

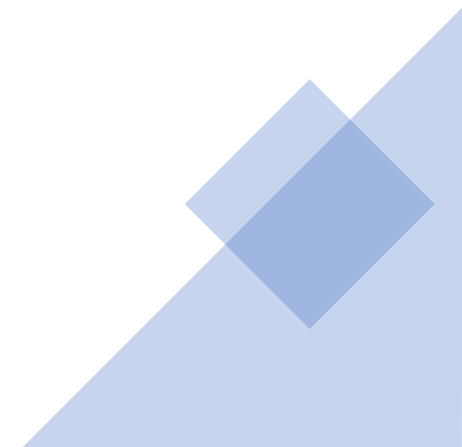


Upravljanje velikim brojem resursa

Računarstvo u oblaku
(Cloud Computing)



Sadržaj

1. Uvod
 2. Mesos
 - A Platform for Fine-Grained Resource Sharing in Data Center by B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. Katz, S. Shenker, I. Stoica from University of California, Berkeley, NSDI 2011.
 3. Omega
 - flexible, scalable schedulers for large compute clusters by Malte Schwarzkopf, Andy Konwinski, Michael Abd-El-Malek and John Wilkes, EuroSys 2013.
- 

Uvod

- Cetri podataka (*Data centers*) se grade od klastera standardnog harvera
- Na tim klasterima se izvr[avaju različite vrste aplikacija: MapReduce, Data Streaming, Storm, S4, etc
- Svaka aplikacija ima sopstveno izvršno okruženje
- Multipleksiranje klastera tako da opslužuju više različitih izvršnih okruženja poboljšava iskorišćenje resursa i smanjuje troškove

Problem upravljanja resursima u velikim klasterima

- Veoma je teško da jedno izvršno okruženje, koje je specifično za određenu vrstu aplikacija, efikasno upravlja resursima na ovakvim velikim klasterima
- Cilj Mesos-a je da omogući izvršavanje različitih okruženja (tipova aplikacija) na jednom klasteru kako bi se omogućilo:
 - Maksimalno iskorišćenje resursa
 - Deljenje podataka između aplikativnih orkuženja

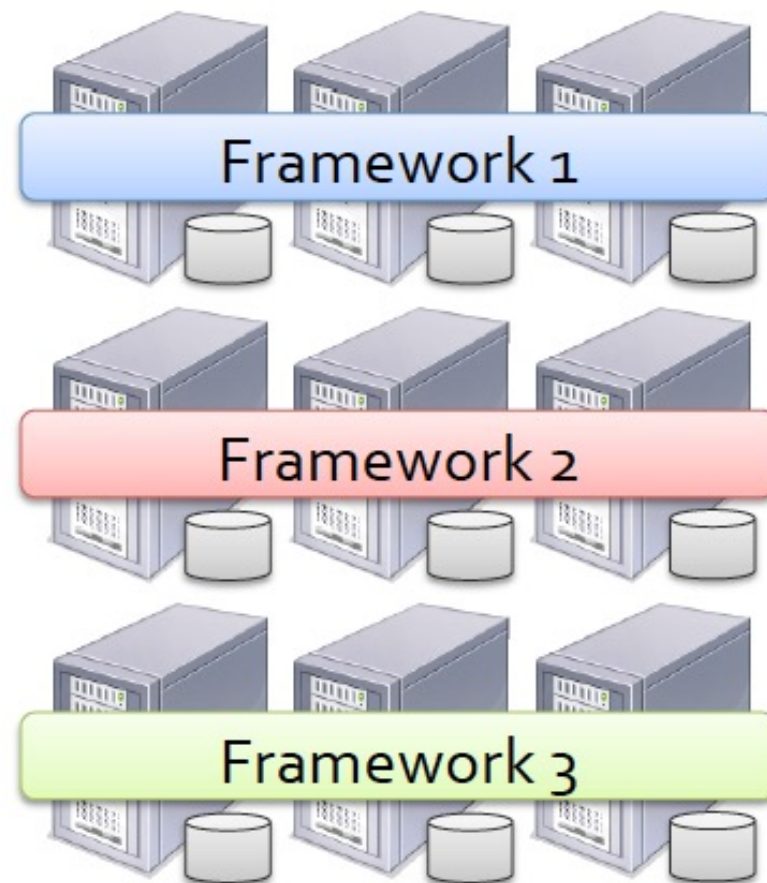
Jednostavno rešenje za deljenje klastera

1. Statistička podela klastera

Jedat tip aplikacije se izvršava na jednoj particiji.

1. Loše iskorišćenje resursa
2. Tvrd rpincip particionisanja ne mora da se poklapa sa promenljivim potrebama aplikacije u realnom vremenu
3. Nepodudarnost između granularnosti statičkih particija i potrebatrenutno aktivnih aplikacija

Coarse-Grained Sharing (HPC):



Storage System (e.g. HDFS)

Mesos

- Ideja Mesos-a je da omogući izvršavanje različitih okruženja (tipova aplikacija) na jednom klasteru
- Izazovi i ciljevi:
 1. **Visoka iskorištenost resursa**
 2. **Podrška za različita okruženja:** svako okruženje može imati različit princip raspoređivanja zadataka
 3. **Skalabilnost:** Sistem raspoređivanja zadataka mora se skalirati da radi sa klasterima od hiljda nodova koji izvršavaju stotine poslova koji se dele na milione pojedinačnih zadataka
 4. **Pouzdanost:** Pošto će sve aplikacije biti zavisne od Mesosa on sam mora biti otporan na otkaze i visoko dostupan
- **Mesos: "tanak" deljeni sloj koji omogućava deljenje resursa na "finom nivou granularnosti" između različitih okruženja** – okruženjima pruža jedinstveni interfejs preko koga one pristupaju resursima u klasteru

Mesos – elementi dizajna sistema

1. Deljenje resursa na finom nivou granularnosti:

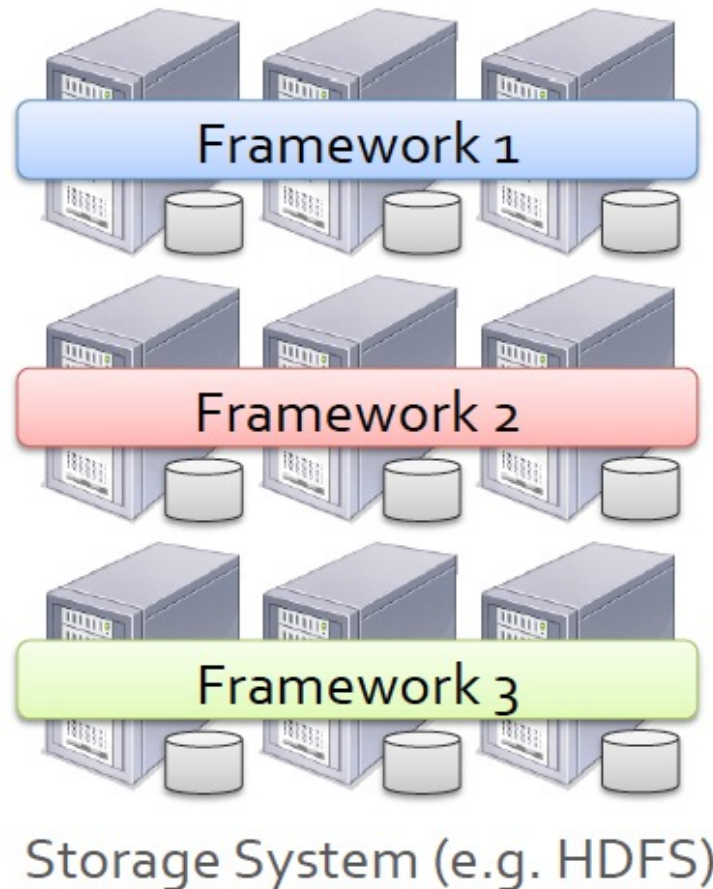
- Alokacija se radi na nivou pojedinačnog zadatka unutar nekog posla
- Poboljšava iskorišćenje resursa, smanjuje latenciju i povećava stepen lokalnosti podataka

2. Nuđenje (ponuda) resursa:

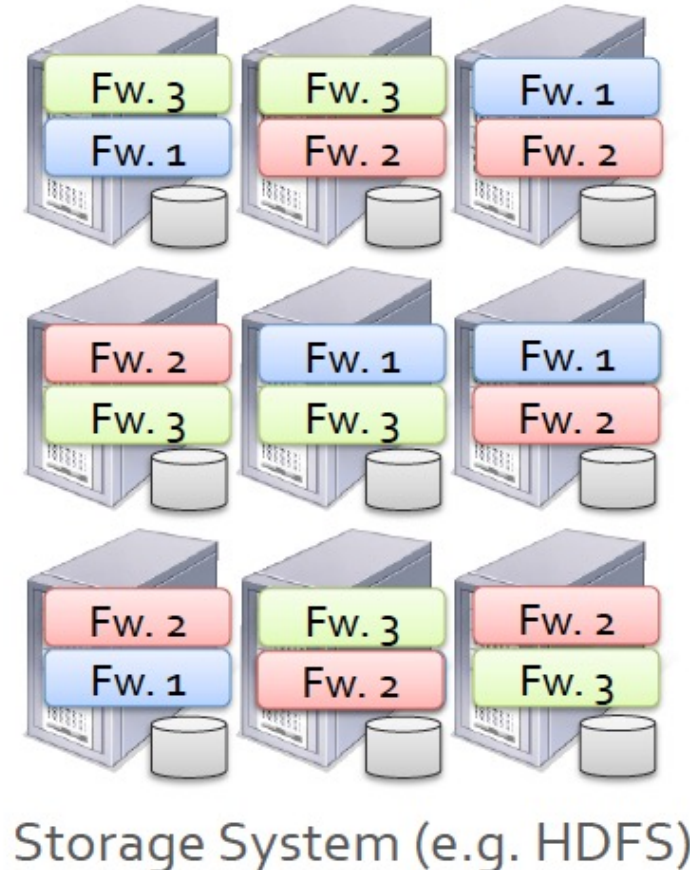
- Ponudi raspoložive resurse izvršnim okruženjima aplikacija
- Jednostavan, skalabilan mehanizam raspoređivanja zadataka koji je upravljivan od strane pojedinačnih aplikacija

Element 1 – fina granulacija deljenja resursa

Coarse-Grained Sharing (HPC):



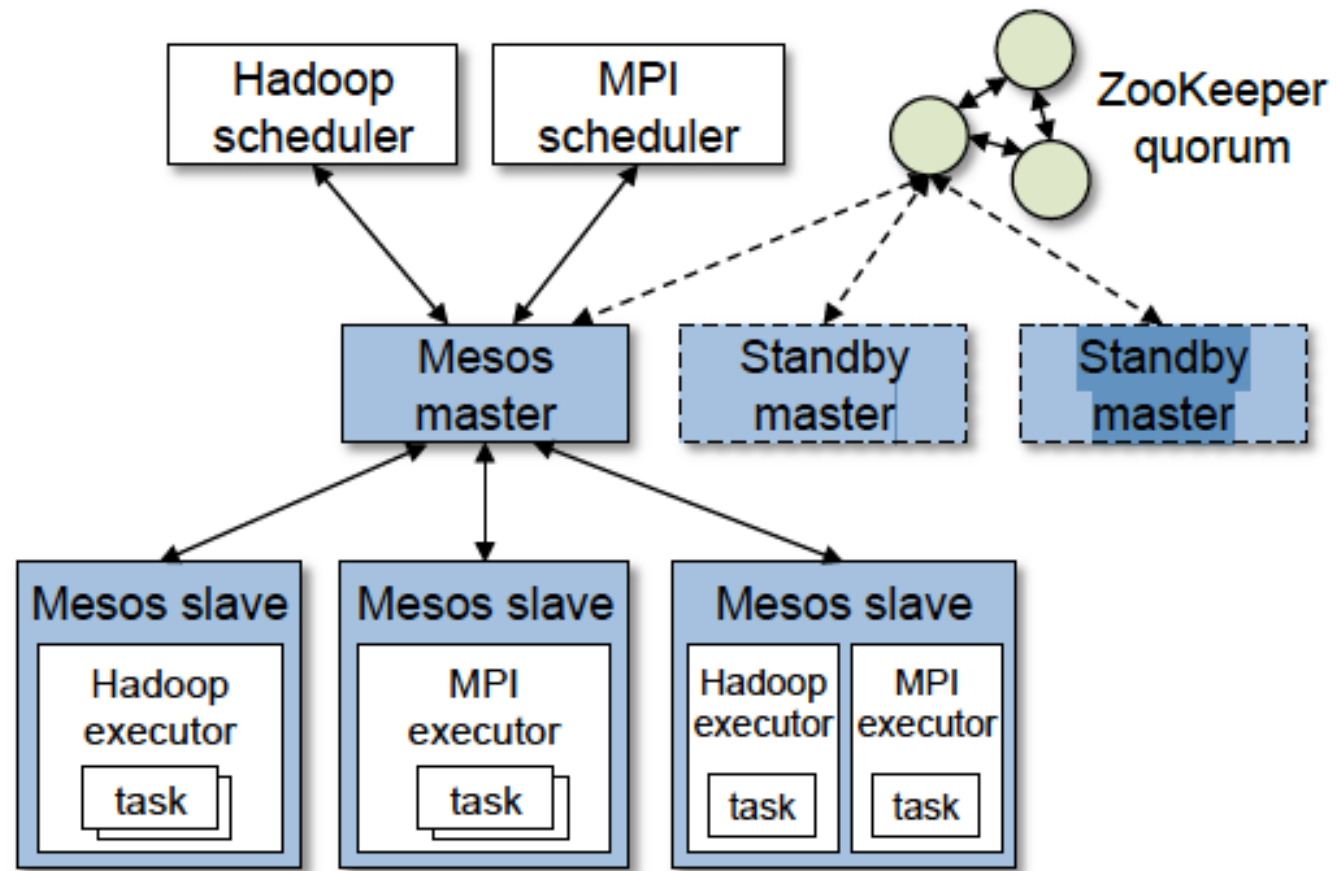
Fine-Grained Sharing (Mesos):



Element 2 – ponuda resursa

1. Mesos predlaže i koristi „nuđenje resursa“:
 - Ponudi raspoložive resurse pojedinačnim okruženjima i pušta ih da same izaberu koji zadatak da pokrenu na pojedinom resursu
2. Prednost: Mesos je u suštini jednostavan i proširiv za buduća okruženja za izvršavanje aplikacija
3. Nedostatak: Kao i bilo koje decentralizovano rešenje, ne nudi optimalne karakteristike

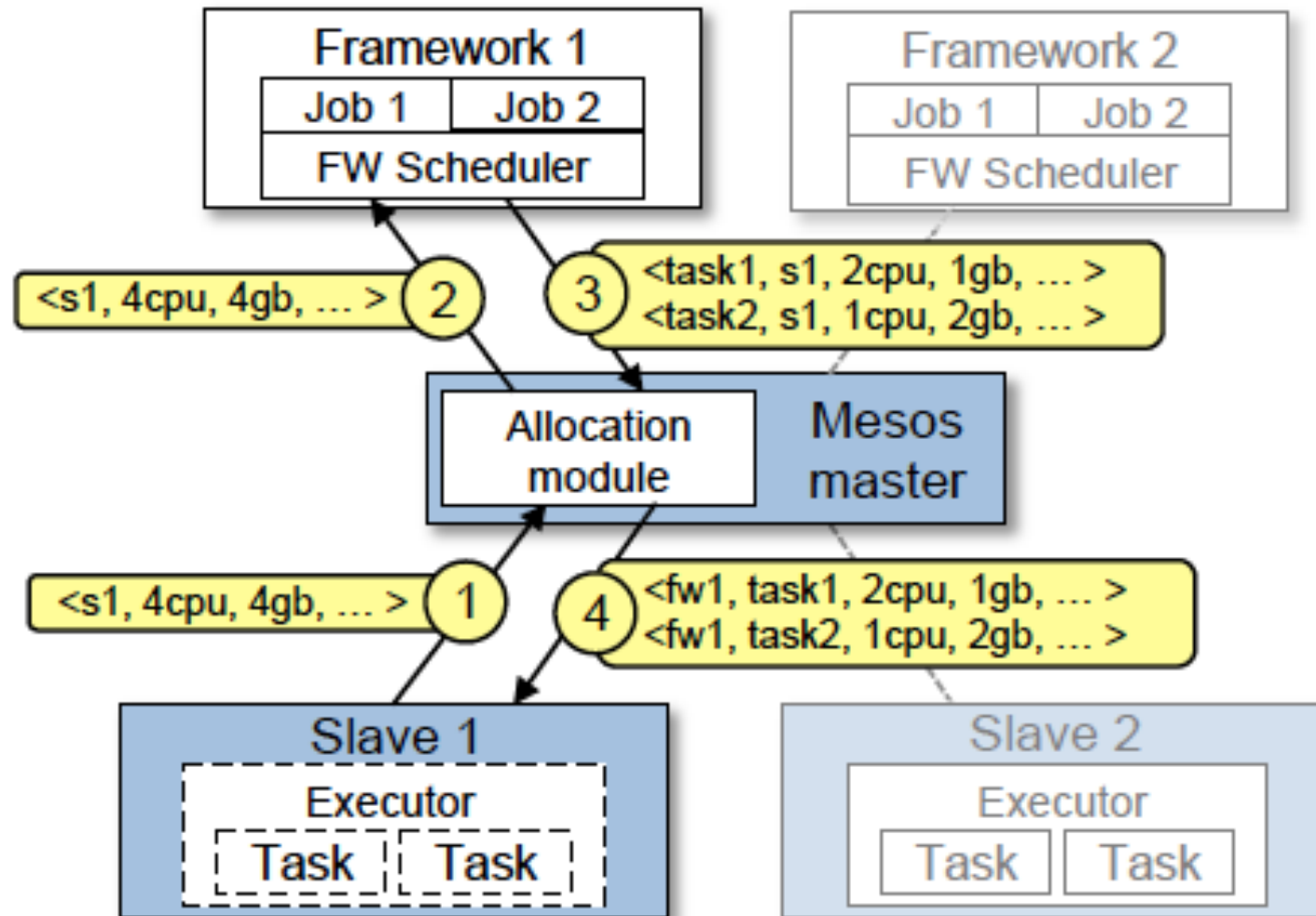
Arhitektura Mesos-a



Arhitektura Mesos-a

- Mesos master: upravlja deljenjem resursa, između različitih okruženje, na finom nivou granularnosti
- Mesos podređeni čvor (slave) na svakom čvoru
- Frameworks - Aplikativno okruženje koje izvršava zadatke pojedinačne aplikacije na svakom podređenom čvoru
- Framework schedulers – aplikativno specifični system za raspoređivanje zadataka
- “Mesos master utvrđuje koliko resursa da ponudi svakom pojedinačnom okruženju, a mehanizmi raspoređivanja pojedinačnih okruženja određuju koji od ponuđenih resursa se koristi (i za šta)”.

Primer ponude resursa



Mesos - Ponuda resursa

- Aplikativno okruženje može odbiti ponuđeni resurs ukoliko on ne zadovoljava postavljene zahteve i može odlučiti da sačeka da odgovarajući resursi postanu dostupni.
- Ovo dovodi do toga da pojedino okruženje može dugo čekati na „prave“ reusurse, a da Mesos u međuvremenu često šalje neodgovarajuće ponude različitim okruženjima – troši se mnogo vremena.
- Okruženja u Mesos-u postavljaju sopstvene filtere tako da specificiraju koje (kakve) ponude resursa će biti uvek odbijane.

Mesos – detalji arhitekture

- Resource Allocation Module

1. Pravično deljenje resursa bazirano na uopštenju max-min pravičnosti pristupa resursima
2. Striktni prioriteti
3. Mesos može i da oduzme resurse (ubije zadatke) kada “pohlepno” (*greedy*) aplikativno okruženje koristi mnogo resursa i zadržava ih za sebe duže vreme. Mesos dozvoljava aplikativnim okruženjima da imaju periode “velike gladi” greedy period (dozvoljava da su povremeno “pohlepne”).

- Izolacija se postiže korišćenjem Linux kontejnera

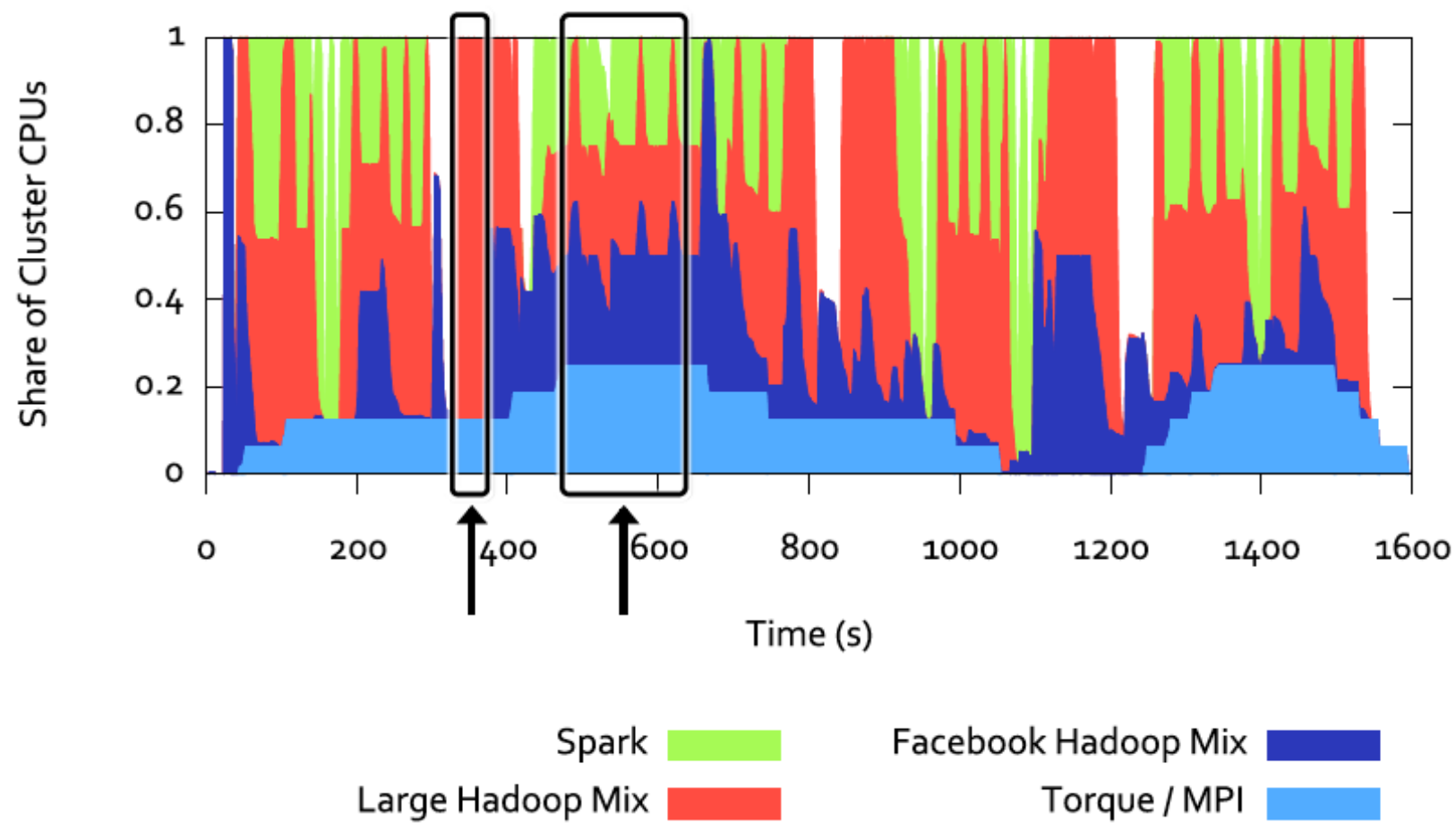
Mesos – detalji arhitekture

- Skalabilna i robusna ponuda resursa
 - Korišćenje filtera: “*nudi samo resurse sa liste L*”, “*nudi samo čvorove koji sadže najmanje R slobodnih resursa*” – ovakva pravila se brzo evaluiraju
 - Mesos broji resurse ponuđene pojedinom okruženju u poređenju sa alokacijom na klasteru
 - Ukoliko pojedinom okruženju treba mnogo vremena da odgovori na ponudu, Mesos je povlači i nudi te resurse nekom drugom okruženju

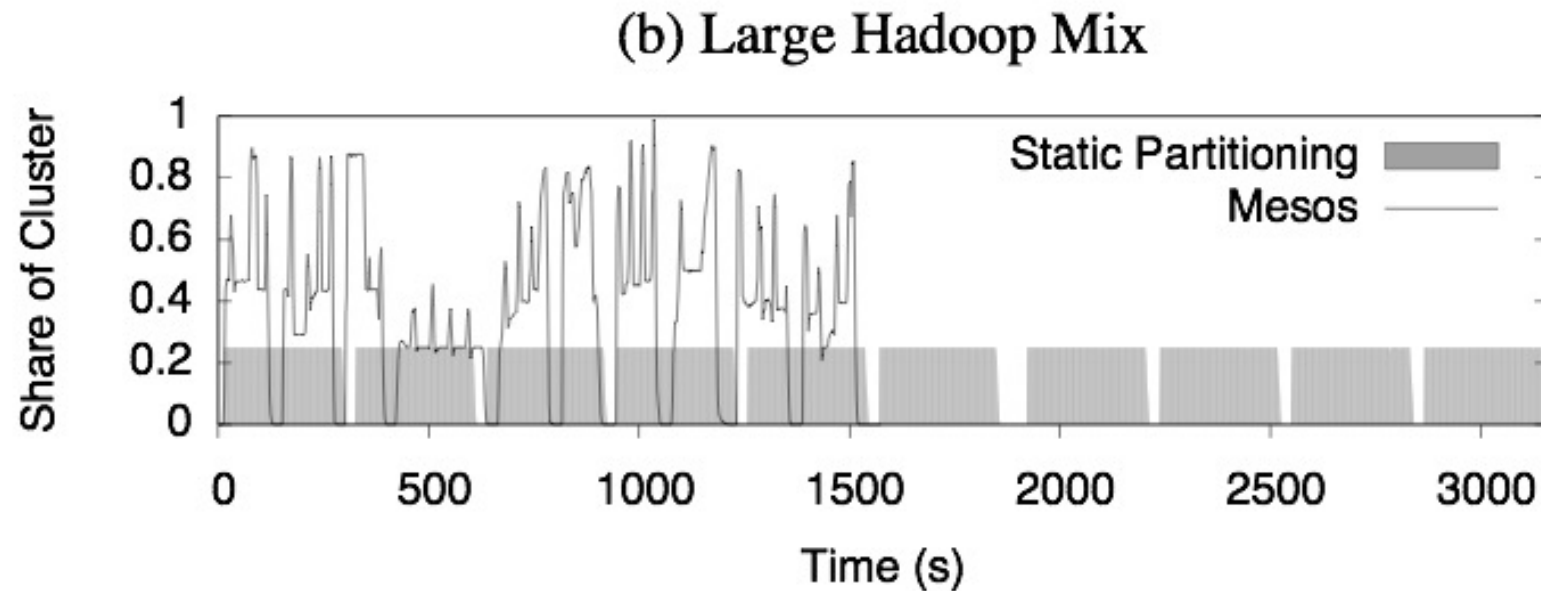
Mesos – evaluacija

- Amazon EC2 sa 92 Mesos čvora
- Miks različitih aplikativnih orkuženja:
 1. Hadoop koji izvršava miks velikih I malih poslova
 2. Jedna Hadoop instance koja izvršava skup velikih *batch* obrada
 3. Spark koji izvršava ML zadatke
 4. Torque koji izvršava MPI zadatke
- 5. Mesos bi trebalo da obezbedi bolje iskorišćenje resursa pri čemu bi pojedinačni poslovi trebalo da se završe minimalno u istom vremenu kao I na statičkim particijama

Mesos – performanse za sva okruženja



Mesos – performanse jednog okruženja poređeno sa statičkim particionisanjem

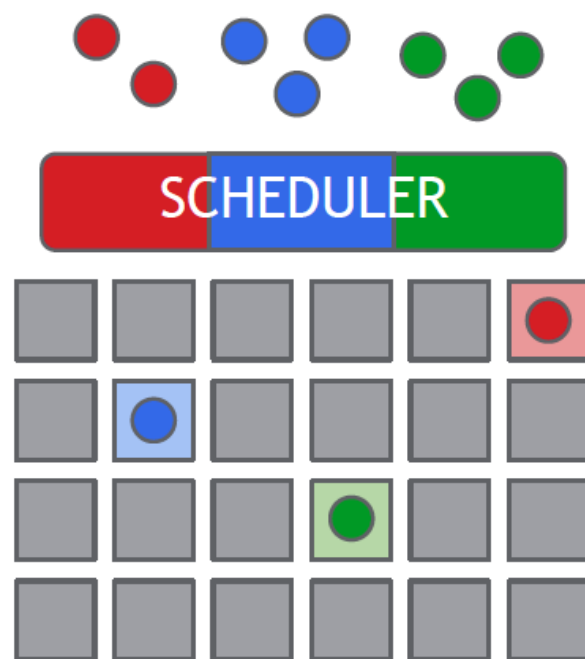


Omega

- Fleksibilni i skalabilni raspoređivač za velike klastere računara
- Izazovi:
 - Google ima različite centre podataka na različitim mestima u svetu
 - Klasteri, ali i količina obrada koja se izvršava (opterećenje) rastu
 - Obrade koje se izvršavaju su raznolike
 - Sve veća brzina pristizanja novih poslova u sistem (*job arrival rate*)
- Nephodan je skalabilan raspoređivač kako bi se prilagodili ovim izazvoima

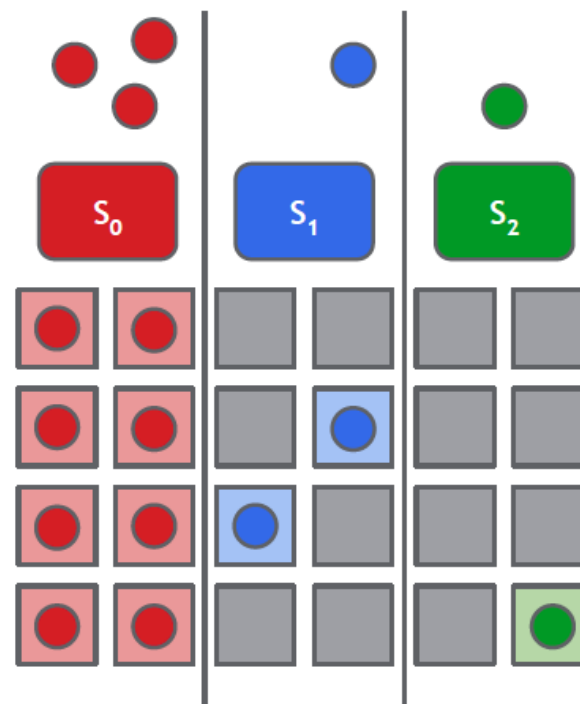
Postojeći pristupi

monolithic scheduler



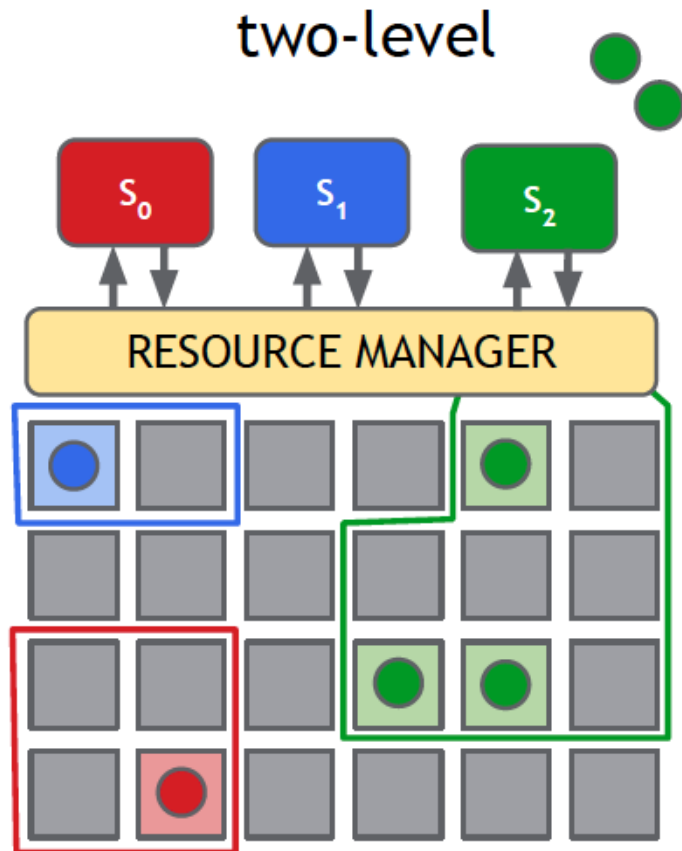
- hard to diversify
- code growth
- scalability bottleneck

static partitioning



- poor utilization
- inflexible

Postojeći pristupi – Mesos ponovo



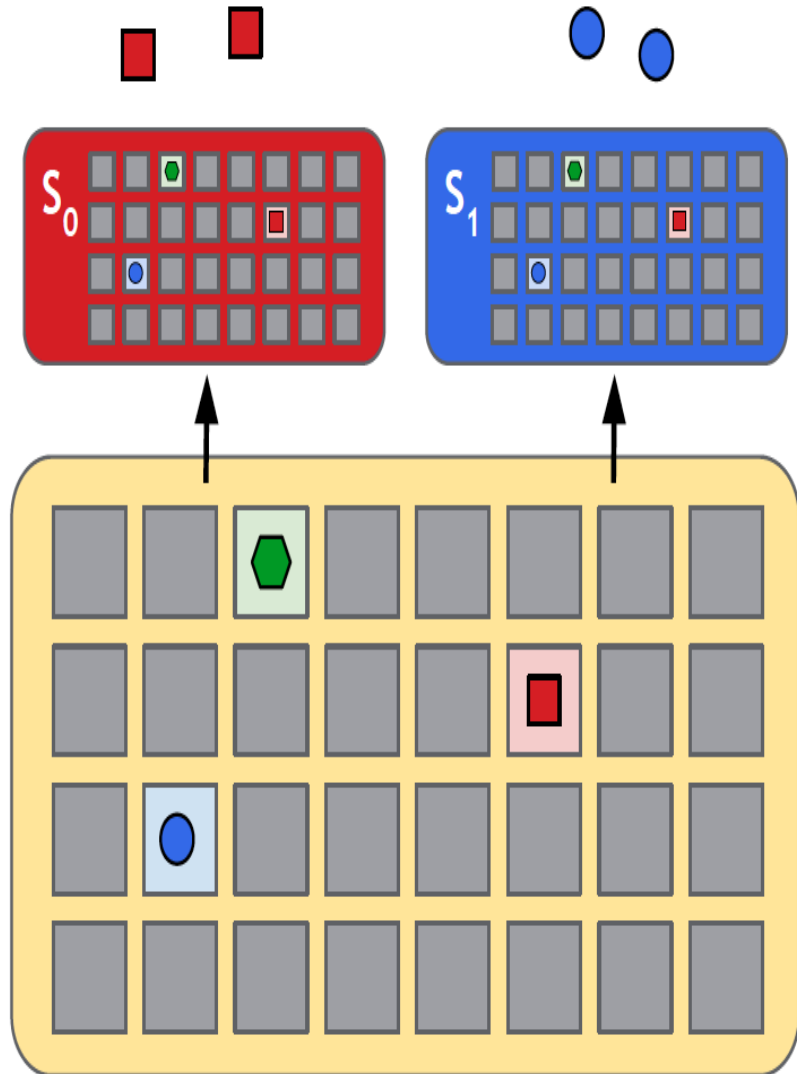
- hoarding
- information hiding

e.g. UCB Mesos [NSDI 2011]

Nedostaci:

- Pesimistično zaključavanje: Mesos izbegava konflikte tako što resurse nudi po jednom okruženju u datom momentu. Mesos bira redosled i veličinu ponude, a okruženje zadržava resurs za vreme odlučivanja → sporo.
- Okruženje nema pristup informacijama o stanju celokupnog klastera. Samim tim ne može osloboditi resurse samostalno, niti zauzeti sve resurse

Omega – deljenje informacija o stanju klastera



Cell State: “otporna” master kopija tabele alokacije resursa klastera

Ne postoji centralizovani raspoređivač (scheduler). Svako okruženje koristi samo svoj raspoređivač, koji ima svoju privatnu, lokalnu i često ažuriranu kopiju stanja klastera – ovu kopiju koristi za donošenje odluka o raspoređivanju zadataka.

Omega – koraci pri raspoređivanju na bazi deljene informacije o stanju klastera

1. Raspoređivač (scheduler) pojedinačnog aplikativnog okruženja donosi odluku
2. U jednom atomičnom upisu ažurira stanje deljene (master) cell state tabele
3. Suštinski, u bilo kom datom momentu, samo jedan od tih upisa može da uspe
4. Raspoređivač posle toga ažurira svoju lokaknu kopiju I ako je potrebno radi ponovo prerapoređivanje

Omega raspoređivanje

- Kod Omega raspoređivači pojedinih aplikativnih okruženja rade u paraleli
- Raspoređivači koriste inkrementalne transakcije da izbegnu “izgladnjivanje”
- Raspoređivači se “dogovaraju” o zajedničkoj skali kojom se iskazuju pririteti pojedinih poslova
- Performanse ovakvog mehanizma raspoređivanja se određuju na bazi broja transakcija koje nisu uspele da se obave I njihovog “troška”
- *“Our performance evaluation of the Omega model using both lightweight simulations with synthetic workloads, and high-fidelity, trace-based simulations of production workloads at Google, shows that optimistic concurrency over shared state is a viable, attractive approach to cluster scheduling.”*

Material

1. Mesos: A Platform for Fine-Grained Resource Sharing in Data Center
by B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. Katz, S. Shenker, I. Stoica from University of California, Berkeley, NSDI 2011.
Sections 1-3.5, 6-6.1.2
2. Omega: flexible, scalable schedulers for large compute clusters
By Malte Schwarzkopf, Andy Konwinski, Michael Abd-El-Malek and John Wilkes, EuroSys 2013.
Sections 1-3, 8