SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1954

Stvarnovremensko praćenje parametara ispravnosti rada u sustavu za raspodijeljenu obradu tokova podataka

Mislav Jakšić

Zagreb, travanj 2019.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

Hvala svima!

SADRŽAJ

1. Uvod

Raspodijeljeni sustavi su nepouzdani. Pogreška u sklopovlju, operacijskom sustavu, programu ili mreži može izazvati ispad bilo kojeg dijela sustava. Ispadi u raspodijeljenom sustavu mogu pokrenuti lanac ispada. Ako ispad ne uzrokuje lanac ispada sustav će i dalje patiti jer će program i dalje zahtijevati računalno vrijeme i memoriju, a s njima neće obavljati koristan posao. U najgorem slučaju ispad može izazvati potpuno zatajenje sustava gdje je jedini lijek iznova pokrenuti sve njegove dijelove. U najboljem slučaju ispad će samo smanjiti učinkovitost sustava. Bez pažljivog nadzora raspodijeljenog sustava teško je otkriti ispad, a još teze otkloniti izvor ispada.

Zadatak nadzora je otkriti ispad i njegov uzrok. [4] ispade dijeli na ispad procesa, pogreške u komunikaciji, vremenske pogreške, pogrešan odgovor i bizantske pogreške. Ako se ispad želi otkriti potrebno je pratiti vrijednosti koje ukazuju da se ispad dogodio. Korisne vrijednosti mogu biti zauzeće memorije, brzina obrade zahtjeva, sadržaj poruke ili duljina uspostave komunikacijskog kanala. Nadzirani program vrijednosti predaje nadzorniku koji je čovjek ili program. Ako nadzor obavlja program on treba biti pouzdaniji od nadziranog programa inače je problem nadzora udvostručen, a ne riješen.

Prvi korak u izradi pouzdanog sustava nadzora je izrada pouzdanog sakupljača vrijednosti. Prije same izrade istražene su ideje, sukobljene su arhitekture i uspoređena postojeća rješenja. Kako bi naglasak na idejama, arhitekturama i izradi rješenja bio podjednak napravljen je sakupljač za jedan raspodijeljeni sustav. Da je napravljen svestran sakupljač koji može sakupljati vrijednosti raznolikog sklopovlja, operacijskih sustava i programa misli o arhitekturi bile bi izražene na uštrp onima o izradi programa. Nadzirani sustav je Apache Kafka, popularni sustav za raspodijeljenu obradu tokova podataka.

2. Sustavi za raspodijeljenu obradu tokova podataka

[1] razlikuje tok podataka (engl. stream) od skupa podataka (engl. batch). Toku podaci dolaze stalno, u nepoznatom redoslijedu, bez znaka kada će prestat dolaziti i podaci će biti odbačeni nakon što budu pročitani. Ovi sustavi su raspodijeljeni jer se tok mora obraditi brzo. Samo neki problemi toka podatka su kartično plaćanje, praćenje korisnika mrežnih stranica, posluživanje reklama i preporuka. Zato sto su problemi raznovrsni razvijeno je mnoštvo alata za obradu tokova. Alate razlikujemo po mogućnostima proizvođača (engl. producer), potrošača (engl. consumer) i posrednika (engl. broker), po načinu slanja i zapisivanju podatak.

Prije razvoja sakupljača vrijednosti potrebno je usporediti i izabrati alat za raspodijeljenu obradu podatak. Kako bi usporedba alata i izvedba sakupljača bila jednostavnija u obzir će se uzeti samo besplatni alati. Umjesto da se usporede svi besplatni alati izabran je predstavnik iz svakog važnog skupa alata:

- Apache Kafka je predstavnik popularnih alata za raspodijeljenu obradu tokova podataka
- Apache Pulsar je predstavnik modernih alata. [2] je objavio Pulsara 2016. dok je [3] objavio Kafku 2011.
- RabbitMQ je predstavnik alat koji guraju podatke prema korisnicima. Za razliku od RabbitMQ iz Kafke i Pulsara korisnici traže podatke

Team je tok poruka. Posluzitelji su posrednici. Proizvodac objavljuje poruke temi. Posrednici zapisuju objavljene poruke. Potrosaci citaju poruke iz pretplacenih tema. Potrosaci koriste pull model. Proizvodaci koriste push model. Proizvodaci mogu objaviti skup poruka odjednom. Pretplacivanje na temu znaci napraviti barem jedan podtok podataka za izabranu temu. Poruke objavljene temi ce biti jednoliko rasporedene u podtokove. Podtok ima iterator poruka. Ako iterator dode do zadnje poruke on se blokira dok se ne objavi nova poruka. Potrosaci mogu zajedno ili neovisno citati poruke.

Tema je podijeljena na pretince. Posrednik brine o barem jednom pretincu.

Pretinac je logicki dnevnik. Dnevnik je napravljen kao skup datoteka iste velicine. Posrednik objavljene poruke doda na kraj poslijednje datoteke. Datoteke se spremaju u stalnu memoriju tek nakon odredenog broja dodavanja ili vremena. Potrosac moze procitati poruku tek nakon sto je ona spremljena u stalnu memoriju. // replicirana koristeci OS -> pogledaj kafka docs -> NE: samo kada je svi zivuci pratitelji imaju zapisanu u svom dnevniku Oznaka poruke je logicki odmak u dnevniku. Odmaci poruka strogo rastu ali nisu uzastopni. Odmak sljedece poruke je zbroj trenutne oznake i duljine poruke. -Svaki pretinac mora stati na server koji je za njega zaduzen. -Pretinci su umnozeni na vise posrednika. -Svaki pretinac ima posrednika vodu dok su svi ostali pratitelji. -Voda pretinca je zaduzen za svako pisanje i citanje pretinca. -U slucaju ispada vode neki pratitelj ce postati novi voda. -Kafka MirrorMaker omogucuju umnazanje cijelog Kafka grozda. +Koristi prirucnu memoriju i tvrdi disk za zapisivanje podataka cim dode poruka, ne radnu memoriju. -> link na rad koji objasnjava vise +Linearno pisanje i citanje na tvrdi disk. +Korsti samo OS pagechache, ne korsti RAM za cache. +Poruke se grupiraju u standardizirani format da se izbjegne cekanje mreze. +Standardizirani format koriste svi: proizvodac, posrednik i potrosac. +Poruke se kopiraju samo iz pagechachea u NIC zahvaljujuci zero-copy i sendfile API. +Podrzava razne formate kompresije podataka.

+Pretinci tema mogu biti replicirani. +Pratitelji vode particija imaju isti dnevnik i odmak kao i voda. +Pratitelji imaju potrosaca koji cita poruke s vode particije i zapisuju poruke u svoj dnevnik. +Posrednik je zivuc ako ima vezu s ZK i ako nije predaleko u citanju poruka s vode (in-sync). +Prepostavlja se da se nece dogoditi bizantski ispad. +Poruka je zapisana samo onda kada ju svi pratitelji imaju zapisanu u svojem dnevniku. +Potrosaci mogu procitati samo one poruke koje su zapisane u sve sinkronizirane replike particije.

Ako potrosac potvrdi odmak potvrdi sve poruke do odmaka. Potrosac od prosrednika trazi broj poruka jednak kolicini podataka i broju poruka nakon zadanog odmaka. Posrednik u indeksu pogleda odmak i nade datoteku s porukama. LIJEPA SLIKA DATOTEKA

Proizvodaci mogu objaviti skup poruka odjednom. -> kopija Potrosaci citaju skup poruka odjednom. -> skob s iteracijom, nije!, jedno je na visokoj jedno na niskoj razini Poruke se pamte samo u prirucnom spremniku za stranice. Stranice se pamte cak kada se Kafka restarta. Minimalno sakupljanje smeca znaci ucinkovita izvedba u VM jeziku. Heuristike stranicenja su ucinkovite jer se poruke citaju uzastopno.

Poruka moze biti procitana vise puta. Da se poruka prenese od datoteke do ko-

munikacijske prikljucnice treba dva poziva za umnazanje i jedan poziv operacijskom sustavu. Zahvaljujuci sendfile API.

Posrednik ne pazi koliko je poruka potrosac procitao. Potrosac pamti koje je poruke procitao. Jednostavan posrednik ali to uzrokuje probleme s brisanjem poruka. Poruke se brisu nakon zadanog vremena. Potrosac se moze vratti unzad i procitati proslu poruku. Zahvaljujuci pull modelu. -Potrosac se brine o odmaku poruka.

Proizvodaci salju poruke na nasumicni ili u odredeni pretinac. Skup potrosaca sastoji se od barem jednog potrosaca. -> mozda stavi u uvod Potrosaci u skupu zajedno citaju poruke iz pretplacenih tema. Svaku poruku cita samo jedan od potrosaca u skupu. Pretinac je najmanja jedinica paralelizma. Poruke u pretincu istovremeno cita samo jedan potrosac u grupi. Broj pretinaca mora biti puno veci od broja potrosaca u najvecem skupu potrosaca. -> ogranicenje -Proizvodaci su zaduzeni za objavu poruke u izabrani pretinac. -> imaju na raspolaganju f() -Potrosaci pripadaju najvise jednoj potrosackoj skupini. -Poruka se dostavlja jednom potrosacu u skupini. -Ista poruka se razasilje razlicitim skupinama potrosaca ako su pretplaceni na istu temu. -Svaki potrosac u grupi cita podjednak broj pretinaca. +Svaki posrednik moze objaviti metapodatke o tome koji posrednik je voda partcije kome proizvodac mora uputiti zahtjev za pisanje. +Proizvodac moze koristiti kljuc da objavi poruku u odredenu particiju.

Kafka koristi Zookeeper-a, pouzdan pruzatelj usluge koncenzusa. U Zookeeper posrednici i potrosaci biljeze promjene, koji potrosac ima pravo citati poruke iz kojeg pretinca i u njega biljeze odmake. -> to se mozda promjenilo Zookeeper podatke umnaza na vise posluzitelja kako se nebi izgubili.

3,3 Kafka dostavlja poruke barem jednom. Vecinu vremena poruka se dostavi tocno jednom. Poruke u pretincu se dostavljaju po redu. Poruke u razlicitim pretincima se ne dostavljaju u redu. -Ako je potreban potpuni redoslijed svih poruka napravi samo jednu particiju. To znaci da samo jedan potrosac moze citati odjednom. +Samo jednom dostava je moguce izvesti vlastitim algoritmom, Kafka Connect, Kafka Stream ili transakcijskim nacinom rada potrosaca/proizvodaca.

4 Poruke nose vrijeme i ime posluzitelja. Proizvodaci broje poslane poruke i javljaju vrijednost. Potrosaci broje procitane poruke i javljaju vrijednost.

-Kolicina spremljenih podataka ne utjece na ucinkovitost. -Podrzava visekorisnicki nacin rada. -Podrzava kvote.

-Osigurava da ako se P1 objavi prije P2 da P1je zapisana u dnevnik prije P2. -Osigurava da potrosaci citaju poruke po vremenu kada su zapisane u dnevnik. -Osigurava da s replikacijom N moze ispasti N-1 posrednika prije gubitka poruke.

-Kod stoga poruke zajedno cita vise potrosaca. -Kada se poruka procita, ona se

brise iz stoga. -Objavi-pretplati salje svaku poruku svim pretplacenim potrosacima. -Skaliranje je nemoguce zbog obujma. -Skup potrosaca omogucuje da svaka tema ima oba dobra svojstva istovremeno. -Pretinac omogucuje paralelno citanje i osigurava redoslijed dostave poruka. -Samo jedan protrosac u grupi moze citati jednu particiju istovremeno. -Broj pretinaca mora biti puno veci od broja potrosaca u najvecem skupu potrosaca. ->kopija -Proizvodac moze cekati potvrdu zapisa poruke.

-Podrzava obradu toka podatak. Kafka Stream.

+Pull model za potrosace podrzava razlicite brzine citanja. +Pull model omogucuje visoku razinu grupiranja poruka. +Problem je kaj mozda posrednik nema podatak za citanje, ali se potrosac moze blokirati dok ne dode odredena kolicina podataka na posrednika.

+Svaka particija ima samo jedan odmak kojim prati do kuda je potrosac procitao poruke. +Poruke zapisane u dnevnik nece biti izgubljene dok god postoji barem jedna ziva replika. +Proizvodaceve poruke ce se isporuciti barem jednom. Duplikate ce Kafka odbaciti. +Proizvodac moze smanjiti zahtjev garancije i ne cekati potvrdu ili reci da zeli cekati samo dok se poruka zapise u particiju vode a ne i slijedbenika. +

STAO NA Replicated Logs: Quorums, ISRs, and State Machines (Oh my!)

+SAmo ISR (in sync replica) mogu postati novi vode particije. +Poruka je zapisana samo kada svi pratitelji particije su zapisali poruku. +Broker moze ponovno postati dio ISR skupa. +Ako sve replike budu unistene prva replika u ISR koja postane ziva ce biti proglasena vodom particije.

+Proizvodaci mogu navesti koliki broj ISR replika zele cekati da potvrde pisanje poruke. +Svaki broker je voda proporcionalnog broja particija. +Kada se dogodi ispad posrednika koji je voda particije umjesto da se glasa za svaku particiju controller posrednik koji ubrza glasanje.

2.1. Apache Kafka

2.2. Apache Pulsar

2.3. RabbitMQ

3. Postojeca rjesenja

4. Arhitektura rjesenja

5. Dizajn rjesenja

6. Izvedba rjesenja

7. Rezultati

8. Zaključak

Zaključak.

LITERATURA

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, i J. Widom. Models and issues in data stream systems. Technical Report 2002-19, Stanford InfoLab, 2002. URL `http://ilpubs.stanford.edu:8090/535/`.

[2] J. Francis i M. Merli. Open-sourcing pulsar, pub-sub messaging at scale, 2016.

[3] N. Kreps, J.and Narkhede i J. Rao. Kafka: a distributed messaging system for log processing. 2011.

[4] I. P. Žarko, K. Pripužić, I. Lovrek, i M. Kušek. *Raspodijeljeni sustavi*, 1.3 izdanju, 2013.

Stvarnovremensko praćenje parametara ispravnosti rada u sustavu za raspodijeljenu obradu tokova podataka

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Real-Time Health Monitoring in Distributed Data Stream Processing System

Abstract

Abstract.

Keywords: Keywords.