ISHOD 6

1. Osnovne mane NUMA arhitekture za višeprocesorske sustave su:

Kompleksnost programiranja:
Programeri moraju biti svjesni različitih brzina pristupa memoriji kako bi optimizirali performanse, što povećava kompleksnost razvoja softvera.

Neoptimalna izvedba u slučaju nebalansiranih opterećenja:
Ako opterećenje nije ravnomjerno raspoređeno između procesora, neki procesori mogu biti preopterećeni dok drugi ostaju neiskorišteni, što može dovesti do degradacije performansi.

Latencija pristupa memoriji:
Pristup memoriji koja se nalazi na udaljenim čvorovima ima veću latenciju u usporedbi s lokalnom memorijom, što može negativno utjecati na performanse aplikacija koje često pristupaju udaljenoj memoriji.

Kompleksnost hardvera:
NUMA arhitekture zahtijevaju kompleksniji hardver za upravljanje memorijskim pristupima i komunikacijom između čvorova, što može povećati troškove proizvodnje i održavanja.

Skalabilnost:
Iako NUMA arhitekture omogućuju bolje skaliranje u odnosu na UMA (Uniform Memory Access), one ipak imaju ograničenja u skalabilnosti jer povećanje broja čvorova može dodatno povećati latenciju memorijskih pristupa i zahtijevati sofisticiranije upravljanje resursima.

Nedostatak standardizacije:
Postoji manjak standardizacije u NUMA implementacijama, što može otežati portabilnost softvera između različitih NUMA sustava.

2. Reperkusije i rješenje za virtualnu mašinu kojoj je dodijeljeno više radne memorije nego što je procesorov memorijski kontroler može podržati:

Reperkusije

Smanjene performanse:
Kada je VM-u dodijeljeno više memorije nego što procesorov memorijski kontroler može podržati, dolazi do povećane latencije pristupa memoriji. To može značajno usporiti rad VM-a.

Paging:
Operativni sustav može početi koristiti swap prostor na disku, što je mnogo sporije od pristupa fizičkoj memoriji. Ovo može dovesti do drastičnog smanjenja performansi.

Nestabilnost sustava:
Prekomjerna dodjela memorije može uzrokovati nestabilnosti sustava, uključujući rušenja VM-a ili cijelog host sustava.

Degradacija performansi drugih VM-ova:
Ako više VM-ova dijeli iste fizičke resurse, prekomjerna dodjela memorije jednoj VM može negativno utjecati na performanse drugih VM-ova na istom hostu.

Rješenja

Pravilna alokacija memorije:
Dodijelite VM-u samo onoliko memorije koliko je podržano od strane procesorovog memorijskog kontrolera. Ovo može zahtijevati optimizaciju i balansiranje resursa između različitih VM-ova.

Korištenje baloniranja memorije (memory ballooning):
Tehnika baloniranja omogućava hypervisoru da dinamički prilagođava količinu fizičke memorije dostupne VM-u u skladu s trenutnim potrebama. Ovo pomaže u sprečavanju prekomjerne dodjele memorije.

Optimizacija aplikacija i operativnog sustava unutar VM-a:
Optimizirajte aplikacije i operativni sustav unutar VM-a kako bi efikasno koristili dostupnu memoriju. Ovo uključuje podešavanje parametara aplikacija, korištenje cache mehanizama, i smanjenje nepotrebnih procesa.

Upravljanje resursima putem cgroups ili drugih kontrolnih mehanizama:
Korištenje kontrolnih grupa (cgroups) u Linuxu ili sličnih mehanizama u drugim operativnim sustavima može pomoći u postavljanju ograničenja na korištenje memorije, CPU-a i drugih resursa za pojedine VM-ove.

Nadogradnja hardvera:
Ako su VM-ovi kritični za poslovanje i često zahtijevaju više memorije nego što je podržano, razmislite o nadogradnji hardverske infrastrukture kako bi se osigurala adekvatna podrška za sve dodijeljene resurse.

Praćenje i analiza performansi:
Kontinuirano praćenje performansi VM-ova i host sustava može pomoći u ranom otkrivanju problema i omogućiti proaktivno upravljanje resursima. Alati za praćenje kao što su Prometheus, Grafana, ili integrirani alati hypervisora mogu biti korisni.