izdelavo diplomskega dela. Zahvaljujem se družini za vso podporo tekom študija.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Hadoop	3
2.1	Delovanje Hadoop-a	5
2.2	Sestavni deli	8
2.3	Nadzor	12
3	Nadzor in upravljanje gruče	15
3.1	Metrike	15
3.2	Apache Ambari	17
3.3	Cloudera	18
3.4	Hortonworks	20
3.5	Splunk	20
3.6	Ostala uporabna orodja za nadzor	22
4	Uporaba orodja Splunk za nadzor Hadoop gruče	25
4.1	Lastnosti gruče	25
4.2	Upravljanje gruče s projektom Ambari	27
4.3	Splunk	28
4.4	Obveščanje	35
4.5	Splunk kot nadzornik gruče Hadoop	36

KAZALO

5 Sklepne ugotovitve

39

Povzetek

Hadoop je platforma za shranjevanje in obdelavo velikih količin podatkov. Pri takšnih platformah, ki se raztezajo čez večje število strežnikov, je potreben nadzor s posebnimi orodji. Poleg velikega števila strežnikov je pri nadzoru potrebno upoštevati tudi dinamičnost platforme in možnosti za razširitev. Orodja, ki se jih uporabljajo za nadzor porazdeljenih računalniških sistemov, so lahko zelo specifična in omejena glede funkcionalnosti, zato je za nadzor Hadoop gruče potrebna kombinacija orodij. Zaradi priljubljenosti in uporabnosti platforme Hadoop smo se odločili, da raziščemo orodja za nadzor, ki nudijo celovito rešitev in lahko sledijo dinamičnosti ter razširljivosti platforme. Postavili smo testno gručo, na kateri smo primerjali delovanje posameznih orodij in opisali težave, na katere smo naleteli. Predlagali smo optimalno kombinacijo orodij za nadzor, opisali njihovo konfiguracijo in jo preizkusili.

Ključne besede:

Hadoop, gruča, Splunk, Ambari, nadzor, upravljanje, HDFS, metrike, dnevniške datoteke, veliki podatki

Abstract

Hadoop is a platform for storing and processing big data. This kind of platform that stretches over multiple servers is difficult to manage. Traditional management systems for computer grids do not allow complete management over Hadoop services because the dynamic and elastic properties. Because the complexity of Hadoop services, the combination of management systems are necessary for complete management. In this thesis we describe this combination of tools for Hadoop cluster management. We deployed test cluster which serves as the base for testing management systems. At the end we described other solutions and problems we encountered in our testing.

Keywords:

Hadoop, cluster, Splunk, Ambari, control, management, HDFS, metrics, logs, big data

Poglavje 1

Uvod

Hadoop je v zadnjih nekaj letih pridobil velik sloves kot platforma s katero je mogoče na enostaven način rešiti problem obdelave velike količine podatkov ali velike podatke (angl. Big Data). Veliki podatki v stroki pomenijo veliko količino podatkov, ki jih proizvedejo raznovrstni računalniški sistemi. Običajno so ti podatki v obliki dnevniških datotek ali velikih tabel, in jih je težko shraniti na en sam sistem še težje pa obdelati v doglednem času. Ocenjuje se, da bo Hadoop v naslednjih nekaj letih na trgu dosegel več kot polovični delež vseh rešitev za omenjene probleme. Prednost pred konkurenco je predvsem v odprtosti in veliki skupnosti, ki skrbi za konstanten razvoj in nadgradnje. Dokaz uporabnosti in zanesljivosti platforme Hadoop je število projektov, ki koristijo prednosti dveh glavnih komponent Hadoopa; Hadoop porazdeljen datotečni sistem HDFS (angl. Hadoop Distributed File System) ter računski del platforme imenovan MapReduce (poimenovan po dvofazni obdelavi podatkov; faze Map ter faze Reduce). Hadoop je licenciran pod licenco Apache v2, kar pomeni, da ga lahko uporablja vsak. To odprtost so dobro izkoristila velika podjetja, ki tudi vračajo svoje izboljšave v skupnost in tako se izvorna koda projekta Apache Hadoopa še razvija in vzdržuje. Danes je na trgu nekaj komercialnih ponudnikov platforme Hadoop, ki jo prilagodijo za bolj specifične probleme ter poimenujejo v stilu dosedanjih produktov. Običajno podjetja izdajo lastno distribucijo platforme

in nudijo licenčna orodja prilagojena za upravljanje in nadzor te distribucije. Na voljo so tudi brezplačne verzije njihovih produktov, toda z omejeno funkcionalnostjo ali pa zgolj za preizkusno obdobje. Večina (tudi komercialnih) rešitev ponuja dodelana upraviteljska orodja toda za nadzor gruče je manj možnosti, saj so pokriti samo osnovnejši parametri. Cilj diplomskega dela je izbrati orodje, ki dopolnjuje funkcionalnost privzetega nadzornika Hadoop gruče. Dober primer privzetega nadzornika je Ambari, ki je prav tako licenciran pod Apache v2 licenco in za nadzor uporablja orodja Nagios in Ganglia. Težava z omenjenima nadzornikoma je slaba prilagodljivost in omejen pogled v delovanje, saj lahko nadzirata izključno prednastavljene parametre. Splunk omogoča iskanje po podatkih, posredovanih s strani delovnih strežnikov, gradnjo pogledov, poročil in obvestil. Administratorju platforme Splunk nudi podrobnejši vpogled v delovanje v realnem času kot tudi zgodovine dogodkov, kar pomaga pri optimiziranju ali odpravi težav.

Poglavje 2

Hadoop

Apache Hadoop je odprtokodna platforma, ki omogoča zanesljivo in skalabilno infrastrukturo za hranjenje in obdelavo podatkov. Sestavljen je z dveh glavnih komponent; Hadoop distribuiran datotečni sistem imenovan HDFS (angl. Hadoop Distributed Filesystem) ter procesne komponente imenovane MapReduce. Hadoop ima svoje korenine v projektu Apache Nutch, ki ga je v času razvijal Douga Cutting, avtor Apache Lucene, znane knjižnice tekstovnega iskalnika. Med razvojem Apache Nutch-a sta Mike Cafarella in Doug Cutting razvila sistem, ki bi premogel bilijonski indeks spletnih strani vendar uporabljena arhitektura ni bila dovolj skalabilna za večje kapacitete. Projekt Nutch, ki se je pričel leta 2002, je doživel preskok leta 2003, ko je podjetje Google objavilo članek o googlovem datotečnem sistemu GFS [3] (angl. Google File System). Arhitektura googlovega datotečnega sistema je bila prilagojena za hranjenje velikih podatkovnih datotek ter shranjevanje podatkovnih tokov. Leta 2004 je Google objavil še članek o svojem programskem modelu MapReduce [8], ki je bil naslednje leto že uspešno implementiran v projektu Nutch. HDFS in MapReduce sta se v začetku leta 2006 odcepila od projekta Nutch in formirala nov podprojekt imenovan Hadoop, toda še vedno kot podprojekt projekta Apache Lucene.

Pred razvojem Hadoop-a so bili podobni sistemi zelo dragi in so za delovanje potrebovali zanesljivo strojno opremo, ki običajno ni bila združljiva

med različnimi ponudniki. Hadoop deluje z uporabo strojne opreme različnih ponudnikov in za svoje delovanje ne potrebuje zahtevne strojne opreme temveč povprečno ali cenovno ugodno (angl. Comodity Hardware). Običajno so podjetja, ki so naletela na težavo hrambe in obdelave velike količine podatkov, sama zaposlila inženirje, ki so sami razvili sistem, ki je bil nalogi kos. Takšni sistemi, ki so bili razviti interno, so bili prilagojeni za reševanje specifičnega problema zaradi česar običajno niso bili objavljeni. Vsak naročnik takega sistema je porabil veliko resursov za implementacijo ter razvoj. Razvoj Hadoop-a je ta sistem računskih gruč odprl širši javnosti.

Hadoop je neodvisen od ponudnika strojne opreme, saj je celotna platforma aplikativna rešitev (deluje na sedmem nivoju sklada ISO-OSI) in je prilagojena za delovanje na strojni opremi z osnovnimi komponentami, ki je cenejša. Z uporabo osnovnih komponent je čas med dvema napakama krajši od tistih, ki so namenjeni za produkcijo in visoko zanesljivost. V primeru majhnih gruč je število napak dovolj majhno, da lahko en sam administrator pravočasno ukrepa ob vseh nastalih težavah. V večjih gručah je število napak, pri uporabi enakih komponent, višje. Nič neobičajnega ni, če je 2% - 5% vse strojne opreme v okvari ali fazi odprave težave. Prilagojenost delovanja na osnovni računalniški opremi je urejena na aplikativnem nivoju, saj HDFS samodejno zazna poškodovane ali nedosegljive bloke podatkov ter jih ponovno razmnoži in razprši po gruči. Podobno kot z bloki podatkov, je tudi pri MapReduce nalogah, kjer v primeru prekinitve naloge, organizira ponovno izvajanje tako, da jih doda v čakalno vrsto. HDFS ima parameter minimalnega števila kopij vsakega bloka podatkov, imenovan `dfs.replication.min` in določa število kopij, ki se morajo stalno hraniti na gruči. Privzeto je parameter nastavljen na 3 in ga lahko spremeni administrator.

Hadoop je na trg prinesel nov pogled na podatke. Pred prihodom platforme Hadoop so bili podatki v večini primerov shranjeni v enem podatkovnem sistemu, pogosto organizirani v tabelah. Razvoj računalniške tehnologije je povzročil porast količine podatkov in potrebo po hitrejši obdelavi. Največ podatkov predstavljajo strojne dnevniške datoteke ter meritve različnih senzor-

jev oz. tipal. Ti podatki so eden izmed razlogov za razvoj platforme Hadoop, kot jo poznamo sedaj. Hadoop je v primerjavi z običajnimi relacijskimi bazami podatkov, porazdeljen in odporen na motnje, omogoča hrambo velike količine podatkov, dinamično shemo, skalabilnost ter hitro obdelavo podatkov. Platforma za počasnost pri pisanju in spreminjanju podatkov žrtvuje za prilagodljivost, hitrost pri branju in veliko kapaciteto datotečnega sistema.

Ponudniki visoko zmogljivih računskih gruč (HPC High Performance Computing) ponujajo različne rešitve za opravilo nalog, vendar so pogosto načrtovana s centraliziranimi sistemi. Centraliziran sistem ima prednost v enostavni postavitvi in zanesljivem delovanju. Olajšan je tudi nadzor in upravljanje, saj specializirani administratorji, ki skrbijo za specifično opremo hitreje in bolj učinkovito ukrepajo kot ostali. Različni sistemi, kot so diskovna polja, procesne komponente, omrežna oprema, ipd., ki potrebujejo posebno oskrbo so ločeni in omogočajo neprekinjeno delovanje tudi v primeru izvajanja vzdrževalnih opravil. Omenjena rešitev v določenih primerih zadostuje potrebam, vendar ima omejitve kot so zmogljivost, prepustnost podatkov ter visoko ceno. Taki sistemi običajno dosežejo svoje zmogljivostne omejitve v primeru velikega števila hkratnih zahtevkov za podatke. Primer zahtevkov po veliki količini podatkov je pri zagonu naloge na porazdeljenem računskem sistemu, kot je MapReduce.

2.1 Delovanje Hadoop-a

Platforma Hadoop je odprtokoden projekt, ki se razvija pod okriljem Apache skupnosti. Izvorna koda je napisana v programskem jeziku Java zaradi česar je, v večini primerov, neodvisna od operacijskega sistema. Hadoop se namesti na operacijski sistem, ki podpira Java, kot dodatna programska oprema. Običajno se namesti na več strežnikov ali gručo strežnikov, ki skupaj tvorijo en sistem. Podatkovni strežniki (angl. Data Node) se med seboj sporazumevajo preko TCP povezav, ki jih vzpostavijo s pomočjo IP (angl. Internet Protocol) protokola. Strežniki so hierarhično urejeni. Imen-

ski strežnik (angl. Name Node) in sekundarni imenski strežnik (angl. Secondary Name Node) sta dva namenska strežnika, na vrhu hierarhije, z ekvivalentno ali boljšo konfiguracijo strojne opreme od podatkovnih strežnikov. Strežnika koordinirata podatkovne tokove, skrbita za organizacijo in pravice datotečnega sistema, nadzirata delovanje gruče in v primeru izpada podatkovnega strežnika naloge in podatke, ki so se tam hranili, porazdelita po preostalih podatkovnih strežnikih.

Imenski strežnik hrani metapodatke v svojem delovnem pomnilniku (RAM - angl. Random Access Memory) za hitrejšo dostopnost in krajši odzivni čas. Slabost delovnega pomnilnika je v neobstoynosti po odvzemu električne energije. Ker datotečni sistem ne more delovati brez metapodatkov se periodično shranjujejo slike datotečnega sistema na magnetni disk imenskega strežnika. V HDFS se metapodatki delijo na dva dela - slika datotečnega sistema imenovana `fsimage` in spremembe v datotečnem sistemu, imenovane `edits`. Slika datotečnega sistema hrani trenutno stanje datotečnega sistema v spremembe pa se shranjujejo le spremembe narejene na datotečnem sistemu. Ta rešitev je tipična za datotečne sisteme ki podpirajo veliko podatkovno prepustnost. Ker se zapisi o spremembah stalno dodajajo, bi se datotečni sistem kmalu upočasnil do te mere, da bi prenos podatkov potekal hitreje, kot pa sama pridobitev informacij o stanju podatkov. Tukaj priskoči na pomoč sekundarni imenski strežnik, ki je zadolžen za posodobitev slike datotečnega sistema iz danih podatkov o spremembah. Torej imenski strežnik svoje trenutno stanje in vse spremembe preda sekundarnemu imenskemu strežniku, ki posodobi sliko datotečnega sistema in jo vrne imenskemu. Imenski strežnik nato nadomesti svojo aktivno sliko datotečnega sistema s popravljeneno ter zavrže spremembe, ki so pripeljale do novega stanja datotečnega sistema. Torej imenski strežnik in sekundarni imenski strežnik sta drug od drugega odvisna in se po zgledu dobre prakse nahajata na ločenem fizičnem strežniku, čeprav platforma dopušča tudi konfiguracijo kjer se nahajata na istem strežniku.

Imenski strežnik je ključnega pomena za delovanje celotnega sistema in v

primeru izpada postane celoten sistem neuporaben. Torej imenski strežnik predstavlja samostojno točko odpovedi (angl. single point of failure). Kljub imenu, sekundarni imenski strežnik ni namenjen za prevzem vseh nalog imenskega strežnika. Razvijalci platforme so v verziji 0.23 izdali rešitev v obliki visoko razpoložljivega imenskega strežnika (angl. High Availability - HA). Administrator lahko določi par strežnikov, ki služita kot imenska in v primeru izpada aktivnega strežnika delo prevzame pasivni. Strežnika se ločita glede na aktivnost, ki jo imata na Hadoop datotečnem sistemu; aktivni (strežnik na katerega se povezujejo uporabniki in servisi s poizvedbami) in pasivni (strežnik na katerega se prenesejo vse spremembe na datotečnem sistemu z aktivnega imenskega strežnika. Strežnik je torej aktivna kopija, ki ne streže poizvedbam). V primeru izpada aktivnega imenskega strežnika, pasivni imenski strežnik prevzame vlogo aktivnega po času, ki je prednastavljen, kot dopusten odzivni čas aktivnega imenskega strežnika. Pasivni imenski strežnik mora ves čas nadzirati stanje in dosegljivost aktivnega ter v primeru izpada zagotoviti, da le-ta ne vzpostavi povezave, saj bi s tem lahko povzročil stanje razcepljenih osebnosti (angl. split brain). Stanje razcepljenih osebnosti je pojav, ki se lahko pojavi v redundantnih sistemih, saj lahko zaradi prekinjene sinhronizacije med dvema sistemoma, sočasno uređita isti podatek, kar ob ponovni sinhronizaciji povzroči napako. Za take primere ima pasivni strežnik možnost "ubiti" aktivnega (angl. STONITH ali shoot the other node in the head). Znotraj platforme je omogočen tudi eleganten prevzem (angl. Graceful Failover) za potrebe preventivnega pregleda aktivnega strežnika.

Prav tako, kot ima Hadoop datotečni sistem vsaj dva strežnika, ki skrbita za koordinacijo in avtorizacijo akcij na datotečnem sistemu ima MapReduce ločen strežnik, ki skrbi za izvedbo nalog in v primeru napake na delovnem strežniku, uredi ponovni zagon naloge in o statusu obvesti administratorja.

2.2 Sestavni deli

Hadoop sestoji z dveh ključnih komponent, to sta Hadoop porazdeljen datotečni sistem (HDFS) in procesna komponenta MapReduce.

2.2.1 HDFS

Hadoop distribuiran datotečni sistem je glavni del platforme, ki omogoča velike pretoke podatkov, tokovno branje in hrambo velikih datotek. Omogoča hrambo datotek večjih od kapacitete največjega diska v gruči.

HDFS se razlikuje od običajnih datotečnih sistemov. Porazdelitev preko večjega števila strežnikov, kjer vsak prispeva svoje omrežne vmesnike, pomnilniške kapacitete in procesorski čas za sestavo enega velikega datotečnega sistema poveča njegovo skalabilnost. Datotečni sistem podpira večino POSIX (angl. Portable Operating System Interface) standada, kar poenostavi uporabo. Primeren je za uporabo, kjer prevladuje tokovno branje podatkov in je zahtevana odpornost na odpoved komponent. Hramba velikih datotek na porazdeljenem datotečnem sistemu, kot je HDFS, je mogoča s paketno razdelitvijo datotek, kjer ima imenski strežnik nadzor in pregled nad vsemi paketi.

Ker HDFS ne bazira na jedru operacijskega sistema, temveč je programska oprema in jo operacijski sistem vidi kot proces, so za komunikacijo potrebni posebni vmesniki. Da se razvijalcem aplikacij za datotečni sistem ni potrebno ukvarjati z organizacijo datotečnega sistema in podatkovnimi tokovi je na voljo tudi visoko nivojski aplikacijski programski vmesnik (angl. Application Programming Interface - API). Preko API vmesnika lahko razvijalec izvaja različne operacije na datotečnem sistemu, osnovni dve sta branje in pisanje.

Stanje datotečnega sistema je potrebno periodično preverjati. V HDFS se preverba datotečnega sistema izvaja periodično vsako uro. Podatkovni strežniki pošljejo imenskemu strežniku poročilo s stanjem podatkovnih blokov, ki jih hranijo, imenski strežnik te podatke posodobi in ustrezno ukrepa. Slaba lastnost te rešitve je v tem, da v primeru vnovičnega zagona gruče, imenski

strežnik v roku ene ure obnovi stanje datotečnega sistema.

Ker se metapodatki datotečnega sistema hranijo v delovnem pomnilniku imenskega strežnika se je pomembno zavedati ocene, da milijon blokov na datotečnem sistemu potrebuje za svoje delovanje okvirno 1GB metapodatkov. Ta podatek je zelo pomemben pri načrtovanju postavitve nove gruče Hadoop.

2.2.2 MapReduce

MapReduce je linearno skalabilen programski model in je ena od dveh glavnih komponent platforme Hadoop. Vgrajen je v jedro platforme Hadoop in prilagojen za uporabo skupaj z HDFS datotečnim sistemom. Razvijalcem programske kode, se ni potrebno ukvarjati s kompleksnimi sinhronizacijskimi in večnitnostnimi problemi saj programski model poskrbi za paralelno izvajanje, zagotavljanje kakovosti izvajanja in skalabilnost in odpornost na napake. Uporabniki morajo sistemu predati dve funkciji; funkcijo Map in funkcijo Reduce, ter konfiguracijsko datoteko. Nalogo nato izvede celotna gruča oz. prednastavljen delež gruče. Sistem poskuša naloge zagnati na strežnikih, ki že vsebujejo podatke potrebne pri izvedbi funkcij. Ta način izbire virov se imenuje lokalnost podatkov (angl. Data Locality). Tako MapReduce skrajša čas izvajanja, saj je mogoče do lokalnih podatkov priti hitreje, kot podatkov, ki se nahajajo na ločenih strežnikih. Zaradi prenosa podatkov preko omrežja, bi se izvajanje naloge bistveno podaljšalo. Ravno zaradi lokalnosti podatkov, MapReduce za svoje delovanje priporoča funkcionalen HDFS, ki je nameščen na istih strežnikih, na katerih se bodo izvajale naloge.

Razvijalec MapReduce nalog napiše dve funkciji; funkcijo Map ter funkcijo Reduce. Platforma nato poskrbi za paralelno izvajanje zadane naloge. Ti dve funkciji nimata omejitev glede količine podatkov kot tudi stopnje paralelnosti, torej jih lahko obdela en računalnik ali gruča računalnikov. Dostop do ogrodja MapReduce je omogoča s pomočjo namenskega API-ja, kar za razvijalce pomeni uporabo klienta za dodajanje nove naloge. Po oddaji naloge se na izbranem delovnem strežniku zažene proces naloge, ki poskrbi

za razdelitev opravil med delovne strežnike, zbiranje podatkov o stanju in vrnitev rezultata. Ta proces se imenuje nadzornik naloge (angl. Jobtracker). Programski model temelji na podatkih oblike: ključ - vrednost, ki sestavljata en podatek. Razdeljena opravila se ustrezno porazdelijo po celotni gruči, kjer vsako opravilo obdela svoj delež podatkov in vrne rezultate, v obliki ključ - vrednost, nadzorniku naloge, ki nato prejete rezultate ustrezno združi in shrani končen rezultat. V koraku Map se običajno pripravi podatke v obliko ključ - vrednost in jih obdela nato sledi korak Reduce, ker rezultate združimo in rezultat preda nadzorniku naloge.

Najbolj pomembna je lastnost skalabilnosti, saj lahko isti program zaženemo na dvakrat večji gruči računalnikov in tako, teoretično, pridobimo rezultate v polovičnem času.

Obdelava podatkov s pomočjo gruče računalnikov se je pojavila pred prihodom platforme Hadoop, vendar je skoraj vsem sistemom največjo težavo predstavljala obdelava velike količine podatkov, saj so delovnih strežniki do njih dostopali preko centraliziranih sistemov za shranjevanje podatkov, ki so bili priključeni na skupno omrežje. Takšna opravila so precej bremenila omrežno opremo, ki je običajno ozko grlo v takšnih sistemih.

2.2.3 Projekti, tesno povezani s platformo Hadoop

Apache projekti, ki dodajajo funkcionalnost in dodajano uporabnost platformi [15], temeljijo na komponentah HDFS in MapReduce.

Apache Ambari je projekt, ki poenostavlja postavitve, konfiguracijo, nadzor in upravljanje gruče Hadoop. Administrator ima popoln nadzor nad delovanjem gruče in lahko poleg HDFS in MapReduce-a nadzira in upravlja z ostalimi Apache projekti, ki so opisani spodaj, preko spletnega vmesnika ter REST vmesnika

Apache Avro projekt za serializacijo podatkov

Apache Chukwa projekt za zbiranje podatkov za upravljanje velikih distribucijskih sistemov

Apache HBase projekt je razširljiva, porazdeljena podatkovna baza, ki podpira hrambo strukturiranih podatkov v velikih tabelah. Izdelana po zgledu googlove velike tabele (angl. Google's Bigtable).

Apache Hive omogoča razvijalcem pisanje ter izvajanje poizvedb podobnim strukturiranem poizvedbenem jeziku SQL (angl. Structured Query Language) imenovanim HiveQL ali Hive poizvedbeni jezik, ki se nato preslika v MapReduce nalogo v gruči. Naloga se porazdeli po celotni ali delu gruče ter vrne rezultat. Je eden izmed najbolj pogosto uporabljenih projektov predvsem zaradi podobnosti SQL-u.

Apache Mahout je programska knjižnica za strojno učenje in rudarjenje podatkov

Apache Pig je visokonivojski podatkovno-tokovni programski jezik za paralelno procesiranje. Temelji na programskem jeziku Java in omogoča dodajanje novih funkcionalnosti. Pig ukazi so podobni programskim jezikom kot so Perl, Python, Ruby, JavaScrip, ipd. se pretvorijo v MapReduce naloge na gruči.

Apache Sqoop , kjer je Sqoop okrajšava za SQL za Hadoop (angl. SQL for Hadoop), je vmesnik ki omogoča prenos podatkov med platformo Hadoop in podatkovno bazo, ki omogoča javanski dostop do podatkovne baze ali bolje znan kot JDBC (angl. Java-based data access).

Apache ZooKeeper je dopolnilo za HDFS in visokozmogljiv koordinator distribuiranih aplikacij. Je tudi obvezna komponenta za izvedbo visoko zanesljivega imenskega vozlišča, saj poskrbi za prenos vlog v primeru težav z imenskim vozliščem.

Dodatna programska oprema, ki dodaja uporabnost računski gruči se stalo razvija in posodablja. Nekaj projektov je tudi na voljo pod licenco Apache v2, v ostalih primerih pa gre za razvoj lastnega projekta, ki rešuje specifično težavo ali komercialne ponudbe. Komercialni ponudniki nudijo

integrirane rešitve projektov v njihove distribucije ter zagotavljajo njihovo medsebojno delovanje. V nasprotnem primeru mora skrbnik poznati delovanje storitev ter nastaviti vse potrebne parametre, ki so potrebni za delovanje projekta na platformi Hadoop.

2.3 Nadzor

Zaradi kompleksnosti in razpršenosti platforme Hadoop v produkcijskih okoljih je za ohranjanje delovanja ter zagotavljanja kakovosti storitev potreben napreden nadzor. Uporaba velikega števila osnovnih računalniških komponent privede do večje verjetnosti pojava okvare, kar predstavlja velik problem pri večjih gručah. Lociranje ter odprava težave je, brez uporabe ustreznih orodij, zamudno opravilo in lahko administrator porabi veliko časa za popravilo. Osnovna računalniška oprema, ki jo lahko uporabimo pri sestavi gruče, ima krajšo življenjsko dobo od tiste, ki je namenjena za produkcijska okolja. V primeru velikih gruč to predstavlja velik delež konstantnega nadziranja in odpravljanja težav.

Uporaba dodatnih orodij je v produkcijskih okoljih skoraj obvezna. Platforma Hadoop ima implementiran vmesnik preko katerega je dostopno operativno stanje in performančne metrike o delovanju sistema. Primer spletnega vmesnika imenskega vozlišča je prikazan na sliki 2.1.

NameNode 'nn.hdp.cime.si:8020'

Started: Thu Sep 12 18:00:58 CEST 2013
Version: 1.2.0.1.3.0.0-107, rd4625cb994e0143f5f4b538f0f2f4a41ad6464a2
Compiled: Mon May 20 02:25:14 EDT 2013 by jenkins
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)

Cluster Summary

88 files and directories, 40 blocks = 128 total. Heap Size is 960 MB / 960 MB (100%)

Configured Capacity	:	111.02 GB
DFS Used	:	2.22 MB
Non DFS Used	:	13.15 GB
DFS Remaining	:	97.87 GB
DFS Used%	:	0 %
DFS Remaining%	:	88.15 %
Live Nodes	:	4
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

Slika 2.1: Spletni vmesnik imenskega strežnika (angl. Name Node) platforme Hadoop

Poglavje 3

Nadzor in upravljanje gruče

Pri nadzoru Hadoop gruče je pomemben splošen pregled delovanja servisov kot tudi možnost za poglobitev v podrobnosti določenega servisa ali strežnika. Pregled delovanja lahko skrajša čas lociranja in odprave napake ali olajša optimizacijo delovanja. Platforma Hadoop omogoča periodično beleženje dogodkov in stanja posameznih procesov ter obdelavo teh podatkov s pomočjo vtičnika. Vtičniki za pogosto uporabljene nadzornike so na voljo že v distribuciji Hadoop-a. Dobljene podatke imenujemo metrike (angl. *metrics*), ki jih logično združujemo v kontekste (angl. *context*) in jih lahko obravnavamo povsem ločeno.

3.1 Metrike

Metrike v platformi Hadoop omogočajo vpogled v informacije o delovanju. Metrike so z razvojem platforme pridobivale nove vrednosti in v prehodu iz verzije 0.20.0 (CHD3) v 0.20.203 (CDH4) doživele obsežnejšo prenovo z generičnimi metrikami in podporo za sočasno uporabo večih vtičnikov. Metrike se ločijo v štiri glavne kontekste:

jvm kontekst vsebuje metrike javanskega virtualnega stroja (angl. Java Virtual Machine). Vse metrike generirajo podatke za ta kontekst.

dfs kontekst vsebuje metrike Hadoop datotečnega sistema. Metrike se razlikujejo glede na vlogo strežnika (imenski strežnik, podatkovni strežnik). Vsebuje podatke o stanju datotečnega sistema, kapacitete, zasednosti, ipd.

mapred kontekst vsebuje metrike MapReduce-a. Tudi te metrike se razlikujejo glede na vlogo strežnika (spremljevalec nalog ali izvajalec nalog). Vsebuje podatke o opravljenih, spodletelih, prekinjenih nalogah, ipd.

rpc kontekst vsebuje metrike klicev oddaljenih procedur (angl. Remote Procedure Call). Vsebuje podatke o trajanju posameznega ukaza, čas čakanja ukaza v čakalni vrsti, ipd.

Metrike lahko po želji vključimo ali izključimo z urejanjem javanske datoteke z lastnostmi imenovane `hadoop-metrics.properties`. Privzeta nastavitve metrik je `NullContext`, kar pomeni, da vsak servis zavrže vse zbrane metrike v sistemsko datoteko `/dev/null` v operacijskem sistemu Linux. Ostali vtičniki metrik, ki jih podpira platforma so `NoEmitMetricsContext`, `FileContext`, `GangliaContext` in `GangliaContext31` ter `JMX` in `REST (+JSON)`. `NoEmitMetricsContext` je vtičnik podoben `NullContext`-u vendat s to razliko, da je trenutne metrike mogoče prebrati, kar s pridom izkoriščata `JMX` in servlet metrik. Vse metrike z izjemo `NullContext`-a porabijo nekaj sistemskih virov, v primerjavi z ostalimi storitvami so le-ti minimalni in vrednost dobljenih metrik upraviči uporabo. `FileContext` periodično shranjuje metrike v določeno datoteko. Težava pri uporabi `FileContext`-a je v načinu implementacije, saj storitev v določeno datoteko stalno dodaja zapise metrik. Rezultat je stalno naraščajoča velikost določene datoteke, ki jo je mogoče izprazniti samo ob ponovnem zagonu storitve. `GangliaContext` in `GangliaContext31` sta namenjena periodičnemu pošiljanju metrik v odprtokoden nadzorni sistem Ganglia. Deluje tako, da na vsakem strežniku kreira nov proces imenovan `gmond`, ki skrbi za shranjevanje in posredovanje metrik. `JMX` ali javanska nadzorna razširitev (angl. Java Management Extensions)

in REST ali reprezentacijski statusni prenos (angl. Representational state transfer) sta novejša vtičnika, ki omogočata dostop do metrik s pomočjo namenskega odjemalca, ki preko oddaljenega klica procedure (angl. Remote Procedure Call - RPC) odgovori z določenimi metrikami. Težava je v številu podprtih metrik dosegljivih preko RPC-jev, saj vsi servisi niso podprti. Imenski strežnik in podatkovni strežnik implementirata tako imenovane M zrna (angl. Management Bean - MBean), ki omogočajo dostop do metrik preko RPCja. Sekundarni imenski strežnik pa ne vsebuje podpore za RPC.

Novejša različica metrik je na voljo od 0.20.203 verzije Hadoop-a in vsebuje nekaj sprememb. Ena izmed izboljšav je vidna preko REST vtičnika za pridobivanje metrik. Starejša verzija vtičnika je dosegljiva preko povezave "http://hadoop01.primer.si:50070/metrics", če predpostavimo, da je "hadoop01.primer.si" oznaka dodeljena strežniku, ki jo lahko sistem za domenska imena (angl. Domain Name System - DNS) preslika v IP naslov. Sistem, ki se odziva na ta IP naslov podpira RPC in ga želimo nadzirati. Z poizvedbo `format=json` lahko pridobimo enake podatke v JSON obliki, torej je dosegljiva na povezavi "http://hadoop01.primer.si:50070/metrics?format=json". V novejši različici pa je podpora za dostop do metrik na voljo na povezavi "http://hadoop01.primer.si:50070/jmx" vendar izključno v JSON formatu.

3.2 Apache Ambari

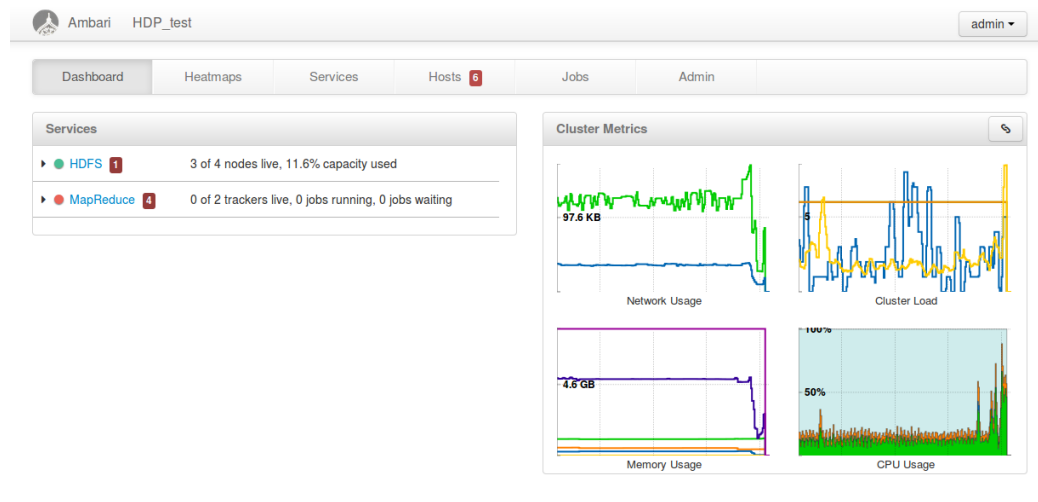
Problem pri velikih gručah računalniških sistemov predstavlja nadzor in upravljanje. Delovanje celotnega sistema je odvisno od posameznih strežnikov, ki za učinkovito delovanje potrebujejo redno vzdrževanje ter posodabljanje. Namestitvev posodobitev je v primeru platforme Hadoop zahtevno opravilo, saj sta ključni komponenti HDFS in MapReduce le predpogoj za delovanje ostalih distribuiranih storitev in, kot že prej omenjenih, projektov tesno povezanih s platformo. Na voljo so različna orodja, ki olajšajo proces namestitve, upravljanja in nadzora Hadoop gruče.

V želji po poenostavitvi postavitve, nadzora in upravljanja Hadoop gruče je nastal projekt Ambari, ki je, prav tako kot platforma Hadoop, izšel pod Apache v2 licenco. Ambari sistemskim administratorjem omogoča poenostavljeno postavitve Hadoop gruče preko spletnega vmesnika, ki vodi namestitve po korakih. Po končani namestitvi pa je preko istega vmesnika upravlja s servisi in ureja nastavitve ter jih uveljavi na celotni gruči. Trenutna verzija projekta Ambari (v času pisanja diplomske naloge je to verzija 1.2.4) podpira upravljanje nad servisi HDFS, MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig in Sqoop. Za nadzor uporabi in namesti že omenjen nadzorni sistem Ganglia za zbiranje in pregled nad sistemskimi metrikami in odptokodno orodje Nagios za obveščanje o kakovosti delovanja.

Orodje Ambari podpira več vrst namestitvev platforme Hadoop na gručo. Namestimo ga lahko na enega gostitelja, kar je primerno predvsem za testne namene, ali na ločene gostitelje. Namestitev na ločene gostitelje je priporočljivo uporabiti za bolj zahtevna testiranja in produkcijski sistem. Dokumentacija za pomoč pri uporabi orodja je na voljo na uradnih straneh [14] in vsebuje vse tehnične podrobnosti in priporočila za namestitve. Zaslonska slika orodja je prikazana na sliki 3.1

3.3 Cloudera

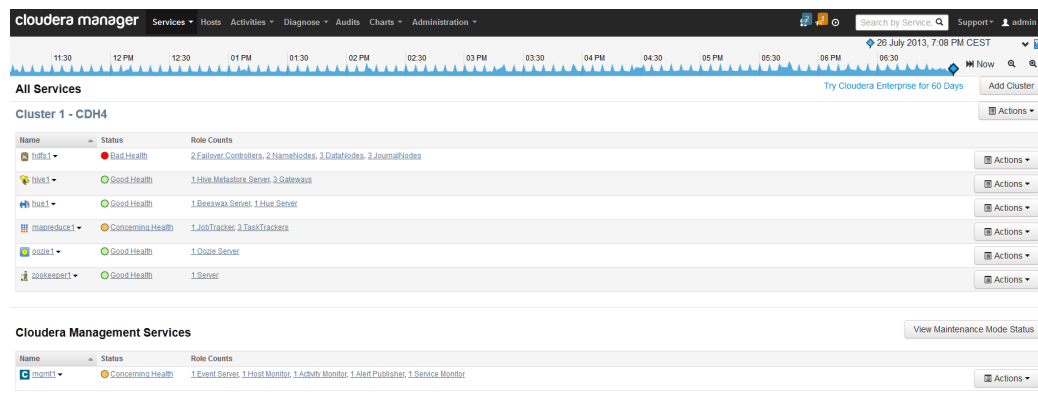
Cloudera je podjetje, ki ponuja celovito rešitev za nadzor ter upravljanje gruče Hadoop in odprtokodno distribucijo platforme Apache Hadoop imenovano CDH (angl. Cloudera's Distribution Including Hadoop). CDH distribucija vsebuje vse jedrne storitve platforme. Nazdornika, ki ju ponuja Cloudera sta imenovana Cloudera Standard, ki je brezplačen vendar z omejeno funkcionalnostjo in Cloudera Enterprise, ki je komercialni produkt s polno funkcionalnostjo in podporo. Cloudera Enterprise je na voljo tudi brezplačno za omejeno preizkusno dobo [16]. Oba nadzornika poskrbita za celotno namestitev, po namestitvi pa tudi za nadzor. Pri standardnem paketu je omejena funkcionalnost beleženja zgodovine dogodkov in stanja in konfiguracija



Licensed under the Apache License, Version 2.0.
See [third-party tools/resources that Ambari uses and their respective authors](#)

Slika 3.1: Projekt Ambari skrbi za konfiguracijo in upravljanje gruč Hadoop

naprednih projektov za Hadoop. Dostop do nadzornika (angl. Cloudera Manager) je urejen preko spletnega vmesnika, primer katerega je prikazan na sliki 3.2.



Slika 3.2: Spletni vmesnik nadzornika Cloudera Manager

3.4 Hortonworks

Hortonworks je podjetje ustanovljeno leta 2011 v Kaliforniji, Združenih Državah Amerike, s strani podjetja Yahoo in Benchmark Capital. Vzdržujejo lastno distribucijo platforme Hadoop in zanjo nudijo tudi podporo. Podobno kot pri Clouderi, tudi Hortonworks nudi odprtokodno in komercialno verzijo platforme Hadoop. Razlika med komercialno in odprtokodno rešitvijo je v podpori in funkcionalnostih, ki jih distribucija omogoča. Distribucija platforme se imenuje Hortonworks podatkovna platforma (angl. HDP - Hortonworks Data Platform), ki vsebuje tudi ostale Apache Hadoop projekte in dodaja nekaj lastnih orodij, ki olajšajo delo in nadzor. Poleg tega nudijo tudi podporo za implementacijo platforme Hadoop na operacijskem sistemu Microsoft Windows, ki pa je v času pisanja diplomske naloge še vedno v preizkusni fazi.

Dostop do orodja je, kot pri orodju Cloudera, urejen preko spletnega vmesnika, primer katerega je prikazan na sliki 3.1. Nadzornik pri distribuciji Hortonworks je tesno povezan s projektom Apache Ambari in je po izgledu identičen.

3.5 Splunk

Orodje za zajem, indeksiranje in korelacijo podatkov imenovano Splunk je napisano v programskem jeziku Python. Omogoča preureditev podatkov v iskalno zbirko. hitro iskanje, enostavno prepoznavo vzorcev, obveščanje o anomalijah v podatkih ter izris grafov. Dostop do orodja je mogoč preko spletnega vmesnika, ki omogoča celoten nadzor nad orodjem. Orodje omogoča iskanje po podatkih v realnem času, obveščanje in analiziranje podatkov ter gradnjo poročil. Deluje tako, da sprejema vse podatke, ki jih posredniki posredujejo ter jih označuje in indeksira. Z indeksiranjem pohitri kasnejše iskanje in omogoča gradnjo poročil ter spremljanje ali so parametri znotraj dovoljenih vrednosti, v realnem času.

Splunk je primeren za nadzor uporabniških storitev, nadzor spletnih ap-

likacijskih okolij, prepoznava profila uporabnikov pri komercialnih storitvah, alarmiranje ob zaznavi anomalij pri delovanju različnih informacijskih storitev, nadzor virtualnih sistemov ipd. Orodje omogoča tudi povezavo s platformo Hadoop in obdelavo podatkov shranjenih na datotečnem sistemu HDFS. Aplikacija za Splunk, ki omogoča integracijo med platformo Hadoop in orodjem Splunk se imenuje Hadoop konektor (angl. Hadoop Connect). Aplikacija omogoča uvoz podatkov v orodje kot tudi brskanje po datotečnem sistemu HDFS [12].

Orodje je licencirano z dvema različnima licencama. Podjetna licenca (angl. Enterprise License) je namenjena podjetjem, ki uporabljajo orodje za komercialne namene in indkesirajo velike količine podatkov, brezplačna licenca (angl. Free License) pa je namenjena osebno uporabo. Brezplačna licenca dovoljuje uporabo orodja, ki sprejme do 500MB podatkov dnevno, pri tem pa ni dovoljeno kvote preseči več kot tri krat v 30 dnevnem obdobju, poleg tega je tudi omejen glede določenih funkcionalnosti. Funkcionalnosti, ki so onemogočene v brezplačni licenci, so: avtenticiranje in uporaba vlog, porazdeljeno iskanje, posredovanje podatko v zunanje sisteme, nadzor nad distribuirano postavitvijo ter obveščanje in nadzor [10].

3.5.1 Splunk Forwarder

Splunk strežnik potrebuje način pridobivanja podatkov z nadzorovanih strežnikov. Z namenom posredovanja dnevniških datotek in ostalih podatkov so bili razviti posredniki (angl. Splunk forwarders). Posrednik je programska oprema, ki izvira iz orodja, katere primarni namen je označiti in varno posredovati podatke Splunk indeksirnemu strežniku. Razvitih je bilo nekaj instanc posrednikov, ki se razlikujejo po funkcionalnosti in načinu uporabe, saj je vrsta sistemov, ki jih orodje nadzira raznolika.

Splunk razvija tri vrste posredovalcev:

Universal forewarder je prilagojen Splunk servis, ki je od originalne verzije obdržal komponente, ki skrbijo za posredovanje podatkov.

Heavy forwarder je instanca Splunk servisa, ki ima onemogočene nepotrebne funkcionalnosti za posredovanje podatkov, z namenom zmanjšanja porabe pomnilnika.

Light forwarder je instanca Splunk servisa, ki ima omogočene ključne funkcionalnosti za posredovanje podatkov. Kljub imenu (light - lahek) posrednika je pri porabi sistemskih virov najbolj varčen univerzalni posrednik in posledično tudi najpogosteje uporabljen.

Težava pri uporabi Splunk posrednikov, ki so nameščeni na delovnih strežnikih je ravno nadzor in upravljanje konfiguracije le-teh. Splunk težavo reši z tako imenovano distribuirano postavitvijo (angl. distributed deployment), ki omogoča konfiguracijo porazdeljenih posrednikov s centralizacijo nastavitvev. Distribuirana postavitvev deluje na principu postavitvenega strežnika (angl. Deployment server) in postavitvenih odjemalcev (angl. Deployment Clients). Postavitveni strežnik, ki je običajno kar indeksni strežnik, se nahaja na omrežju in hrani konfiguracijo Splunk komponent, ki so prednastavljene kot postavitveni odjemalci. Postavitveni odjemalci periodično poizvedujejo postavitveni strežnik po konfiguraciji, ki jo v primeru sprememb ali posodobljene, tudi posreduje. Zaradi raznolikosti naprav, ki se nahajajo na omrežju, je običajno potrebnih več različnih konfiguracij. Konfiguracijo lahko ločimo v skupine, ki ločujejo naprave med seboj glede na ime strežnika, vrsto operacijskega sistema, vrsto strežnika, vrsto servisov, ki tečejo na strežniku, ipd. Uporaba postavitvenega strežnika torej poenostavi nadzor nad upravljanjem nastavitvev posrednikov, kar je priročno pri velikih sistemih, predvsem v primerih avtomatizacije postavitve gruče, kjer je posrednik že vgrajen v sliko operacijskega sistema in takoj po namestitvi le tega, posrednik samodejno pridobi ustrezno konfiguracijo.

3.6 Ostala uporabna orodja za nadzor

Preostala manj znana orodja za nadzor Hadoop gruče delujejo na podoben način kot Nagios, Ganglia ali Splunk. Večina teh orodij podatke zajema s

vmesnikov, ki so že implementirani v platformo Hadoop.

Primer orodja, ki za svoje delovanje uporablja protokol za enostaven nadzor omrežnih sistemov (angl. Simple Network Management Protocol - SNMP) je Cacti, ki ga lahko uporabimo za nadzor omrežne opreme. Protokol SNMP podpirajo omrežne naprave kot so stikala, tiskalniki, usmerjevalniki, IP telefoni, ipd. Primeren je pri nadzoru omrežne opreme, kot so usmerjevalniki in stikala, ki povezujejo gručo Hadoop.

Poglavje 4

Uporaba orodja Splunk za nadzor Hadoop gruče

Sledeči primer konfiguracije vsebuje konfiguracijo nadzornega orodja Splunk za potrebe nadzora gruče Hadoop je bil pripravljen za potrebe testiranja. Testiranje bo odločilo uporabno vrednost in primernost orodja kot produkcijski nadzorni sistem.

4.1 Lastnosti gruče

Za testne namene je bila postavljena majhna gruča, ki je bila sestavljena iz šestih virtualnih sistemov ter virtualne omrežne opreme. Uporaba virtualnih sistemskih virov za podlago platforme Hadoop je znan trend, ki je primerna rešitev za določene večuporabniške oblačne storitve. Primer uporabe virtualizacije je v večjem številu uporabnikov, ki potrebujejo določene sistemske vire in dodajanje le-teh po potrebi (angl. on demand).

Virtualizacija je proces simuliranja fizične strojne opreme s pomočjo hipervisorjev na način, ki posplošuje fizično stojno opremo do mere navidezne, na kateri lahko gradimo visoko nivojske storitve in procese. Uporaba virtualizacije omogoča dinamično določanje virtualnih sistemskih virov, uporaba že obstoječe strojne opreme, štetje procesorskega časa ter zniževanje oper-

Namen	CPU	RAM (GB)	Disk (GB)
Upravljalški strežnik	1	1.5	30
Imenski strežnik	1	1.5	30
Sekundarni imenski strežnik	1	1.5	30
Podatkovni strežnik	1	1.5	30
Splunk strežnik	2	2.0	80

Tabela 4.1: Porazdelitev sistemskih virov testne gruče. Naziv predstavlja namen uporabe virtualne instance, CPU predstavlja število procesorskih jeder namenjenih instanci, RAM predstavlja količino delovnega pomnilnika v GB in Disk predstavlja velikost virtualne diskovne enote v GB.

ativnih stroškov. Slabe lastnosti pri uporabi virtualizacije pa so ozka grla, procesorski čas je uporabljen za procese gostiteljskega operacijskega sistema in omejevanje performančnih možnosti fizičnega strežnika. Pri postavitvi platforme Hadoop, na virtualizirani strojni opremi, je potrebno upoštevati omejitve, kot so procesorski čas, število jeder in hitrost podatkovnega medija gostiteljskega sistema [9].

Razdelitev sistemskih virov je bila v času pisanja diplomske naloge, omejena in ne predstavlja primerne konfiguracije za produkcijske namene. Razdelitev sistemskih virov je prikazana v tabeli 4.1. Splunk strežniku je bilo v testni gruči dodeljenih večja količina sistemskih virov zaradi zahtevnih operacij, kot so iskanje, indeksiranje in shranjevanje zbranih podatkov.

Majhnost gruče ni predstavljal velikega bremena na omrežni opremi. Uporabljena je bila gigabitna (1Gbps) povezava med posameznimi virtualnimi instancami čeprav bi zadostovala že 100MB povezava. Za nadzor gruče so skrbeli trije sistemi, ki so opisani v nadaljevanju. Uporabljen je bil operacijski sistem je CentOS 6.4 Minimal, 64 bitna verzija z jedrom verzije 2.6.32.

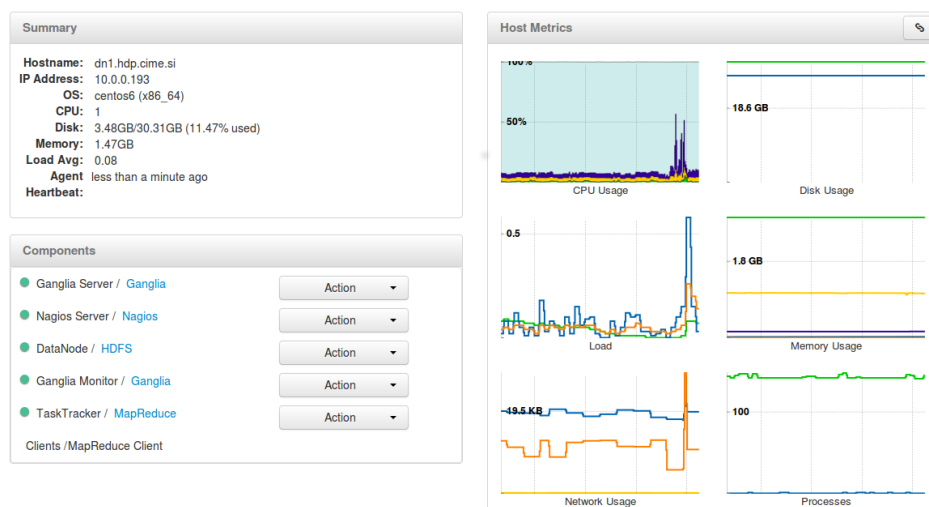
4.2 Upravljanje gruče s projektom Ambari

Začetna namestitev je bila opravljena s pomočjo orodja Ambari, kot del Hortonworks distribucije Hadoopa. Orodje omogoča začetno postavitve in konfiguracijo s pomočjo čarovnika, ki poskrbi za koordinacijo namestitve na posamezne strežnike. Preizkusili smo tudi Clouderino distribucijo platforme.

Predpriprava upraviteljskega strežnika zahteva namestitev podatkovne baze in paketa za sinhronizacijo systemske ure. Sledi namestitev orodja Ambari. Namestitev upraviteljskega orodja Ambari je avtomatizirana s paketno namestitvijo, ki vključuje tudi manjkajoče komponente (angl. Dependency Software). Testna gruča uporablja orodje Hortonworks Ambari verzije 1.2.4.9, ki je na voljo na lokalnem repozitoriju inkubatorja [13], ter uporablja programski jezik Java. Na strežniku je nameščena Java JDK verzije 6u21.

Namestitev programske opreme na delovne strežnike je v celoti avtomatizirana s strani projekta Ambari. Ročna predpriprava delovnih strežnikov skoraj ni potrebna, z izjemo omrežnih nastavitvev in dodatka za sinhronizacijo systemske ure. Sinhronizacija ure je pomembna zaradi časovno občutljivih zahtevkov in sinhronizacije med procesi. Celotna namestitev poteka preko spletnega vmesnika, katerega po postavitvi uporabimo za upravljanje in pregled stanja platforme servisov.

Namestitev poteka v nekaj korakih. Prvi korak je vnos imena gruče, sledi izbor servisnega sklada (angl. Service Stack) ali verzije HDP distribucije. V naslednjem koraku določimo na katere strežnike se bo namestila distribucija. Pri namestitvi velja omeniti, da je za pravilno delovanje potreben DNS (sistem za domenska imena, angl. Domain Name System) zapis za vsak strežnik, saj se pri naslavljanju strežnikov na aplikativnem nivoju uporabljajo strežniška imena in ne IP naslovi. Namestitev poteka s prijavo na strežnik z uporabo protokola SSH, preko katerega se namesti Ambari odjemalec (angl. ambari-client). Odjemalec sprejema ukaze namestitvenega strežnika. Orodje zahteva vnos zasebnega ključa certifikata s katerim se je mogoče prijaviti v strežnik preko protokola SSH. Po prenosu in namestitvi komponent, potrebnih za delovanje, lahko administrator določi kateri servisi bodo nameščeni na



Slika 4.1: Podrobnejši pregled delovanja delovnega strežnika v spletnem vmesniku Ambari

določene strežnike glede na lastne potrebe. Nekatere projekte je mogoče dodati tudi po opravljeni namestitvi. Orodje Ambari omogoča različne načine namestitve gruče in poskrbi za konfiguracijo posameznih servisov znotraj platforme, kot so imenski strežnik, sekundarni imenski strežnik, nadzornik nalog, ipd.

Zaslonski izris Projekta Ambari je prikazan na sliki 4.1, kjer je v pogledu v enega izmed delovnih strežnikov, vidna integracija z nadzornima sistemoma Nagios in Ganglia.

Uporaba projektov, kot je Ambari, pohitri proces namestitve jedrnih storitev Hadoop in poskrbi za vse podrobnosti pri namestitvi. V nasprotnem primeru je potrebna ročna namestitev vsake komponente platforme na strežnike, kar pa je lahko zelo zamudno opravilo.

4.3 Splunk

Konfiguracija orodja Splunk je zahtevnejša, saj je potrebno orodje prilagoditi za uporabo s Hadoop gručo. Splunk je takoj po namestitvi pripravljen sprejemati podatke. Analizo prejetih podatkov lahko uporabnik zažene tudi kas-

neje, ko je potreben podrobnejši pregled posamezne storitve ali procesa.

V testnem primeru je bila v orodju uporabljena aplikacija, ki omogoča prilagoditev pogleda uporabniškega vmesnika za potrebe nadzora Hadoop gruče.

Pomembna lastnost orodja Splunk je v prilagodljivosti pri konfiguraciji, saj lahko v že obstoječi infrastrukturi dodamo nove opazovalne parametre, spreminjamo sprejemljive meje in dodajamo poglede. Splunk sprejema podatke v celoti in jih ne prireja po shemi, ki kasneje omejujejo iskanje in obdelavo. Ta lastnost je pomembna pri iskanju vzroka neznane oz. nove težave v gruči, kasnejše preventivno ukrepanje in pomoč pri optimizaciji delovanja. Z orodjem lahko izluščimo podatke iz zapletenih ne strukturiranih oblik ter jih uredimo tako, da anomalije izstopajo in orodje nanje tudi pravočasno opozori. Primer iskalnega niza s pomočjo katerega pridobimo podatke o zasedenosti datotečnega sistema HDFS:

```
index=hadoopmon_metrics sourcetype="hadoop_dfsreport" "-----" |  
rex field=_raw "DFS Used: (?<dfsUsed>\d+)" | table _time dfsUsed
```

Iskalni niz vrne rezultat v obliki tabele ali grafa. Pogled lahko prilagodimo z ukazom `table` ali `chart` ali izberemo ustrezen prikaz pri iskanju. Grafični in tabelarni prikaz podatkov je prikazan na slikah 4.2 in 4.3, podpira tudi še nekaj drugih vrst vizualizacij podatkov.

4.3.1 Namestitev orodja Splunk

Na operacijskem sistemu CentOS 6.4 je namestitev orodja Splunk mogoča z enim ukazom. Orodje je potrebno prenesti z uradne spletne strani kjer se je, zaradi licenčnih pogojev, potrebno predhodno registrirati. Po prenosu se namestitev izvrši z ukazom:

```
rpm -i splunk-5.0.4-172409-linux-2.6-x86_64.rpm
```

Najenostavnejša postavitev, ki je izbrana privzeto, je namestitev indeksne in iskalne komponente na isti strežnik.

	_time ↕	dfsUsed ↕
81	9/11/13 9:53:59.000 AM	2113536
82	9/11/13 9:53:27.000 AM	2113536
83	9/11/13 9:52:57.000 AM	2113536
84	9/11/13 9:52:27.000 AM	2113536
85	9/11/13 9:51:58.000 AM	2113536
86	9/11/13 9:51:26.000 AM	2113536
87	9/11/13 9:50:57.000 AM	2113536
88	9/11/13 9:50:27.000 AM	2012871
89	9/11/13 9:49:57.000 AM	2012871
90	9/11/13 9:49:27.000 AM	2012871

Slika 4.2: Primer prikaza tabele v orodju Splunk



Slika 4.3: Primer prikaza grafa v orodju Splunk

Indeksiranje je pridobivanje informacij ter označevanje pridobljenih podatkov. Vsak vnos orodje sprejme in obdela (angl. parse), kot ločen dogodek ter preveri časovno oznako oz doda, v primeru da manjka, preveri oznake za novo vrstico, identificira gostiteljevo ime, izvor in tip izvora ter opsijsko zamaskira občutljive podatke. Podatki nato potujejo skozi indeksler, ki dogodke razdeli na segmente, po katerih je možno iskati, gradnja indeksne strukture ter shranjevanje podatkov in indeksa na podatkovni medij ter opsijsko posindeksno stiskanje. Indeksna komponenta (angl. Indexer) odpira dostop do spletnega vmesnika, ki je dosegljiv na prednastavljenih vratih. Privzeto je številka vrat nastavljena na vrata 8000.

Komponenta za iskanje (angl. Search head) poskrbi za izvedbo iskalne poizvedbe, združevanje pridobljenih rezultatov in prikaz le-teh uporabniku.

Na nadzorni strežnik je bil nameščen Splunk verzije 5.0.4 za linux 2.6, prav tako v 64-bitni verziji. Za nazoren pregled gruče Hadoop je bila v orodje Splunk nameščena aplikacija Splunk App for HadoopOps, ki je na voljo za brezplačen prenos z uradne spletne strani [11]. Aplikacija že vsebuje prilagojene poglede in osnovne iskalne nize, ki jih je potrebno ustrezno prilagoditi za vsako distribucijo platforme.

Aplikacija doda pogled spletnega vmesnika (angl. Splunk WebUI) za potrebe nadzora gruče in iskalne nize, ki so potrebni za delovanje tega pogleda. Namesti tudi distribucijsko namestitev in privzeto konfiguracijo posrednika za distribucijsko namestitev. Na nadzorni strežnik je bila nameščena aplikacija Splunk App for HadoopOps verzije 1.1.2. Administrator lahko tudi sam prilagodi privzeti vmesnik za pogled nad stanjem Hadoop gruče.

4.3.2 Namestitev Splunk posrednikov

Postopek namestitve je vseboval namestitev in konfiguracijo Splunk posrednikov (angl. Splunk forwarder). V tesni gruči je na vsak delovni in imenski strežnik nameščen Splunk posrednik. Uporabljeni so bili univerzalni posredniki (angl. Universal forwarder), na katere je potrebno namestiti prilagojeno aplikacijo, ki vsebuje izvršljive datoteke za nadzor delovanja strežnika. Na

delovne in imenske strežnike je bil nameščen posrednik splunkforwarder-5.0.4 za linux 2.6, 64-bitna verzija. Uporabi težkega in lahkega posrednika smo se izognili zaradi večje porabe sistemskih virov.

Namestitev samega posrednika na strežnik je poenostavljena s paketno namestitvijo ali, kot že prej omenjeno vgradnjo v sliko operacijskega sistema. Namestitev poskrbi za prenos ustrezne programske opreme, ki je potrebna za delovanje orodja.

Na posrednike je potrebno namestiti komponento za pregled gruče Splunk TA for HadoopOps. Nameščena je bila verzija 1.1.1. Podobno kot nadzorni strežnik Splunk je tudi za posrednike potreben prenos z uradne spletne strani in sprejem licenčnih pogojev. Namestitev je prav tako sprožena z ukazom:

```
rpm -i splunkforwarder-5.0.4-172409-linux-2.6-x86_64.rpm
```

Zagon Splunk orodij po zagonu operacijskega sistema se vključi z ukazom:

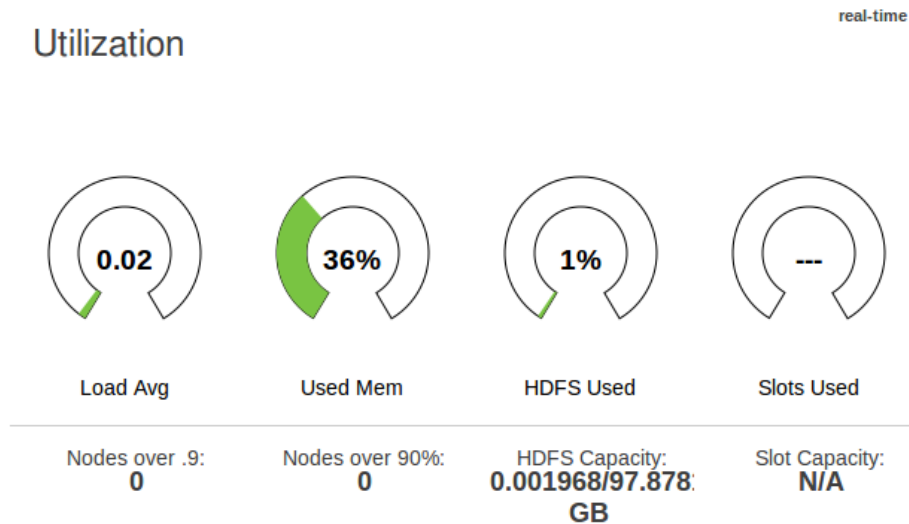
```
/opt/splunk/bin/splunk enable boot-start
```

4.3.3 Konfiguracija Splunk nadzornika

Na nadzornik Splunk je bila nameščena aplikacija Splunk App for HadoopOps verzije 1.1.2. Namestitev vsebuje privzete nastavitve za pregled, ki jih je potrebno še prilagoditi okolju, ki ga nadzira. Na voljo so privzete nastavitve za omenjeni distribuciji Hortonworks in Cludera.

Testa gruča je vsebovala namestitev iz Hortonworksovega repozitorija. Privzete nastavitve niso delovale. Urediti je bilo potrebno imena parametrov in metrik v nekaterih iskalnih nizih. Iskalni niz, ki vrne celotno kapaciteto delovnih strežnikov smo zamenjali z nizom:

```
index=hadoopmon_metrics sourcetype="hadoop_dfsreport" "-----" |  
head 10 | rex field=_raw "Configured Capacity: (?<confCap>\d+)"  
| table _time confCap
```

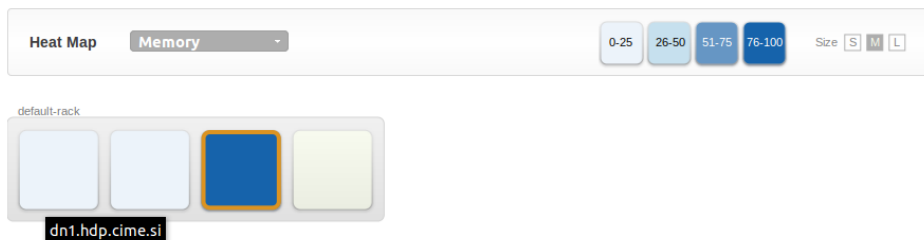


Slika 4.4: Pregled porabe sistemskih virov gruče Hadoop v orodju Splunk

Po ureditvi iskalnih nizov je aplikacija prikazovala veljavne vrednosti stanja gruče. Primer pregleda zasedenost pomnilnika, datotečnega sistema in obremenjenost procesnih enot je prikazan na sliki 4.4.

Splunk vsebuje celovit pregled delovanja gruče z grafičnim prikazom kvalitete delovanja posameznih strežnikov. Vmesnik omogoča izbiro pogleda v posameznih dimenzijah kot so zasedenost delovnega pomnilnika, obremenjenost procesne enote, zasedenost diska. Tak pregled omogoča administratorjem pregled nas delovanjem celotne gruče in tako lahko hitreje najdejo strežnik, ki zaradi pomanjkanja posameznih sistemskih virov povzroča napake pri izvajanju nalog in tako omeji porabo virov za ta strežnik. Na slikah je prikazan pregled strežnikov v nadzornikih Splunk, Ambari in Nagios.

Prikaz delovanja strežnikov v orodju Nagios vsebuje več podatkov o delovanju posameznega strežnika vendar ne omogoča odkrivanja vzroka za morebitno težavo ali napako. Orodje Ambari omogoča pregled zasedenosti diskovnih enot ter obremenjenost centralne procesne enote, medtem ko ne prikazuje zasedenost pomnilnika in je za to urejena preusmeritev na orodje Ganglia. Ambari omogoča podrobnejši pregled delovanja posameznega stre-



Slika 4.5: Pregled delovanja strežnikov v orodju Splunk

Dashboard Heatmaps Services Hosts Jobs Admin						
All (4) ● Healthy (3) ● Master Down (0) ● Slave Down (1) ● No Heartbeat (0) ! Alerts (1) + Add New Hosts						
Name	IP Address	CPU	RAM	Disk Usage	Load Avg	Components
● dn1.hdp.cime.si ...	10.0.0.193	1	1.47GB	<div style="width: 100%;"></div>	0.76	DN, GM and 4 more
● dn2.hdp.cime.si ...	10.0.0.194	1	996.6MB	<div style="width: 100%;"></div>	0.89	DN, GM and 3 more
● nn.hdp.cime.si ... !	10.0.0.191	1	1.47GB	<div style="width: 100%;"></div>	0.32	DN, GM and 2 more
● sn.hdp.cime.si ...	10.0.0.192	1	1.47GB	<div style="width: 100%;"></div>	0.36	DN, GM and 1 more

Slika 4.6: Pregled delovanja strežnikov v orodju Ambari

Service Overview For All Host Groups

agent-servers (agent-servers)				All Servers (all-servers)				ganglia-server (ganglia-server)			
Host	Status	Services	Actions	Host	Status	Services	Actions	Host	Status	Services	Actions
dn1.hdp.cime.si	UP	11 OK 1 WARNING	[Actions]	dn1.hdp.cime.si	UP	11 OK 1 WARNING	[Actions]	dn1.hdp.cime.si	UP	11 OK 1 WARNING	[Actions]
dn2.hdp.cime.si	UP	4 OK	[Actions]	dn2.hdp.cime.si	UP	4 OK	[Actions]				
nn.hdp.cime.si	UP	14 OK 1 CRITICAL	[Actions]	nn.hdp.cime.si	UP	14 OK 1 CRITICAL	[Actions]				
sn.hdp.cime.si	UP	3 OK	[Actions]	sn.hdp.cime.si	UP	3 OK	[Actions]				

jobtracker (jobtracker)				nagios-server (nagios-server)				namenode (namenode)			
Host	Status	Services	Actions	Host	Status	Services	Actions	Host	Status	Services	Actions
nn.hdp.cime.si	UP	14 OK 1 CRITICAL	[Actions]	dn1.hdp.cime.si	UP	11 OK 1 WARNING	[Actions]	nn.hdp.cime.si	UP	14 OK 1 CRITICAL	[Actions]

region-servers (region-servers)				slaves (slaves)				tasktracker-servers (tasktracker-servers)			
Host	Status	Services	Actions	Host	Status	Services	Actions	Host	Status	Services	Actions
dn1.hdp.cime.si	UP	11 OK 1 WARNING	[Actions]	dn1.hdp.cime.si	UP	11 OK 1 WARNING	[Actions]	dn1.hdp.cime.si	UP	11 OK 1 WARNING	[Actions]
dn2.hdp.cime.si	UP	4 OK	[Actions]	dn2.hdp.cime.si	UP	4 OK	[Actions]	dn2.hdp.cime.si	UP	4 OK	[Actions]
nn.hdp.cime.si	UP	14 OK 1 CRITICAL	[Actions]	nn.hdp.cime.si	UP	14 OK 1 CRITICAL	[Actions]				
sn.hdp.cime.si	UP	3 OK	[Actions]	sn.hdp.cime.si	UP	3 OK	[Actions]				

Slika 4.7: Pregled delovanja strežnikov v orodju Nagios

žnika, kot je prikazano na sliki 4.1. Orodje Splunk za razliko od orodja Ambari in Nagios prikazuje samo eno dimenzijo sistemskih virov naenkrat, kar lahko izbiramo z izbirnim menijem. S klikom na posamezni strežnik pa odpremo podrobnejši pregled delovanja.

4.4 Obveščanje

Iskalni nize lahko uporabnik shrani za kasnejše ogled ali prilagodi svojim potrebam. V primeru, da iskanje prikaže podatke pomembnih parametrov, lahko ta niz shranimo in uredimo obveščanje. Obveščanje v orodju Splunk nam omogoča, da uporabnik določi iskalni niz, ki vrne končen rezultat, nato določimo dopustne meje le-tega. Pri nastavitvi opozorila (angl. Alert) lahko določimo urnik izvedbe iskalnega niza in tako določimo pogostost obveščanja. Izbiramo lahko med:

Sprožitev v realnem času , ki sproži opozorilo na vsako ujemanje rezultata,

Periodična izvedba iskalnega niza , ki določi urnik, kdaj se bo izvedel iskalni niz,

Nadziraj skozi drseče okno , ki nastavi časovni okvir, znotraj katerega se izvaja iskalni niz.

Izbira pogostosti obveščanja je odvisna od vrste iskalnega niza. Primer uporabe sprožitev v realnem času je v poizvedbah, ki vrnejo izjemne dogodke, kot so izpad strežnika, vdor v sistem, ipd. Iskalni niz, ki se preverja periodično je primeren pri zaznavi sprememb v delovanju. Nadzor znotraj drsečega okna je primeren za iskalne nize, ki so občutljivi na anomalije v določenem časovnem okviru. Dober primer je zaznava DOS (angl. Denial Of Service - preprečevanje storitev) napada na strežnik, kjer z iskalnim nizom spremljamo število zahtevkov za spletno stran in v primeru DOS napada, pri katerem se močno poveča število zahtevkov, znotraj prednastavljenega časovnega okvira in sproži opozorilo.

V domeni datotečnega sistema HDFS, je običajno potrebno administratorja obvestiti, ko HDFS doseže določeno stopnjo zasedenosti, saj je v takem primeru potrebna nadgradnja gruče ali ukrep, ki zmanjša porabo prostora. Primer iskalnega niza, ki vrne trenutno zasedenost datotečnega sistema HDFS v procentih je:

```
index=hadoopmon_metrics sourcetype="hadoop_dfsreport" "-----" |  
head 1 | rex field=_raw "DFS Used%: (?<dfsUsedPct>\d+)" |  
eval dfsUsedPct = if( dfsUsedPct < 90, "", dfsUsedPct ) |  
table dfsUsedPct
```

Za ta iskalni niz je primerna sprožitev v realnem času ali periodična izvedba. Iskalni niz izbere en podatek iz metrik, ki so tipa poročil o datotečnem sistemu (parameter `hadoop_dfsreport`) ter izlušči ven odstotno vrednost zasedenosti datotečnega sistema s pomočjo regularnega izraza. Pogojni stavek izpiše odstotno vrednost v primeru, da je le-ta večja od 90 odstotkov ter rezultat tabelira. V primeru, da tabela ni prazna (zasedenost datotečnega sistema je večja od 90%) se bo sprožilo opozorilo.

Akcijo opozorila nastavi administrator. Opozorilo lahko sestavljajo obvestilo v obliki elektronske pošte ali zagon poljubne skripte, administrator lahko nastavi tudi pomembnost opozorila, kar vpliva na razporeditev obvestil na seznam.

4.5 Splunk kot nadzornik gruče Hadoop

Privzeta orodja za nadzor Hadoop gruče, ki so dodana v distribucijo komercialnega ponudnika platforme, so lahko omejena po funkcionalnosti in uporabnosti. Prednost takšnih orodij je v nastavitvah, za katere je poskrbljeno s strani distribucije in ponudnika, administrator jo lahko začne uporabljati takoj po namestitvi (angl. *out of the box*). Prednastavljena orodja za nadzor močno skrajšajo čas pri postavitvi sistema in so nastavljena tako, da prikazujejo vitalne metrike. Nadzorna orodja je seveda mogoče tudi kasneje nadgraditi in prilagoditi lastnim potrebam.

V primeru uporabe orodja Ambari, lahko med postopkom namestitve administrator izbere katera nadzorna orodja bodo nameščena. Na voljo sta orodja Nagios in Ganglia. Dobra lastnost orodja Ambari je enostavnost uporabe. Tudi uporabniki, ki nimajo izkušenj o delovanju platforme Hadoop, lahko v nekaj minutah končajo celotno namestitev, saj podrobnosti pri namestitvi uredi Ambari. Slabost pri uporabi orodja pa je v slabi prilagodljivosti, saj pri namestitvi ne omogoča zahtevnejše konfiguracije. Velja omeniti še, da je za majhne in srednje velike gruče konfiguracija, ki jo omogoča orodje Ambari, dovolj obsežna. Omenjena nadzorna sistema sta povezana s projektom Ambari, za delovanje Hadoop storitev pa nista nujno potrebna.

Ganglia je prilagojena za nadzor porabe sistemskih virov v računalniških gručah. Porabo sistemskih virov je pri nadzoru kompleksnega sistema kot je Hadoop potrebno nadzirati, ampak za napreden nadzor potrebujemo več informacij. Ganglia ne omogoča celovite rešitve za iskanje vzroka težave v delovanju sistema. Ker je Hadoop platforma, ki razpolaga s sistemskimi viri, se lahko pojavi nihajoč vzorec v porabi delovnega pomnilnika, ki pa je značilen za tip naloge, ki se v tistem trenutku izvaja na gruči.

Nagios je orodje za nadzor storitev in sistemov. Omogoča obveščanje v primeru izpada in planiranje posodobitev ali vzdrževalnih del. Hadoop omogoča dostop do metrik o delovanju preko spletnega vmesnika, kar s pridom izkorišča Nagios. Orodje ne beleži zgodovine o delovanju, ampak se v večini primerov uporablja za pregled trenutnega stanja gruče in storitev. Prednost pri uporabi orodja je pregleden vmesnik, ki nazorno prikazuje trenutno stanje, primer prikaza je viden na sliki 4.7. Tudi Nagios ne omogoča vpogleda v metrike posamezne naloge. Ker je orodje enostavno za uporabo in ne porabi veliko sistemskih virov, je primerno za uporabo pri nadzoru jedrnih storitev platforme Hadoop.

Pri Clouderini distribuciji Hadoopa je nadzornik integriran v samega upravljalca distribucije, kar pomeni, da so vse spremembe v konfiguraciji gruče takoj vidne v nadzornem sistemu. Slaba lastnost omenjenega nadzornika je v načinu shranjevanja podatkov, saj za delovanje potrebuje funkcionalen HDFS

datotečni sistem. To pomeni, da v primeru izpada celotne gruče sistem ne bo dostavil obvestil o napaki. Da se takim primerom izognemo, je mogoče nadzornik ločiti od primarne delovne gruče, za kar pa je potrebna dodatna strojna oprema.

V implementaciji testne gruče je bil uporabljen Ambari, s pomočjo katerega je bila nameščena HDP distribucija platforme. Namestitev je vsebovala tudi privzeto konfiguracijo nadzornih sistemov Ganglia in Nagios. Dodana je bila tudi vsa programska oprema, potrebna za delovanje nadzornika Splunk. Preizkusili smo tudi Cloudera Manager.

Splunk je izmed vseh preizkušenih orodij predstavljal najboljšo rešitev za nadzor Hadoop gruče. Orodje se uporablja preko spletnega vmesnika. Alarmiranje in proženje obvestil je v celoti nastavljivo. Problem pri uporabi orodja Splunk je v količini podatkov, ki jih generirajo posredniki, saj lahko že majhna gruča preseže kvoto 500MB podatkov na dan, kar predstavlja omejitev brezplačna licence. Zahtevna začetna konfiguracija orodja je verjetno razlog, zakaj administratorji raje posežejo po integriranih orodjih. Cloudera Manager je eno izmed najbolj izpopolnjenih orodij za nadzor Hadoop gruče, toda je vezana na lastno distribucijo platforme. Splunk te omejitve nima, toda je potrebna dodatna kontrola iskalnih nizov, saj lahko ob posodobitvi platforme pride do menjave metrik, kar pa povzroči napake v delovanju nadzorovanih storitev.

Vsi omenjeni nadzorni sistemi omogočajo ločevanje od nadzorovanega sistema, kot to omogoča Cloudera in za samo delovanje gruče niso nujno potrebni.

Poglavje 5

Sklepne ugotovitve

V diplomski nalogi smo spoznali delovanje dveh osnovnih komponent Hadoopa: HDFS in MapReduce. Spoznali smo, da je za vzdrževanje porazdeljenega sistema, ki ga sestavljajo osnovne računalniške komponente, potreben napreden nadzor. Uporaba več nadzornih sistemov hkrati je ključna za napreden nadzor Hadoop gruče. Nadzorni sistemi, ki so vgrajeni v posamezno distribucijo Hadoop, so primerni za uporabo v produkcijskih okoljih, vendar slabša prilagodljivost onemogoča dodatne poglede, ki bi lahko pripomogli k optimizaciji delovanja. Za majhne in srednje velike gruče so preventivni ukrepi bolj pomembni, saj je delež gruče, ki jo predstavlja določena računalniška komponenta višja, kot pri večjih gručah. Zgodna napoved izpada je ključnega pomena pri zagotavljanju kakovosti delovanja. Čeprav je arhitektura platforme Hadoop že prilagojena za delovanje na osnovni računalniški opremi, je faktor tveganja izgube komponent mogoče znižati. Splunk se je izkazal za uporabno orodje pri nadzoru gruče toda sama kompleksnost postavitve in optimizacija delovanja ni primerna za majhne gruče. Izbira orodja za napreden nadzor je odvisna od kompleksnosti porazdeljenega sistema in znanja ljudi, ki bodo ta orodja uporabljali.

Testna gruča s porazdeljenimi sistemskimi viri ni predstavljala primerne konfiguracije za produkcijsko okolje. Gruča šestih računalnikov ne predstavlja reprezentativne konfiguracije, toda dokazuje zmožnost delovanja platforme

z omejenimi viri, čeprav virtualnimi. MapReduce naloge, ki so se izvajale na platformi, so bile zgolj testne in niso predstavljale velikega bremena na sistemu.

Cilj diplomske naloge je bil preizkusiti nekatere napredne rešitve nadzora Hadoop gruče. Predstavljena vgrajena orodja za uporabo v majhnih do srednje velikih gručah, povsem zadostujejo potrebam po nadzoru. Za zelo specifične primere je orodje Splunk v veliko pomoč, čeprav je mogoč vpogled v dnevniške datoteke tudi preko spletnih vmesnikov platforme.

Naslednji korak je izboljšava Splunk aplikacije za nadzor platforme Hadoop in razvoj vtičnika za integracijo metrik z orodjem Splunk.

Literatura

- [1] Apache Licence v2.0, 2013. Povzeto po <http://www.apache.org/licenses/LICENSE-2.0.html> dne 13.9.2012.
- [2] P. Zadronzny, R. Kodali. *Big Data Analytics Using Splunk*. Apress, 2013
- [3] S. Ghemawat, H. Gobioff, and S. Leung *The Google File System*. Google, 2003.
- [4] Hadoop Metrics, 2009. Povzeto po <http://blog.cloudera.com/blog/2009/03/hadoop-metrics/> dne 13.9.2012.
- [5] E Sammer. *Hadoop Operations*. Oreilly, 2012.
- [6] T White. *Hadoop: The Definitive Guide, Third Edition*. Oreilly, 2012.
- [7] HDP product dependencies and licenses. Povzeto po <http://hortonworks.com/hortonworks-data-platform-product-dependencies-and-licenses/> dne 13.9.2012.
- [8] J. Dean and S. Ghemawat *MapReduce: Simplified Data Processing on Large Clusters*. Google, Inc., 2004.
- [9] Virtual Hadoop. Povzeto po <http://wiki.apache.org/hadoop/Virtual%20Hadoop> dne 13.9.2012.
- [10] Types of Splunk licenses. Povzeto po <http://docs.splunk.com/Documentation/Splunk/5.0.4/Admin/TypesofSplunklicenses> dne 13.9.2013.

- [11] Splunk App for HadoopOps. <http://www.splunk.com/view/hadoopops/SP-CAAHA2> dne 13.9.2013.
- [12] Splunk Hadoop Connect. <http://www.splunk.com/view/hadoop-connect/SP-CAAHA3> dne 13.9.2013
- [13] Ambari Repository. <http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.2.4.9/ambari.repo>, 20.8.2013.
- [14] Ambari dokumentacija. <http://incubator.apache.org/ambari/1.2.4/installing-hadoop-using-ambari/content/index.html>, 20.8.2013.
- [15] Hadoop Core. Povzeto po <http://hadoop.apache.org/core/> dne 13.9.2013.
- [16] Cloudera Manager. Povzeto po <http://www.cloudera.com/content/cloudera/en/products/cloudera-manager.html> dne 13.9.2013.