UNIVERSITETI I TIRANËS
FAKULTETI I EKONOMISË
DEPARTAMENTI I STATISTIKE DHE INFORMATIKE E ZBATUAR

Temë Diplome

“Sistemi Quasar:Menaxhimi efiçent i burimeve në clustera”

Diplomë e ciklit të parë të studimit
”BACHELOR”

Punoi:
Ledio Bega
IE 304

Udhëheqës:
Prof. DR. Besa Shahini

Tiranë, Tetor, 2018

Dedikime dhe falenderime:

Në radhë të parë falenderoj familjen time,e cila më ka mbështetur shumë gjatë këtyre viteve, dhe pa të cilët nuk do të kasha arritur asgjë nga ato që kam arritur deri më sot.

Gjithashtu falenderoj gjithë miqtë e mi,të cilët më janë gjendur gjithmonë pranë si në momente të mira ashtu edhe në ato të vështira.

Së fundmi falenderoj gjithë stafin e pedagogëve të Fakultetit Ekonomik që me përkushtimin më të madh dhe dëshirën e pashtershme për të na mësuar na kanë ndjekur këto vite dhe vecanërisht pedagogen udhëheqëse Prof.Dr.Besa Shahini për mbështetjen,udhëzimet,këshillat,të cilat më ndihmuan shumë që të realizoja punimin me sukses.

Tabela e përmbajtjes

Tabela e Figurave

Abstrakti

Shoqëria njerëzore ka pësuar një ndryshim të jashtëzakonshëm në fushën e teknologjisë dhe kompjuterave,që nga koha kur ata u krijuan,e deri në ditët e sotme.Fillimisht,kompjutera shumë të mëdhenj vendoseshin në dhoma të vecanta dhe vetëm profesionistët ishin të aftë t'i përdornin.Më pas,me zhvillimin teknologjik,erdhi koncepti i Cloud Computing.

Në fushën e teknologjisë së informacionit,Cloud Computing shfaqet si një nga cështjet më të kërkuara.Cloud computing është termi që i referohet ndarjes së burimeve kompjuterike dhe shërbimeve të niveleve të larta që mund të menaxhohet lehtësisht. Por sot cloud computing po përballet me disa vështirësi të konsiderueshme.

Duke qenë një mënyrë kosto efiçente përdoruesit i kanë braktisur sistemet e mëparshme dhe i jane drejtuar cloud-it, kështu harxhojnë më pak kohë për të mirëmbajtur serverat e kompanive dhe kanë gjetur si opsion ngarkimin e datacenterave. Kështu që lind nevoja për të ndërtuar më shume datacenter-a për shkak të numrit të përdorueve në rritje, por ky do të ishte një investim që do të kërkonte mbase miliona dollarë, prandaj zgjidhja është që të kërkojmë mbështetje tek proceset teknologjike që realizojnë një performancë më të lartë. Kështu,në këto kushte cili do të ishte konfigurimi më i mirë që një kompani do t'i bënte infrastrukturës së saj? Zgjidhja është e thjeshtë:Clustering. Ky term i referohet konfigurimit të rrjetit të kompjuterave në një mënyrë të tillë që të funksionojnë si një kompjuter i vetëm.

Për rrjedhim qëllimi i këtij punimi është njohja me sisteme që ndihmojnë në rritjen e efiçencës së ruajtjes së të dhënave në Clustera,më konkretisht sistemi Quasar.Punimi është i ndarë në katër pjesë: pjesa e parë flet për Clustera-t, jep përshkrim të shkurtër të tyre,shpërndarja e burimeve,përcaktimi i burimeve,mënyrat e menaxhimit.

Kapitulli i dytë është menaxhimi i koordinuar i clusterave,i cili flet për mangësitë që ka sistemi aktual i Clusterave dhe menaxhimi i koordinuar i tyre.

Kapitulli i tretë është pjesa praktike në këtë punim,ku është krijuar një cluster virtual me dy nyje në platformën Windows Server.

Kapitulli i katërt na njeh me sistemin Quasar,njohje me këtë sistem menaxhimi,klasifikimin e tij,shpërndarja dhe përcaktimi.

Kapitulli i fundit bën një krahasim të quasar me sisteme të tjera të ngjashme,dhe si përfundim jepen konkluzionet e këtij punimi

Hyrje

Në fushën e teknologjisë së informacionit,Cloud Computing shfaqet si një nga cështjet më të kërkuara. Cloud computing është termi që i referohet ndarjes së burimeve kompjuterike dhe shërbimeve të niveleve të larta që mund të menaxhohet lehtësisht.

Cloud Computing mund të përkufizohet gjithashtu si grupi i elementëve të një rrjeti që ofrojnë shërbime që nuk adresohen individualisht ose menaxhohen nga përdoruesit,por në vend të kësaj,i gjithë ky rrjet hardwaresh dhe softwaresh mund të mendohet si një re(cloud). Qëllimi i Cloud Computing është që përdoruesit të përfitojnë nga këto teknologji,pa patur nevojën për njohuri të thella ose ekspertizë në ndonjë nga teknologjitë. Cloudi synon të ulë kostot duke ndarë burimet dhe të arrijë ekonomi shkalle.

Por sot cloud computing po përballet me disa vështirësi të konsiderueshme. Duke qenë një mënyrë costo efiçente përdoruesit i kanë braktisur sistemet e mëparshme dhe i jane drejtuar cloud-it, kështu harxhojnë më pak në mirëmbajtjen e serverave personal të kompanive dhe kanë ngarkuar datacenterat. Prandaj që lind nevoja për të ndërtuar më shume datacenter-a por ky do të ishte një investim qindra miliona dollarësh; na ngelet të kërkojmë mbështetje tek proceset teknologjike që garantojnë një performancë më të lartë.

Për rrjedhim paraqes njohjen me procedura dhe sisteme që ndihmojnë në rritjen e efiçencës së menaxhimit të ruajtjes së këtyre të dhënave, dhe cila është zgjidhja më e mirë që garanton performancë pa cënuar QoS.

Për tu njohur më shumë me këtë temë na duhet të bëhemi familjarë me disa terma si mëposhtë:

Cluster kompjuter – konsiston në një set kompjuterash të lidhur ngushtësisht së bashku dhe që punojnë po së bashku për një qëllim të vetëm.

Një framework – është një abstrakt ku softwaret sigurojnë një funksionalitet gjenerik. Funksionalitetet mund të fitohen duke shtuar kodin e vetë përdoruesit dhe kështu kemi një sistem/aplikacion specifik.

Scale out – shtimi i më shumë komponentëve në paralel për të përballuar një ngarkesë

Scale up – rritja e parametrave të burimit për të përballuar një ngarkesë.

Metodologjia

Së pari kërkimet gjatë kësaj pune janë realizuar duke mbledhur dhe analizuar faktet dhe nxjerrjen e rezultateve në bazë të tyre. Krahas një trajtimi të gjerë teorik te clustera-ve dhe sistemit Quasar,në vijim do të shpjegohet metodologjia e përdorur për mbledhjen dhe analizimin e të dhenave.

Së dyti,studimi i clustera-ve dhe Quasar sjell nevojën e përdorimit të një qasjeje krahasuese. Për të kuptuar më mirë si funksionon realisht sistemi Quasar, për të parë performancën e tij në situata reale dhe për ta krahasuar me sisteme të tjera, është parë e nevojshme që t'i drejtohem literaturës si ndërkombëtare dhe kombëtare,duke qenë se nuk ka patur informacion të bollshëm në lidhje me këtë temë.

Gjithashtu, duke qenë se ky studim ka edhe qëllim krijimin e një clusteri virtual, pra pjesën praktike, është parë e nevojshme përdorimi i platformës Windows Serever,për të realizuar me sukses këtë proces.

Përfundimi me sukses i këtij punimi nuk ka qenë i lehtë, duke qenë se ka patur vështirësitë e veta, për një sërë arsyesh të lartpërmendura.Por ajo që është e rëndësishme është që kjo temë është interesante dhe aktuale, dhe gjithashtu në një nga fushat më të paguara.

1. Hyrje Në Sistemet Cluster,Menaxhimi i Tyre

Pjesët përbërëse të Cluster-it janë të lidhura nëpërmjet LAN-eve të shpejta,dhe cdo nyje ekzekuton një instancë të OS,pra një Cluster mund të shikohet si një sistem i vetëm kompjuterik.Në shumicën e rasteve cdo nyje përdor të njëjtin hardëare,dhe të njëjtin sistem shfrytëzimi.

1.1.Pse u krijuan Clusterat?

Clusterat zakonisht janë zhvilluar për të përmirësuar performancën dhe disponueshmërinë e kompjuterave.Clusterat u krijuan si pasojë e nevojës në rritje për mikroprocesorët me kosto të ulët,dhe rrjetat me shpejtësi të lartë.

1.2.Menaxhimi i Clusterave

Menaxhimi i një strukture cluster përfshin një tërësi shërbimesh ku mund të përmendim sigurinë, tolerancën ndaj gabimit monitorimin.Menaxhimi kryhet nga një Cluster Manager,që zakonisht është një backend GUI,ose një softëare i tipit command-line.Por,përpara se të fillojmë me procesin e menaxhimit,do të shpjegojmë shkurtimisht se përse duhet menaxhimi i Clustera-ve dhe cfarë benefitesh na sjell ai.Cdo biznes i madh sot në botë ka nevojë të ketë një infrastrukturë të rrjetit në një mënyrë të tillë që të minimizojë ose eliminojë nyjet e rrjetit që mund të dështojnë ose janë problematike,sepse nëse ato do të dështonin,do të ndalonin operacionet kritike të kompanisë duke sjellë humbje të mëdha.Ajo që kompanitë bëjnë është që i vendosin sistemet e tyre të prodhimit në një konfigurim clusteri.Kjo do të thotë rritje e numrit të pajisjeve hardëare dhe softëare ne mënyrë që në rast se një pajisje primare dështon,operacionet mund të transferohen për t'u përpunuar në një pajisje sekondare. Ky punim do të fokusohet në dy temat më kryesore për eficencën e burimeve: shpërndarja dhe përcaktimi i burimeve të ngarkesave me të dhëna.

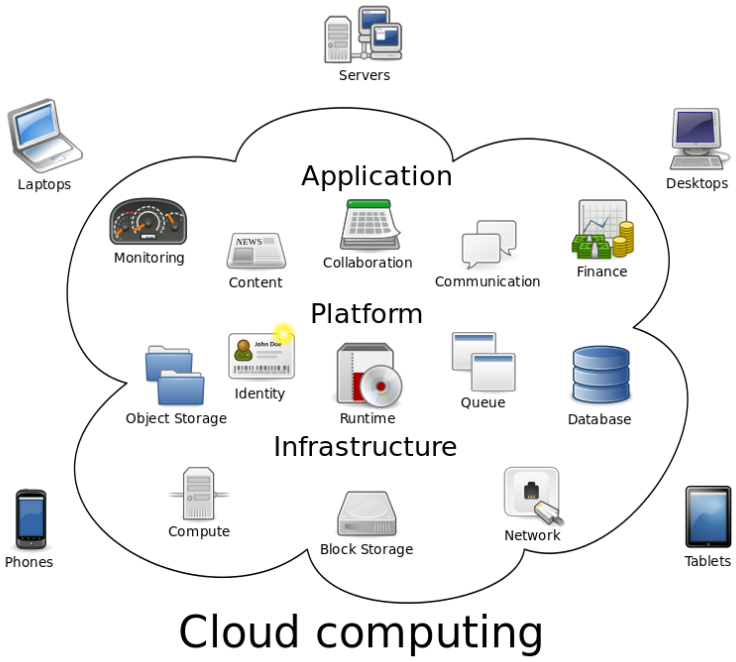

Figure 1:Paraqitja grafike e një sistemi cluster

Burimi: Cloud_computing.svg.png

1.2.1.Shpërndarja e burimeve

Shpërndarja merret me përcaktimin e numrit të burimeve të përdorura nga një ngarkesë. Kjo përkthehet në përcaktimin e numrave të serverave, numrat e core-ve, sasia e memorjes dhe e bandwidth-it të burimeve për sever.

Menaxheret më të njohur si Mesos, Omega dhe Toque presin që ngarkesat e të dhënave të bëjnë një rezervim të burimeve. Mesos i proceson kërkesat dhe bazuar në rëndësinë e kërkesës i bën ofertë burimesh strukturave individuale, më pas strukturat mund ti pranojnë ose refuzojnë këto oferta. Ndërsa CloudScale, PRESS, AGILE e bëjnë këtë parashikim për burimet, online madje ndonjëherë dhe pa e patur një ide fillestare të ngarkesës.

1.2.2.Përcaktimi i burimeve

Përcaktimi merret me selektimin e burimeve specifike që kënaqin një shpërndarje. Dy sfidat e mëdha të kësaj pune janë heterogjeniteti i serverit dhe interferenca midis vendndodhjes së njëjtë të ngarkesave të të dhënave. Këtu mund të përmendim Quasar dhe Paragon. Paragon përdor metoda klasifikimi të cilat llogarisin impaktin e heterogjenitetit dhe interferences së ngarkesës. Paragon e përdor këtë informacion për të pëcaktuar çdo lloj ngarkese tek lloji i duhur i serverit që mundëson performancën më të mirë dhe bashkëvendos ngarkesat që nuk interferojnë me njëra tjetrën.

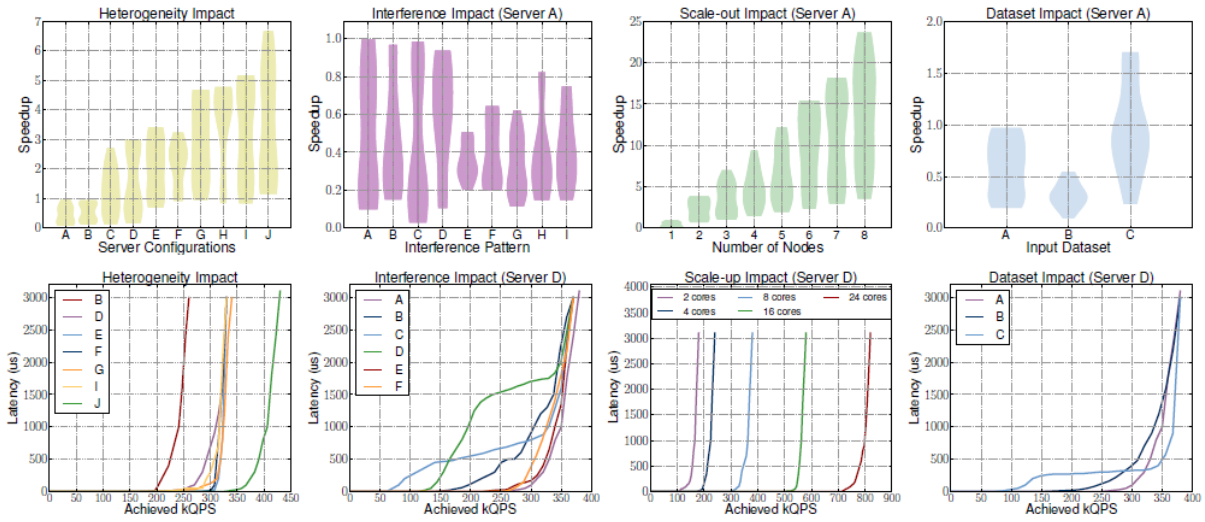

Figure 2:: Impakti i heterogjenitetit, interferences, përshkallëzimeve dhe performancës së grupit të të dhënave (dataset).

Burimi: 2014-asplos-quasar-Stanford-paper

Në rreshtin e parë të figurave mund të shohim grafikisht të paraqitura performancën e të dhënave në framework-un e Apache Hadoop dhe në rreshtin e dytë në sipas sistemit memcached.

Apache Hadoop është një sistem software(open souerce) që përdoret për të dhëna dhe procesim të ndarë të sasive shumë të mëdha të të dhënave në clustera.

Memcached është një sistem memorjeje që përdoret zakonisht për të përshpejtuar databazat dinamike të websiteve duke kapur të dhëna dhe objekte nga RAM-i për të reduktuar numrin e herëve që duhet lexuar një e dhënë nga burimi i saj.

2. Menaxhimi i kordinuar i Cluster-ave

Pavarësisht progresit në teknologjinë e menaxhimit të clusterave, përdorimi i burimeve është mjaft i ulët në përgjithësi për cloud-et publike dhe private. Dy janë mangësitë kryesore që ka menaxhimi aktual i cluster-ave.

Së pari është e vështirë që një përdorues ose ngarkesë të kuptojë nevojën e tij/saj për burime dhe më pas ta shprehë si një kërkesë për rezervim.

Së dyti përcakitimi i burimeve dhe shpërndarja e tyre janë të lidhura ngushtësisht. Një shpërndarje efiçente varet nga madhësia dhe tipi i burimeve të lira dhe gjithashtu nga sjellja e ngarkesave të tjera që po ekzekutohen në cluster.

platforms	A	B	C	D	E	F	G	H	I	J
cores	2	4	8	8	8	8	12	12	16	24
memory(GB)	4	8	12	16	20	24	16	24	48	48

interference	A	B	C	D	E	F	G	H	I
pattern	-	memory	L1I $	LL $	disk I/O	network	L2 $	CPU	prefetch

input	A	B	C
hadoop	netflix: 2.1GB	mahout: 10GB	wikipedia: 55GB
memcached	100B reads	2KB reads	100B reads-100B writes

Figure 3:Platforma serveri (A-J), forma interference (A-I) dhe madhësi (dataset) (A-C) të përdorura për të analizuar.

Nga tabela shohim të ilustruar problematikat mbi impaktin e disa shpërndarjeve, përcaktimeve dhe ngarkesave komplet të ndryshme në dy aplikacionet e përmendur në figurë nga të cilët njëri është batch dhe tjetri me kohë përgjigjeje kritike. Për Hadoop raportojmë shpejtësinë mbi nyjen A duke përdorur gjithë coret dhe memorjen e mundshme nga serveri. Nga tabela shohim se vlerat

ndryshojnë sipas kapaciteteve të burimeve, pra varen nga coret dhe nga memorja e çdo serveri. Për memcached[1] mund të raportojmë vonesën midis leximeve.

Në rreshtin e parë të tabelës në fig.3 do të shohim të ilustruar sjelljen e Hadoop. Grafiku i heterogjenitetit tregon se konfigurimi i serverit prezanton një variancë për performancën mesatare deri në 7 herë ndërsa sasia e shpërndarjes së burimeve ndërmjet serverave varjon deri në 10 herë. Grafiku i interferencës na tregon se për serverin A, duke patur parasysh sasinë e burimeve të përdorura, Hadoop mund të mos jetë i ndjeshëm ndaj disa tipe interferencash ose ngadalësimesh deri në 10 herë. Gjithashtu grafiku i scale out tregon se në varësi të burimeve të përdorura për server scale out mund të jetë linear ose sublinear. Si përfundim grafiku i dataset-it tregon se kompleksiteti i datasetit dhe madhësia mund të ndikojnë deri në 3 herë mbi performancen e Hadoop.

Rezultatet janë të ngjashme për memcached, sic mund ti shohim dhe në rradhën e dytë të tabelës në fig. 3.

Pra është mëse e qartë për një përdorues apo ngarkesë që të përkthjë një qëllim në një fakt të rezervimit të burimeve. Që të përcaktojmë në mënyrë të saktë një shpërndarje, na duhet të kuptojmë përshkallëzimet, heterogjenitetin në serverat e alokueshëm momentalisht dhe interferencën midis të dhënave që po procesohen në moment dhe të dhënave të tjera specifike që po presin. Prandaj ndarja e burimeve dhe shpërndarjes nëpërmjet rezervimeve të parakohshme duket se nuk është optimale qoftë në termat e performancës, qoftë në termat e efiçencës. Njësoj jo-optimale do të jetë situata nëse si fillim do të mereshim me shpërndarjen, pastaj përcaktimin. Menaxhimi i clusterave duhet që ti përballojë të dyja punët në një mënyrë të integruar.

[1] Memcached – një sistem open source, i një performance të lartë me një memorje të ndarë dhe që është e drejtuar nga objektet. Përdoret për të përshpejtuar ëeb-et dinamike duke zbutur të dhënat e databazave të tyre.

3. Krijimi i një Cluster-i virtual me dy nyje

Në këtë pjesë do të krijojmë një cluster virtual nëpërmjet platformës Windows Server. Windows Server është një grup sistemesh operimi të dizenjuara nga Microsoft që kryejnë një numër funksionesh si menaxhim,ruajtje të dhënash(data storage),kontrollon aplikacionet dhe komunikimin etj.Për të bërë të mundur krijimin e clusterit,na duhet fillimisht të hapim programin dhe të shtojmë failover clustering feature.Failover Clustering është një tipar i Windows Server që shërben për të përmirësuar disponueshmërinë e shërbimeve dhe aplikacioneve

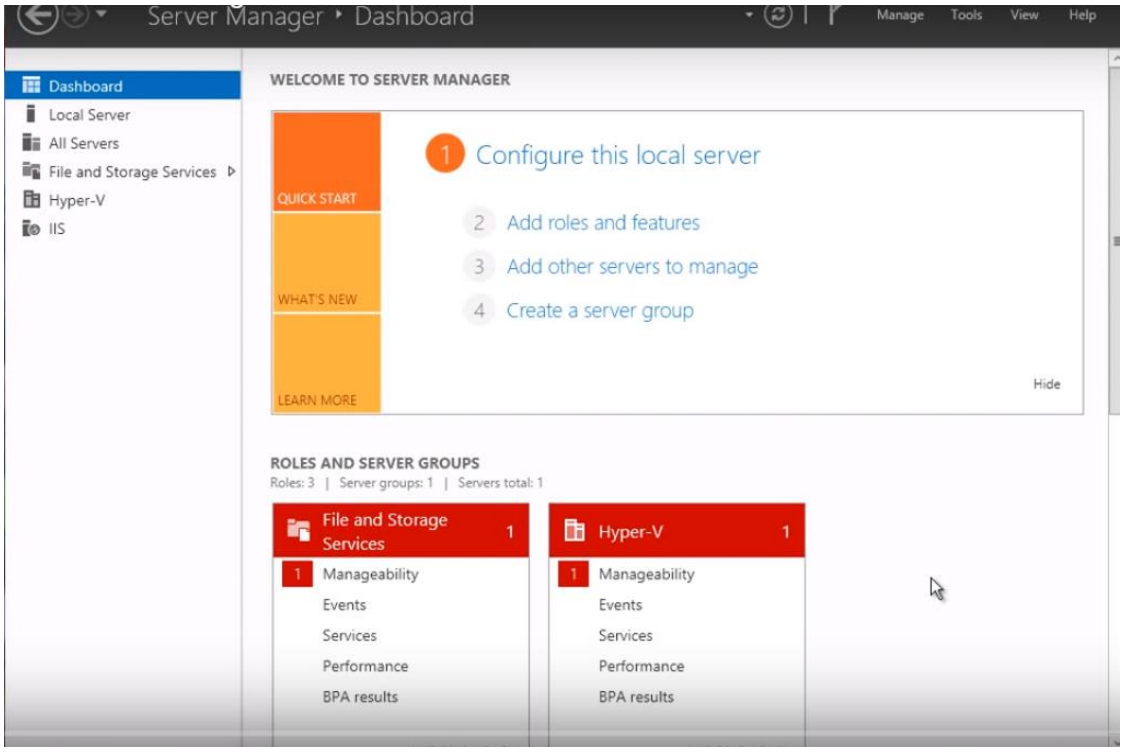

Figure 4:Një pamje e përgjithshme e Windows Serever

Burimi:Autori

Për t'i shtuar këtë tipar,duhet ta instalojmë failoverin duke klikuar në Local Server>Tasks>Add Roles and Features dhe klikojmë te opsioni failover clustering.Më pas presim deri sa të mbarojë instalimi.Pasi instalimi përfundon,vazhdojmë me pjesën e konfigurimit.

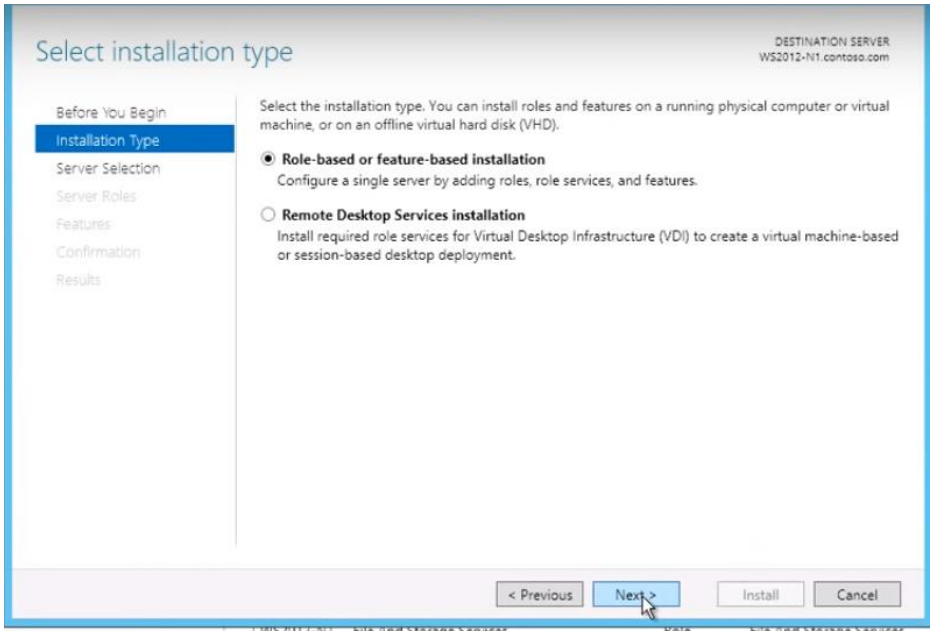

Figure 5:Instalimi

Burimi:Autori

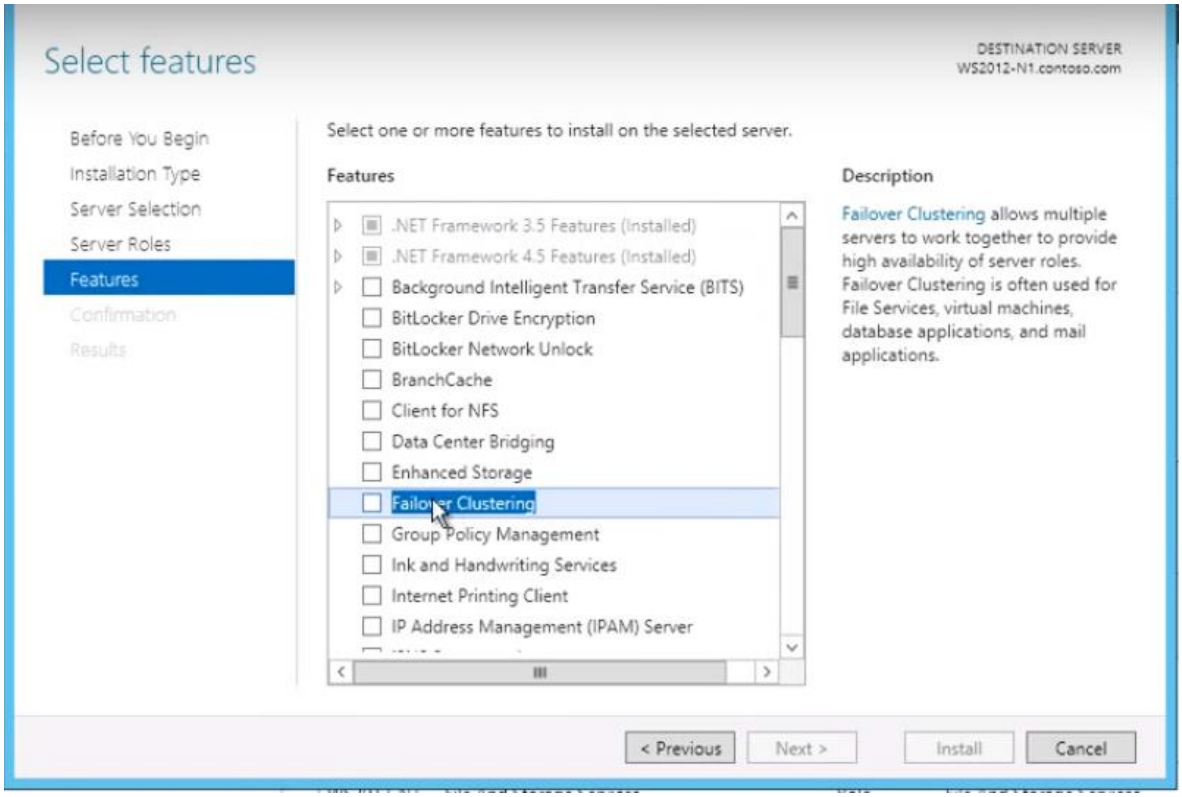

Figure 6:Zgjedhim opsionin failover clustering

Burimi:Autori

Për të bërë konfigurimin përpara se të krijojmë clusterin,futemi te menuja validate configuration.Na shfaqet menuja e mëposhtme.Emrat që janë nënvizuar në fushën selected janë dy nyjet që do të krijojmë.

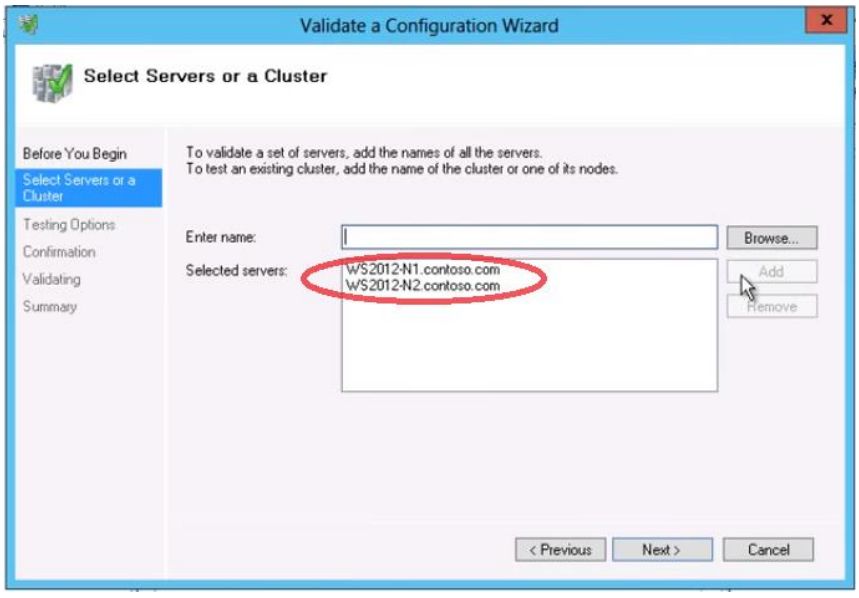

Figure 7:Menuja e shfaqur

Burimi:Autori

Më pas procedura është standarde,klikojmë në seksionin testing dhe i japim next.Ajo cfarë kemi bërë këtu është dy nyjet që krijuam i testojmë nëpërmjet një sërë testesh që i ka të gatshme Windows Server.Në përfundim të testit,na mbetet pjesa e fundit,që është krijimi i clusterit.

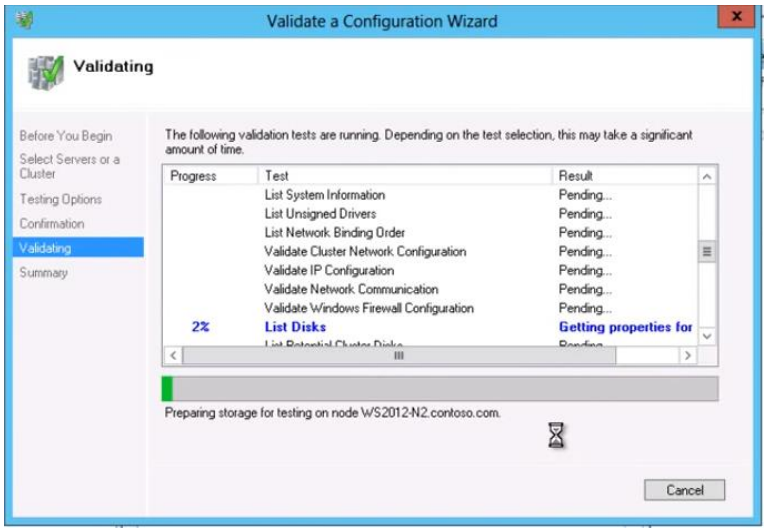

Figure 8:Nyjet i vendosen testimit për validimin e tyre

Burimi:Autori

Futemi në menunë create cluster,në dritaren që na shfaqet fusim emrat e dy nyjeve që i validuam më parë.

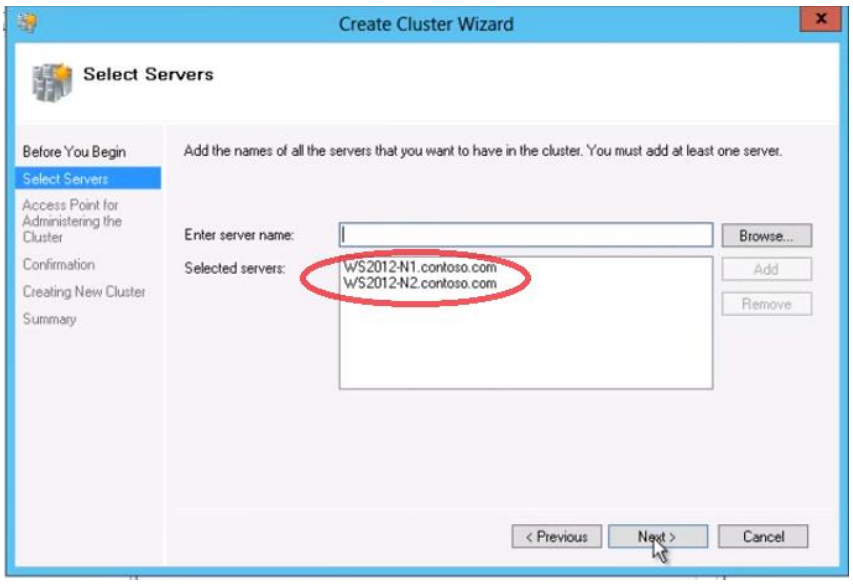

Figure 9:Create cluster wizard

Burimi:Autori

Në përfundim duhet të na shfaqet tabela e mëposhtme.

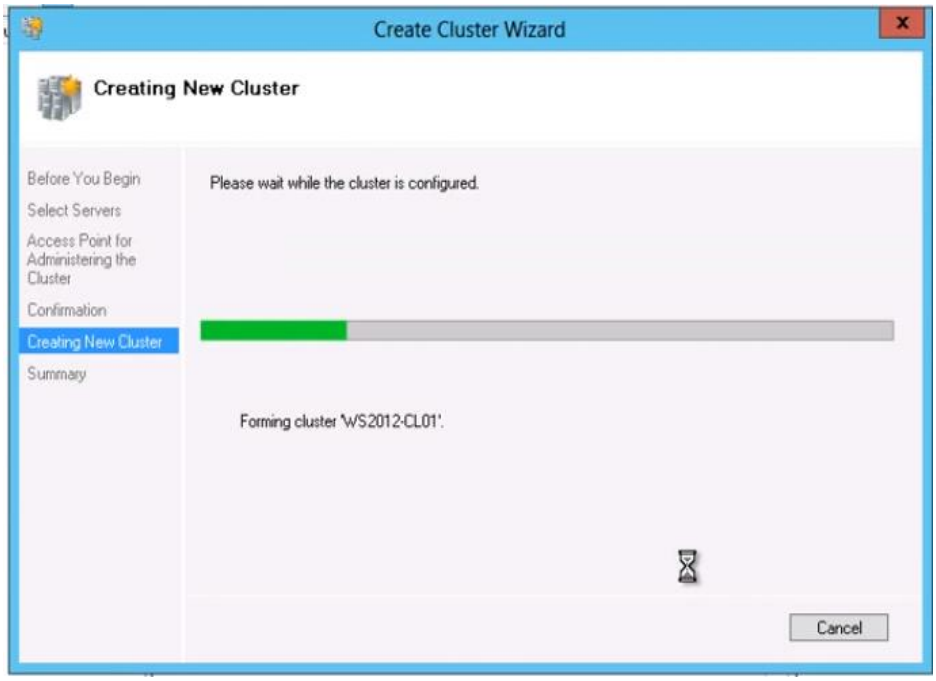

Figure 10:Krijimi i clusterit

Burimi:Autori

4.Sistemi Quasar

4.1 Njohje me sistemin e menaxhimit Quasar

Definicion i termit Quasar:Një sistem menaxhimi i clusterave që rrit përdorimin e burimeve duke rritur ndërkohë performancën e aplikacioneve.Quasar rekomandohet si një mënyrë e mirë menaxhimi për tre arsye kryesore:

- I largohet rezervimit të burimeve dhe adopton një metodë të drejtuar nga rregullimi i performancës. Quasar ka gjithashtu një interface të nivelit të lartë që i lejon përdoruesit të integrohen në disa lloj procesimesh, të shprehin maksimumin e dëshiruar të performancës që procesimi duhet të takojë. Interface ndryshon sipas tipit të ngarkesave.
- Quasar përdor teknika të shpejta klasifikimi për të vlerësuar në mënyrë të saktë impaktin që shpërndarja e burimeve të ndryshme kanë në performancë.

- Quasar prezanton një interface për mbi performancën e kërkuar në mënyrë që të vendosë qëllimet e duhura për procesimin e të dhënave.

4.2 Klasifikim i shpejtë dhe i saktë

Shpesh teknika filtrimi janë përdorur në sistemet me të dhëna shumë të rralla. Një nga rastet më të njohura ishte sfida e Netflix, ku teknika si Shpërbërja e vlerave teke (SVD) dhe PQ-reconstruction përdoreshin për të rekomanduar filma tek përdoruesit që kanë vlerësuar shumë pak filma të tjerë. Të dhënat në SVD në këtë rast do të ishin në formën e një matrice me pak elementë që do të specifikonin vlerësimet për film.

Në Paragon, filtrimi bashkëpunues u përdor për të klasifikuar shpejt ngarkesa sipas interferencës dhe heterogjenitetit. Pak aplikacione janë klasifikuar si të lodhshme për ta kaluar punën e tyre në servera të ndryshëm dhe majde me një varietet te konsiderueshëm interference.Paragon tregoi se filtrimi bashkëpunues mund të klasifikojë shpejt dhe saktë aplikacione të panjohura duke respektuar për dhjetra konfigurime serverash dhe burime interference.

Motori i klasifikimit në Quasar e tejkalon Paragon-in në dy mënyra:

- Së pari, ai përdor filtrimin bashkëpunues për t'ju afruar impaktit të scale out dhe scale up(pra më shumë servera dhe më shumë burime për server) sipas performancës së aplikacionit. Këto klasifikime shtesë janë të nevojshme për të bërë të mundur shpërdarjen e burimeve.
- Së dyti, i Quasar adopton mënyra të ndryshme për klasifikimin e ngarkesave të tipeve të ndryshme. Kjo është e nevojshme sepse tipe të ndryshme ngarkesash kanë tipe të ndryshme kufizimesh.

Mbi të gjitha Quasar klasifikon për të dyja llojet e përshkallzimeve(scale-out dhe scale-up), heterogjenitetin dhe interferencën. Të katerta klasifikimet bëhen si të pavarura nga njëra-tjetra dhe në mënyrë paralele për të reduktuar kompleksitetin dhe mbivendosjet. Algoritmi bën të mundur kombinimin e informacionit për të katërta.

Klasifikimi scale-out

Ky klasifikim shpjegon si varion performanca në një madhësi burimesh të përdorur në një server. Klasifikimi do të realizohet në bazë të core-ve të kompjuterit, memorja dhe kapaciteti i kujtesës.

Kur një ngarkesë merret në dorëzim ne e klasifikojmë shpejt sipas dy shpërndarjeve të zgjedhura rastësisht. Parametrat dhe kohëzgjatja e scale-out varen nga tipi i ngarkesës. Shërbimet që ekzekutojnë në bazë të vonesës kritike si prsh memcached profilizohen për 5-10 sekonda sipas dy vendndodhjeve të ndryshme në bazë të core-ve dhe memorjes. Në aplikacione si Cassandra të cilat mbajnë "shënime" për gjendjet e mëparshme, Quasar pret derisa të mbarojë vendosja e plotë e shërbimit, pastaj e profilizon ngarkesën e inputit në vendndodhje të ndryshme. Kjo merr pak a shumë një kohëzgjatje prej 3-5 minutash maksimumi, e cila tolerohet për shërbimet që kanë një kohë të gjatë ekzekutimi.

Profilizimi mbledh madhësitë e performancave në formatin e qëllimit të performancës që ka çdo aplikacion dhe i vendos në matricën A të dhënat për çdo ngarkesë sipas përshkallëzimit dhe konfigurimit. Konfigurimi përfshin llogaritjet, memorjen dhe vendin e kujtesës ose vlerat e procesimit dhe parametrave për ngarkesat si Hadoop. Kjo në teori mund të cojë në disa vendime jo-optimale por devijimet ne jetën reale jane te vogla. Klasifikimi i bërë duke përdorur SVD dhe PQ-reconstruction na con në derivimin e performancës së ngarkesës.

Klasifikimi i heterogjenitetit:

Ky klasifikim kërkon një tjetër ekzekutim për të profilizuar në një tjeter lloj serveri të zgjedhur rastësisht duke përdorur të njëjtat parametra dhe të njëjtën kohë si në rastin e përshkallëzimit sipas rritjes së parametrave të serverit. Filtrat bashkëpunues e vlerësojnë performancën e ngarkesës nëpërmjet serverave të tjerë.

Klasifikimi i interferencës:

Ky lloj klasifikimi vlerëson sensitivitetin e ngarkesës së interferencës së shkaktuar dhe toleruar në disa burime të shpërndara, duke përfshijmë CPU, hierarkinë e cache-se, kapacitetin e memorjes dhe bandwidthin, kujtesën dhe bandwidthin e rrjetit. Ky klasifikim nuk kërkon një ekzekutim të ri të profilizimit. Ai përdor kopjen e parë të klasifikimit sipas përshkallëzimit nëpërmjet rritjes së parametrave të serverave dhe injekton nga një nga një dy microbenchmark që janë miniprograme të cilat shërbejnë për të matur performancën e një pjese kodi. Pas injektimit, Quasar e rrit intensitetin e tij derisa performanca e ngarkesës të bjerë poshtë nivelit të pranueshëm të QoS

Klasifikimi paralel apo një klasifikim kompleks?

Klasifikimi si në rastin efiçent dhe në rastin joefiçent duhet ti analizojë të katërt komponentët. Problemi qëndron cilën nga metodat do të zgjedhim si përfundim, metodën që bën analizën duke i krahasuar dhe bërë profilizimin në pralel, apo metodën komplekse që i përfshin të gjithë parametrat për të bërë një profilizim përfunditmtar. Klasifikimi i zgjatur mund të adresojë raste që klasifikimi me katër nëndarje mund ti vlerësojë në mënyrë të dobët. Megjithëse këto raste janë të rralla, ato mund të rezultojnë në rezultate me performancë të papritur. Nga ana tjetër rritja eksponenciale e veprimeve në rastin e klasifikimit kompleks rrit kohën e kërkuar për të performuar klasifikimi.

Për tju adresuar këtij problemi ekziston një cikël feedbacku që updaton matricën me të dhënat e duhura kur performanca e matur gjatë kohës së ekzekutimit derivon nga koha aktuale e vendosur në matricë sipas klasifikimit. Ky cikël bëhet për shkak të klasifikimeve jo të sakta dhe asiston në mënyrë të vazhduar.

Vlerësimi

Dy figurat në vazhdim do të paraqesin tabelat me të dhënat mbi secilin nga rastet e klasifikimit.

Default density constraint: 2 entries per row, per classification												
	scale-up			scale-out			heterogeneity			interference		
Classification err.	*avg*	90^{th}	*max*	*avg*	90^{th}	*max*	*avg*	90^{th}	*max*	*avg*	90^{th}	*max*
Hadoop (10 Jobs)	5.2%	9.8%	11%	5.0%	14.5%	17%	4.1%	4.6%	5.0%	1.8%	5.1%	6%
Memcached (10)	6.3%	9.2%	11%	6.6%	10.5%	12%	5.2%	5.7%	6.5%	7.2%	9.1%	10%
Webserver (10)	8.0%	10.1%	13%	7.5%	11.6%	14%	4.1%	5.1%	5.2%	3.2%	8.1%	9%
Single-node (413)	4.0%	8.1%	9%	-	-	-	3.5%	6.9%	8.0%	4.4%	9.2%	10%

Figure 11:Vlerësimi i klasifikimit sipas Quasar.

Burimi: 2014-asplos-quasar-Stanford-paper

	8 entries per row		
	exhaustive classification		
Classification error	*avg*	90^{th} *%ile*	*max*
Hadoop (10 Jobs)	14.1%	15.8%	16%
Memcached (10)	14.1%	16.5%	18%
Webserver (10)	16.5%	17.6%	18%
Single-node (413)	11.6%	12.1%	13%

Figure 12:Krahasim i gabimeve të klasifikimit

Burimi: 2014-asplos-quasar-Stanford-paper

Janë përdorur 40 servera clouster dhe aplikacione nga Hadoop (10 punë data minig) , shërbimet me vonesë kritike (10 punë memcached, dhe 10 ngarkesa të dhënash nga Apache webserver) dhe 413 benchmarks nga SPEC, PARSEC, SPLASH-2, BioParallel, Minebench dhe SpeCJBB. Punët e memcache dhe webserving dallojnë nga mënyra si i ndajnë kërkesat, mënyra e inputit të datasetit apo e outputit të ngarkesës. Hadopp jobs diferencojnë në termat e logjikës së aplikacioneve.

Nga tabela e parë shohim se në një mesatare prej 90 % dhe maksimumet e gabimeve për cdo aplikacion dhe tip klasifikimi i shohim të paraqitura në tabelën e dytë. Gabimet tregojnë një devijim midis matjeve dhe parashikimit të performancës apo sensitivitetit ndaj interferencës. Në një mesatare gabimet e klasifikimit janë më pak se 8% përgjatë gjithë tipeve të aplikacioneve, ndërkohë që maksimumi arrin me pak se 17%, duke garantuar se informacioni që drejton menaxhimin e cluster-ave është i saktë.

Nga tabela e parë mund të shohim gjithashtu dhe rezultatet e klasifikimit kompleks të cilat janë pak më të larta, veçanërisht për aplikacionet që arrijnë herët në sistem, gjithsesi devijimi nga mesatarja dhe maksimumit është më i ulët, duke qenë se kompleksiteti i procesit mund të parashikojë në mënyrë të saktë performancën për rastet e veçanta që modeli me katër klasifikime nuk i parashikonte dot.

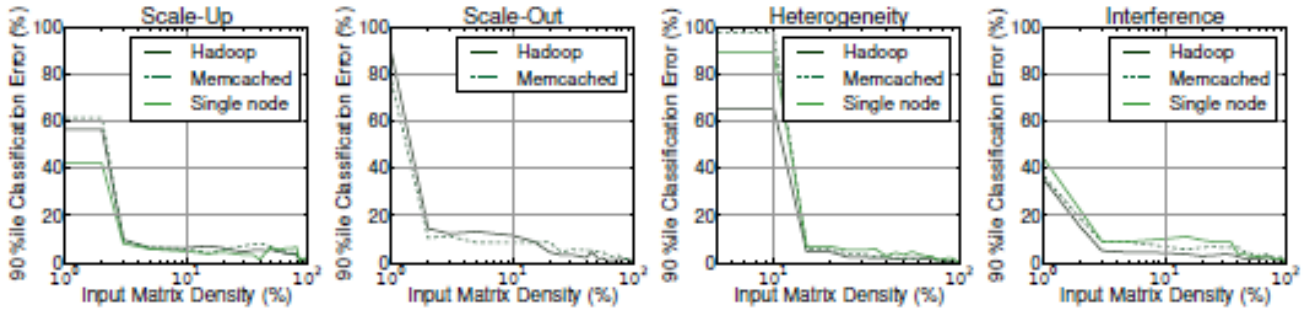

Figure 13:Sensitiviteti i vërtetësisë së klasifikimit ndaj densitetit të inputit të kufizimeve të matricës

Burimi: 2014-asplos-quasar-Stanford-paper

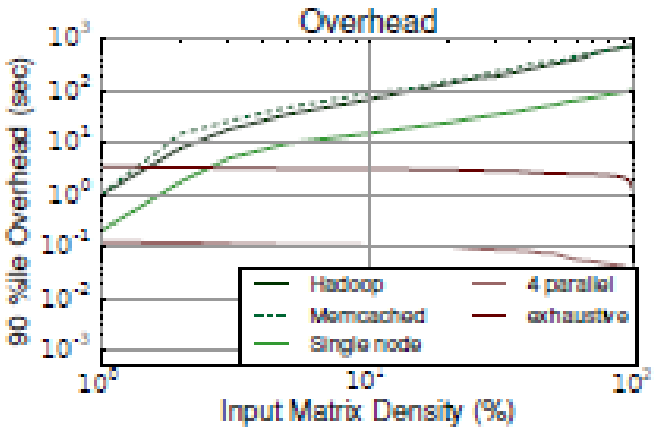

Figure 14:Profilizimi dhe vendimi për densitete të ndryshme të kufizimeve për inputet.

Burimi: 2014-asplos-quasar-Stanford-paper

Mund të japim gjithashtu një vlerësim mbi numrin e procesimit të profilizimit, prsh vërtetësia e klasifikimit në bazë të rritjes së densitetit të inputit në matricë. Për qartësi më të madhe iu referohemi të dhënave për webserverat që kanë një formë të ngjashme me atë të memcached. Për të katërt llojet e klasifikimeve pocesi i një klasifikimi të vetëm rezulton me një numër të madh gabimesh. Dy ose më shumë rreshta inpute rezultojnë një një zvogëlim të shkallës së gabimeve. Kjo sjellje është konsistente midis tipeve të aplikacioneve megjithse vlerat ekzakte të gabimeve mund të vajojnë. Fig 12 na tregon gjithashtu rezultatet nga klasifikimi për modelin me katër procese në paralel dhe skemën komplekse. Siç mund të pritej rritja e procedurave llogaritëse i korespondon një rritje e konsiderueshme në kohë shpesh me dy kurba maksimumesh.

4.3 Shpërndarja dhe përcaktimi

Klasifikimi i jepet për t'u kryer një programi që përcakton sasinë dhe tipin ekzakt të burimeve të shpërndara. Objektivat e këtij procesi organizues janë që të shpërndajë sasinë e fundit të burimeve të nevojitura për të realizuar synimin e performancës së një ngarkese. Ky lloj modeli përdoret nga shumë servera.

Organizuesi përdor outputin e klasifikimit që të rankojë serverat e lirë duke ulur kualitetin e burimeve, psh platformat me performancë më të lartë dhe me interferencë minimale do të zgjidhen të parat. Më pas ai mat vendet te tjera të lira në burime derisa performanca të jetë në rregull sipas kufizimeve.Prsh nëse një webserver duhet të qëndrojë midis 100'000 QPS me kohë vonese 10milisekonda por serverat e rankuar më shumë mund të arrijnë deri në 20'000 QPS,

ngarkesa do të përdorë 5 servra për të realizuar performancën sipas kufizimeve. Kur e arrin shpërndarjen, algoritmi së pari rrit burimet për çdo nyje dhe rrit punën për çdo server dhe më pas shperndan drejtimin përgjatë makinerisë.

4.4 Përmbledhje e kapitullit

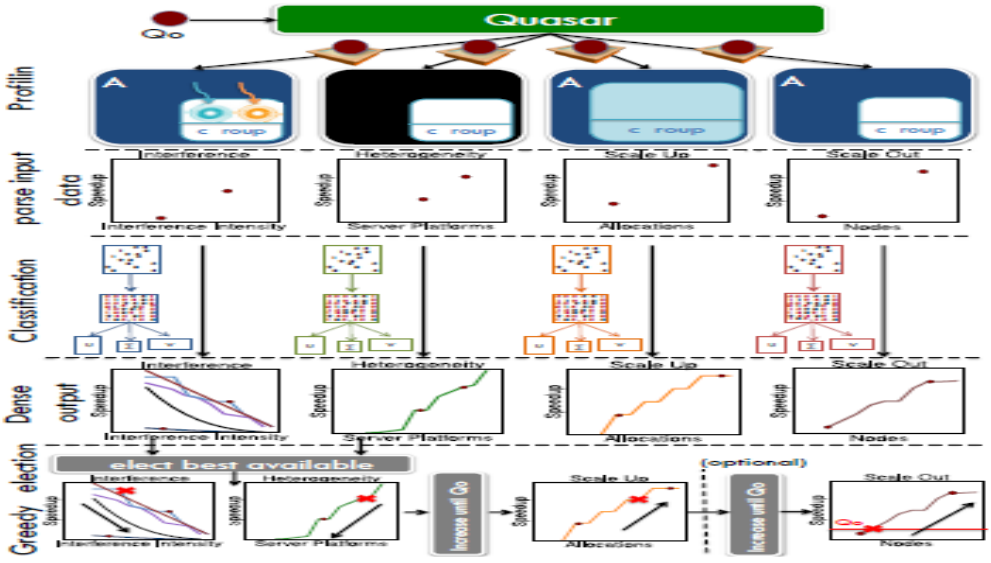

Figure 15:Hapat për menaxhimin e clusterave me Quasar

Figura më lartë na paraqet gjithë hapat që i duhen të bëjë Quasar për menaxhimin, duke filluar nga lartë me short-run et e vogla që më pas mirpresin ngarkesat me të dhëna, të cilat kalojnë përmes klasifikimit tek produkti i klasifikuar sipas metodës me katër profilizimet paralele për të arritur më në fund tek algoritmi i përcakitimit dhe shpërndarjes dhe oraganizatori (greedy scheduler) dhe në fund tek procesimi i të dhënave sipas QoS.

Le ta shtjellojmë pak më shumë për ta kuptuar më mirë. Me ardhjen e të dhënave, Quasar fillon profilizimin për shpërndarjen sipas përshkallëzimit të duhur, përcaktimit, heterogjenitetit dhe interferencën. Kjo kërkon deri në katër ekzekutime që ndodhin në paralel. Të gjitha kopjet e profileve janë sandboxes[2], dy platformat e përdorura janë A dhe B dhe secili tip profili prodhon dy pikë në grafikun koresponduese në grafikun e përshpejtimit të ngarkesës. Profili ekzekutohet me datasetin aktual tl ngarkesës. Profilizimi total si rezultat varet nga tipi i ngarkesës dhe merr më pak se 5 minuta në të gjitha rastet e ekzaminuara. Për shërbimet që nuk mbajnë shënime mbi gjendjet e tyre prsh ngarkesat e vogla batch që janë një fraksion i konsiderueshëm i ngarkesave të

[2] Një vend virtual ku një aplikacion i patestuar mund të ekzekutohet I sigurtë.

datacenterave, profilizimi total merr 10-15 sekonda. Ndërsa për shërbimet si Cassandra që mbajnë shënime mbi gjendjet ku nevoja e setupit është e nevojshme ndikon vetëm tek njëri nga llojet e ekzekutimit të profilizimit. Prandaj sapo shërbimi ka filluar të punojë, profilizimi merr vetem pak sekonda për tu plotësuar dhe kështu kategorizimi na mundëson karakterisitikat e të gjithë ngarkesës. Mbi të gjitha përfundimet e Quasar janë shumë të shpejta edhe pë aplikacionet me kohë të shkurtër ekzekutimi apo për shërbimet online me kohë të gjatë ekzekutimi.

Quasar mban të dhëna për lloj ngarkese dhe për lloj serveri. Për lloj ngarkese përfshinoutputin e klasifikimit. Për një cluster me 10 tipe serverash dhe 10 burime interference na duhen afershisht 256 byte për ngarkesë. Për lloj serveri updatohet për çdo përcakitim ngarkese. Quasar ka gjithashtu nevojë për pak memorje për klasifikimin e mesëm të rezultateve dhe pë rankimin e serverave gjatë përcakitimeve. Mbi të gjitha hapësira shtesë që ze Quasar është shumë e vogël.

5.Krahasimi me sisteme të tjera

Këtu do të shikojmë situata të ndryshme krahasuese të këtij sistemi me procedura të tjera veprimi për menaxhim dhe efektivitet të clusterave. Paragrafët në vijim përmbledhim skenarët e ngarkesave që vlerësojnë performacën e Quasar.

5.1 Single batch job

Performanca:

Framework-ët si Hadoop,Storm dhe Spark janë konsumues të mëdhenj të burimeve në cloud-et publike dhe private.Në skenarin e parë,një punë e vetme në Hadoop(single batch job) është duke u përpunuar në një kohë në një cluster të vogël.Ky skenar i thjeshtë na lejon që të krahasojmë alokimin e burimeve të realizuar nga Hadoop me atë të kryer nga Quasar.

Në këtë rast Quasar e përmirëson performancën për të gjitha punët mesatarisht nga 29% deri në 58%. Quasar arrin një performancë jo më pak se 5.8% nga masa që kërkohet detyrimisht, duke

lënë të kuptohet informacioni sesi shpërndarja dhe caktimi i burimeve ndikojnë në performancë. Kur burimet shpërndahen sipas Hadoop, performanca devijon nga targeti i vendosur me mesatarisht 23%.

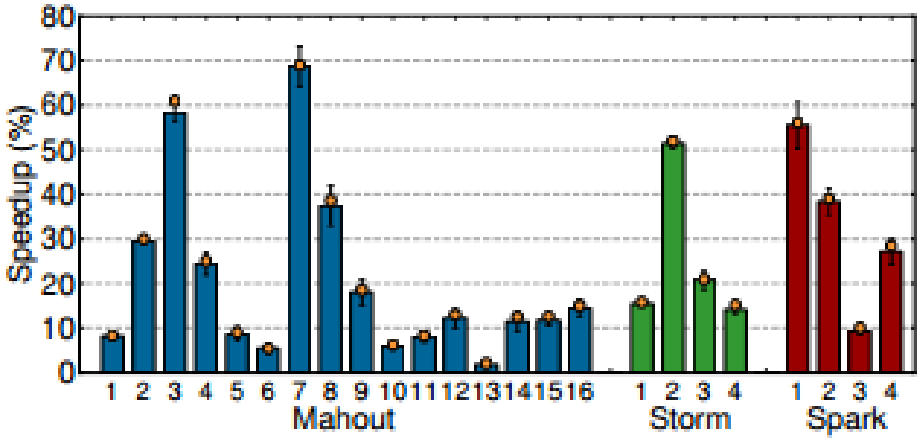

Figure 16:Shpejtësimin e performancës për Hadoob, Storm dhe Spark jobs me Quasar.

Burimi: 2014-asplos-quasar-Stanford-paper

5.2 Multiple Batch Frameworks

Skenari i dytë prezanton një mjedis realist për clusterat batch processing. Clusteri është i ndarë ndërmjet punëve nga framework-ët si Hadoop,Storm,Spark.

Performanca :

Mesatarisht, performanca përmirësohet me 27% dhe është jo më pak se 5.3% e masës që kërkohet, një përmirësim domethënës. Quasar përveç matjes dhe konfigurimit më të mirë të punëve, mund të detektojë kur 2 punë Storm dhe Spark që kërkojnë shumë memorie (memory-intensive) interferojnë dhe kur është e mundur që ta përdorin bashkarisht sistemin në mënyrë efiçente. Quasar lejon kapacitetin e mbetur të cluster-it të përdoret për punët në kohë përpjekjeje më të mirë pa ndërhyrë në punët primare sepse është në kontroll të interferencave. Punët me

perpjekjen më të mirë i gjejmë rreth një mesatareje 7.8% në performancën e kërkuar që çdo punë do të mund të arrinte nëse do të ishte duke u ekzekutuar vetëm në serverin që i përputhet karakteristikës së të dhënës.

5.3 Shërbimi me vonesa të vogla në ekzekutim

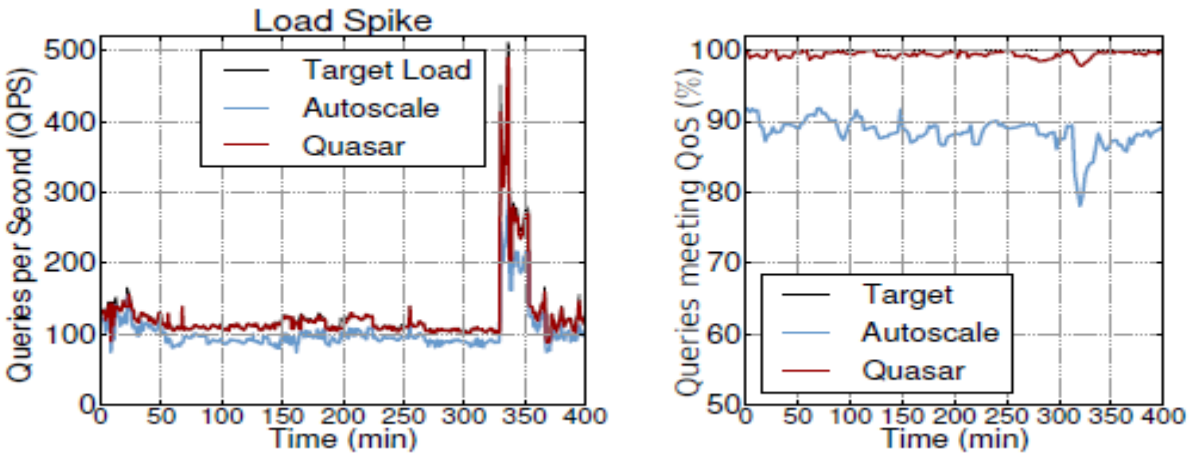

Figure 17: Rezultate për HotCRP(a), Fraksioni i takimit dhe vonesës së kërkesave (b)

Burimi: 2014-asplos-quasar-Stanford-paper

Figura 17 (a) na tregon throughput-in agregat për HotCRP që arrihet me Quasar dhe sistemin auto përshkallëzues kur trafiku hyrës është i sheshtë.Ndërkohë që diferencat absolute janë të vogla, është e rëndësishme të theksojmë se menaxheri i përshkallëzimit shkakton rënie të shpeshta të QPS si pasojë e interferencës nga punët me perpjekje më të mirë që mund të zënë burimet pa nevojën për t'i përdorur. Me Quasar, HotCRP operon i pashqetësuar dhe punët best-effort arrijnë kohë ekzekutimi jo më shumë se 5% minimalisht, ndërkohë që me auto-scale ato arrijnë minimalisht një përqindje prej 24. Kur trafiku nuk është i sheshtë (varion)(fig 17 b) , Quasar gjurmon targetin e QPS-në me kujdes, ndërkohë që efekti autopërshkallëzues i ofron një nivel 18% më të ulët QPS se mesatarja, të dyja falë interferencës dhe konfigurimit mbioptimal që ka lidhje me rritjen e madhësisë së parametrave të serverave. Sjellja e qetë e Quasar mundësohet falë përdorimit të të dy llojeve të përshkallëzimitpër të arritur targetin e dëshiruar të QPS dhe duke lene numrin më të madh të coreve të lira për punët që kërkojnë ekzekutim sipas përpjekjes më të mirë. Për ngarkesën me një goditje fikse Quasar ndjek QPS me një mesatare 4 % dhe arrin

vonesën e QoS për afërsisht të gjithëa kërkesat. Kur goditja arrin Quasar si fillim përshkallëzon burimet aktualë te serverave dhe më pas thjesht përdor dy servera shtesë të tipit të duhur për të përballuar trafikun e mbetur. Sistemi i auto përshkallëzimit observon rritjen e ngarkesës kur goditja arrin dhe alokon 4 servera më shumë. Për shkak të vonesës së lartë të rezervimit të me shumë serverave dhe faktit se auto përshkallëzimi nuk është i ndjeshëm ndaj heterogjenitetit apo interferencës, bën që të dështojë arritja e garantimit të vonesave për mbi 20% të kërkesave në kohën e goditjes së pikut.

5.4 Shërbimet me vonesë kritike

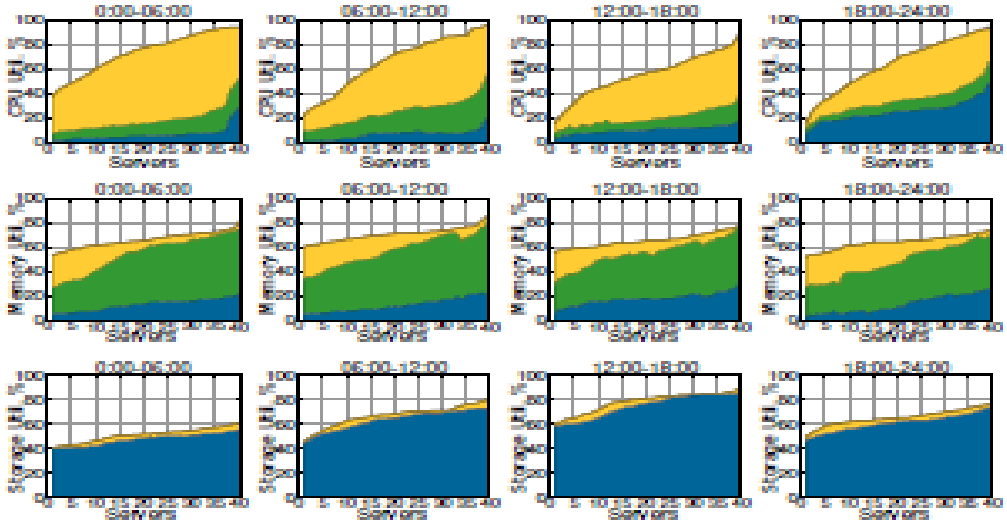

Figure 18:Mesatarja e përdorimit të burimeve përgjatë gjithë serverave për cdo katër 6-orarsh të shënuar.

Burimi: 2014-asplos-quasar-Stanford-paper

Clusteri që po ekzekutohet si memchached (jeshile), Cassandra (blue), përpjekja më e mirë (e verdhë) dhe menaxhohet nga Quasar.

Performanca: Figura më poshtë tregon gjithë procesin e memcached dhe Cassandra ndër kohë dhe shpërndarjen e vonesës së shpërndarjes. Quasar e ndjek targetin nga afër për të dy shërbimet, ndërkohë që menaxheri i autopërshkallëzimit i ul ndërveprimet në një mesatare prej 24% dhe 12% për memcached dhe Casandra, reciprokisht. Diferencat në vonesa janë më të mëdha midis

dy menaxhimeve. Quasar arrin vonesën QoS për memcached për 98.8% të kërkesave, ndërkohë autopërshkallëzimi vetëm për 80% të kërkesave. Për Cassandra, Quasar e arrin vonesën e QoS për 98,6% të kërkesave, ndërkohë që auto përshkallëzimi për 93 % të kërkesave. Memcached është e bazuar në memorje dhe ka një vonesë më agresive kundrejt QoS, duke e bërë atë më sensitive ndaj burimeve më se optimale të përdorimit dhe shpërndarjes në një cluster të shpërndarë (shared).

Përdorimi: Figura më lartë tregon përdorimin e CPU, sipas kapacitetit të memorjes dhe bandwidthit të diskut përgjatë serverave të clusterit kur menaxhohet nga Quasar për 24 orë. Cdo kolonë është një foto e përdorimit mesatar çdo 6 orë. Meqë memchached dhe Cassandra kanë kërkesa të ulëta ndaj CPU, përvec periudhës 18:00-24:00 kur Cassandra performon koleksionin e "plehrave", shumica e kapacitetit CPU është e lidhur me punët që kërkojnë përpjekjen më të mirë. Numri i punëve të përpjekjes më të mirë varion sipas kohës sepse ngarkesa ekzakte e memcaches dhe Cassandres ndryshon. Shumica e memorjes tani përdoret për të kënaqur kërkesat e memcached. Cassandra është afër përdoruese e diskut I/O. Disa servera nuk i kalojnë 40-50% të kapacitetit të përdorimit nga provat emësipërme. Këto makina për të cilat përdorimi i lartë rritet dramatikisht, rrisin gjithashtu probabilitetin e violencës së kushteve të QoS për shërbimet me vonesë kritike. Në tërësi, përdorimi i clusterave është më i lartë sesa nëse cdo shërbim do të ekzekutohej në makinat e dedikuara.

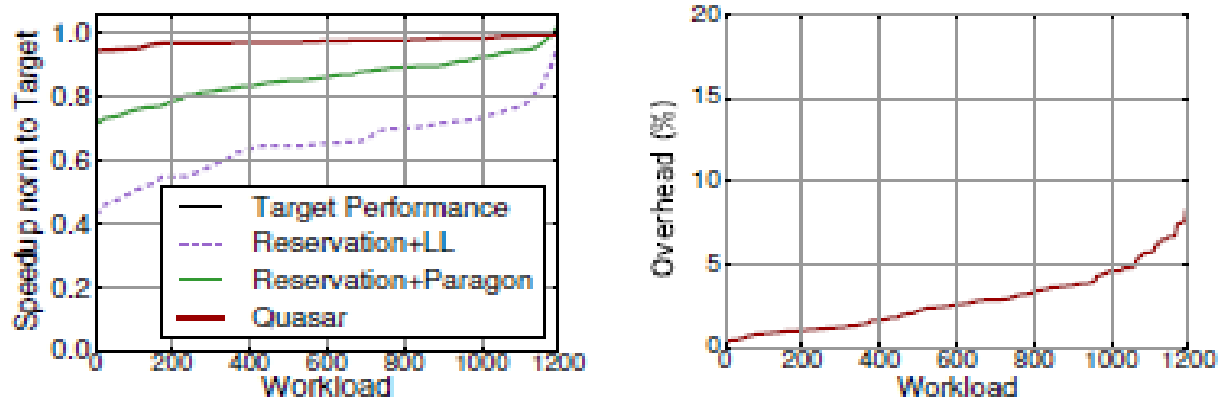

Figure 19:(a) Performancë rreth 1200 ngarkesa në 200 EC2 server me Quasar

Burimi: 2014-asplos-quasar-Stanford-paper

5.5 Mundësuesit e shërbimeve Cloud, të një shkalle të lartë

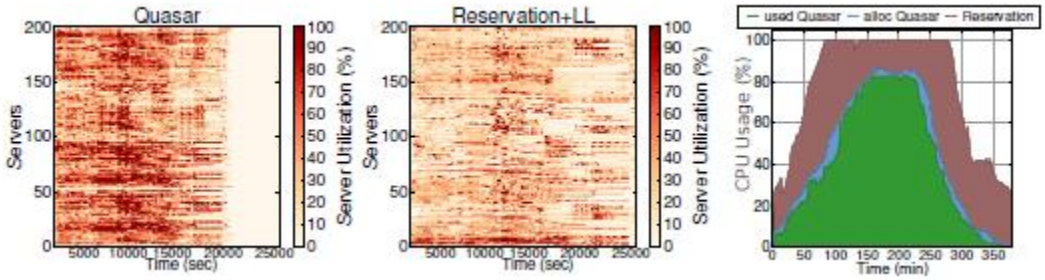

Figure 20:Përdorimi i Cluster-it për 1200 ngarkesa me servera 200 EC2

Burimi: 2014-asplos-quasar-Stanford-paper

Performanca: Figura më lartë prezanton vlerësimet e menaxhimit Quasar. Nga figura do të komentojmë performancën e 1200 ngarkesave të renditura nga performanca më e keqe tek më emira dhe të normalizuara në targetin e tyre te performancës. Quasar arrin 93% të targetit mesatar, nderkohë që sistemet që punojnë në bazë të rezervimit me Paragon arrijnë 83%. Kjo tregon nevojën për të performuar shpërndarjen dhe percaktimin së bashku, përndryshe nuk kemi një përdorim eficent të burimeve.

Përdorimi: Figura më lartë tregon përdorimin e CPU për server. Mesatarja e përdorimit për Quasar është 62, ndërohë që arrin kufizimet e performancës për batch dhe aplikacionet me kohë kritike vonese. Në figurën (c) shohim alokimin dhe burimet e përdorura për Quasar krahasuar me burimet e rezervuara nga rezervimi dhe Sistmin LL. Mbivendosja me Quasar është e ulët sepse Quasar ka informacion të detajuar sesa të ndryshme janë përcaktimet dhe shpërndarja, mund të vendosë në masën e duhur shpërndarjet ku janë më agresive, ndërkohë që arrin kufizimet e performancës pa cënuar QoS.

Konkluzione

Në këtë punim kemi shpjeguar koncepte të rëndësishme si cloud dhe cluster, jemi njohur me Quasar, një sistem menaxhimi cluster-ash që performon i kordinuar me përcaktimin e burimeve dhe shpërndarjen në të njëjtën kohë.

Quasar i largohet metodave me bazë rezervimin dhe prezanton një metodë të re. Në vend që të kemi shumë përdorues që kërkojnë gjithmonë burimet më të mira ai tani thjesht specifikon performancën që i duhet dhe lë menaxherin të merret me rregullimin e burimeve. Quasar analizon metodat e përshkallëzimit të problemit duke marrë parasysh heterogjenitetin e të dhënave dhe interferencën në performancë. Quasar shoqërohet nga një algoritëm qe perdor këto informacione dhe bën përcaktimin më të mirë të mundshëm për burimet.

Ky sistem tashmë mbështet disa procese pune analitike, webservers, NoSQL datastores, si dhe aplikacionet e vetme me një nyje, batch.

Nga krahasimet e shumta që i bëmë këtij sistemi në disa lloj situatash mund të themi se Quasar përmirëson përdorimin agregat të cluster-ave dhe performancën e aplikacioneve individuale.

Referenca

[1] A Comparative Analysis: Grid, Cluster and Cloud Computing:Kiranjot Kaur1, Anjandeep Kaur Rai2 M.Tech Scholar, Department of Computer Science and Engineering,Lovely Professional University, Phagwara, India1, 2

[2] An Introduction to Data Mining
http://www.thearling.com/text/dmwhite/dmwhite.htm

[3] Allen B. Downey and Dror G. Feitelson. The elusive goal of workload characterization. SIGMETRICS Perform. Eval. Rev., 26(4):14–29, March 1999.

[4] Anton Beloglazov and Rajkumar Buyya. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. IEEE Trans. Parallel Distrib. Syst., 24(7):1366–1379, July 2013.

[5] Christina Delimitrou and Christos Kozyrakis. The Netflix Challenge: Datacenter Edition. Los Alamitos, CA, USA, July 2012. IEEE Computer Society.

[6] Christina Delimitrou, Sriram Sankar, Kushagra Vaid, and Christos Kozyrakis.

[7] Christina Delimitrou and Christos Kozyrakis. Quasar: Resource-Efficient and QoS-Aware Cluster Management. In Proceedings of the Nineteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). Salt Lake City, UT, USA, 2014

[8] Cluster management: Maintenance and monitoring. https://www.cloudera.com/content/cloudera/en/developers/home/developer-admin-resources/cluster-management.html.14

[9] Decoupling Datacenter Studies from Access to Large-Scale Applications: A Modeling Approach for Storage Workloads. In Proceedings of the IEEE International Symposium on Workload Characterization (IISWC). Austin, TX, USA, 2011.

[10] Novella Bartolini, Giancarlo Bongiovanni, and Simone Silvestri. Self-overload control for distributed web systems. In Proceedings of the International Workshop on Quality of Service (IWQoS). Enschede, 2008

[11] Ubik: Efficient Cache Sharing with Strict QoS for Latency-Critical Workloads https://people.csail.mit.edu/sanchez/papers/2014.ubik.asplos.pdf

[12] What does “scale out” vs. “scale up” mean? http://ethancbanks.com/2014/12/17/what-does-scale-out-vs-scale-up-mean/

.[13] https://www.quora.com/What-is-cluster-management-And-why

[14] https://www.quasardata.com/services/services/colocation/high-performance-computing/

[15] https://docs.microsoft.com/en-us/windows-server/failover-clustering/create-failover-cluster