Prolećni semestar, 2020/21

CS220 - Arhitektura Računara

NUMA

Non-Uniform Memory Access

Uvod

Kako višeprocesorski (višejezgarni) sistemi postaju popularniji i potrebniji paralelizacija je sve neophodnija. Pisanje visoko paralelizovanih programa je daleko od jednostavnog zadatka i zbog toga su mnogi ograničeni kada je u pitanju skalabilnost.

Problem skalabilnosti

Postoji mnogo razloga za nedostatak skalabilnosti paralelnih programa i oni mogu da budu softverski i hardverski. Jedan od glavnih faktora vezanih za performanse paralelnih programa su performanse sistemske memorije odnosno sisema vezanog za sistemsku memoriju. U prošlosti (kraj devedesetih, početak dvehiljaditih) sistemska memorija je fukkcionisala daleko brže od procesora i samim tim multiprocesorski sistemi nisu imali problem sa odzivom memorije. U suštini procesor nije morao ni u jednom trenutku da čeka da mu sistemska memorija bude dostupna da bi učitavao vrednosti iz nje. Danas kod modernih sistema imamo situaciju da je memorija više put sporija od procesora (operativna frekvencija, clock) tako da često dolazi do čekanja odnosno praznog hoda procesora prilikom obavljanja određenih zadataka. Ovo je pogotovo često kod visoko paralelizovanih zadataka gde više procesora (logičkih ili fizičkih) pristupa memorija i mora da čeka da onda postane dostupna. Ovakva situacija odnosno pojava se industriji pominje kao “hitting the memory wall”.

Rešenja

Prefetch

Mnogi sistemi imaju način rešavanja ovih problema. Jedno od rešenja su prefetch sistemi koji analiziraju šablone pristupa memoriji sistema i učitavaju određene delove memorije u keš odakle kasnije kada budu potrebni dostupni su procesoru mnogo brže od standardnog pristupa. Ovo je naravno tačno kada se u pitanju zadaci (workload) koji imaju predvidive pristupe memoriji što naravno često nije slučaj.

Rešavanjem problema “memorijskog zida” takođe se umnogome povećava kompleksnost već kompleksnih sistema koji se iz generacije u generaciju menjaju. Menjanjem ovih sistema se i menja potrebno znanje programera koji piše programe od kojih se očekuju visoke performanse.

Kao što je već navedeno moderni procesori rade na mnogo većim frekvencijama nego operativna memorija i zbog toga imamo najviše problema koji se tiču performansi. Ograničavanje broja pristupa memoriji je ključ performansi modernih računara.

Keširanje

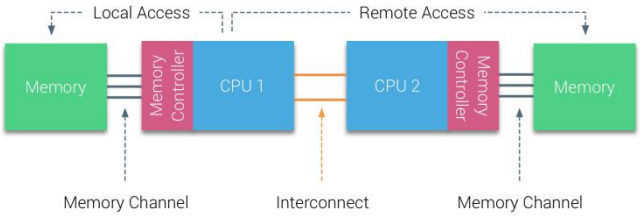
Jedno od rešenja je povećanje brze keš memorije koju procesori imaju i korišćenje sofisticiranih algoritama koji iskorišćavaju ovu keš memoriju što efikasnije. Utilizacija keš memorije u progamima funkcioniše tako što je algoritam napisan an određeni način da pristupa određenim delovima memorije češće i iz tog razloga se taj deo memorije smešta u keš i time se povećavaju performanse. Problem kod ovog pristupa je što kod nekih programa je to jednostavno nemoguće uraditi jer ima previše naizgled nasumičnih pristupa memoriji i zbog toga nataje problem koji se zove “cache miss” odnosno kada u toku izvršavanja nekog programa ze performanse krucijalni delovi memorije bivaju izbačeni iz keša. Takođe povećanje kompleksnosti programa i operativnih sistema umnogome oteževa ovaj zadatak pisanja keš-optimizovanih algoritama i programa. U današnje vreme naravno multiprocesorski sistemi bez ikakve implementacije NUMA arhitekture kontinualno koče više od jednog procesora iz prostog raloga što glavnoj memoriji može da pristupi samo jedan procesor u jednog ciklusu.

Multi-channel memorija

Prvi načini rešavanja ovog problema su implementacija mutlikanalne memorije (dual-channel, triple-channel…) koja omogućava paralelizovaniji pristup memoriji kod višeprocesorskih sistema. NUMA naravno rešava ovaj problem tako što opisuje arhitekturu da svaki procesor ima svoju memoriju kojoj može da pristupa mnogo brže nego memoriji drugih procesora ali po potrebi može da zatraži od njih sadržaje njihovih “lokalnih” memorija. Ovo “prenošenje” podataka se može ostvariti softverski ili hardverski ali u svakom slučaju ono dovodi do usporavanja oba procesora u toku “ne-lokalnog” pristupa memoriji ali generalno to nema toliki uticaj na ukupne performanse sistema. Naravno ovo zavisi umnogome od workload-a koji je na sistemu.

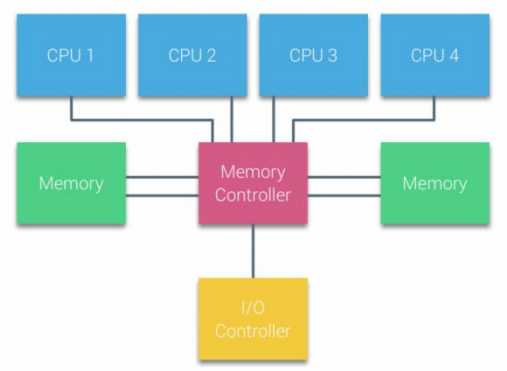
NUMA

Primer evolucije u dizajnu memrijskih sistema je NUMA ili **N**on-**U**niform **M**emory **A**ccess koja arhitektura deljene memorije korišćena u modernim multiprocesorskim sistemima. Po ovoj arhitekturi svaki procesor ima svoju memoriju odnosno svoj kontroler memorije i mora da ima mogućnost pristupa memoriji drugog procesora. Ovaj pristup memoriji drugog procesora se često ostvarijuje preko izuzetno performantnog interkonekta (interconnect - IC) (Slika 1). Lokalna memorija procesora pruža veliki protok (bandwidth) i brz odziv (latency) dok pristup memoriji preko navdenog interkonekta drugih procesora , odnosno NUMA domena, pruža manji protok i sporiji odziv. NUMA arhitektura je logična posledica skaliranja SMP (symmetric multiprocessing) arhitektura.



Slika 1 Dijagram NUMA arhitekture

UMA



Slika 2 Dijagram UMA arhitekture

Za razliku od NUMA-e, UMA tj. Uniform Memory Access arhitektura ima identično vreme pristupa za sve procesore. Ali kao što na slici možemo da vidimo ovde nema mnogo prostora za paralelizaciju iz prostog razloga što svaki od fizičkih ili logičkih procesora koriste isti kontroler memorije i samim tim limitira pristup memoriji.

Kod UMA sistema procesori su povezani preko front-side bus-a za Northbridge. Northbridge zadrži kontroler memorije i sva komunikacija mora da prođe kroz njega. Takođe I/O kontroler je povezan za Northbridge tako da bilo koje I/O operacije moraju da prođu kroz njega da bi došle do procesora. Kao što smo pomenuli ovde su načini optimizacije bili dodavanje kanala komunikacija između Northbridge-a i ostalih komponenti. Brzine komponenti koje su tradicionalno bile na ovaj način povezane sa procesorom daleko premašiji fizičke kapacitete ove arhitekture i ukazuju na njen nedostatak skalabilnosti.

Razvoj

NUMA je bila komercijalno razvijana tokom 90-ih godina od strane lidera HPC industrije (Unisys, HP, Honeywell, IBM, Compaq i dr.). Prva komercijalna implementacija NUMA arhitekture je bila XPS-100 serija servera koju je dizajnirao Honeywell. Danas većina modernih operativnih sistema (Linux, Windows, BSD i dr.) podržava NUMA arhitekturu.

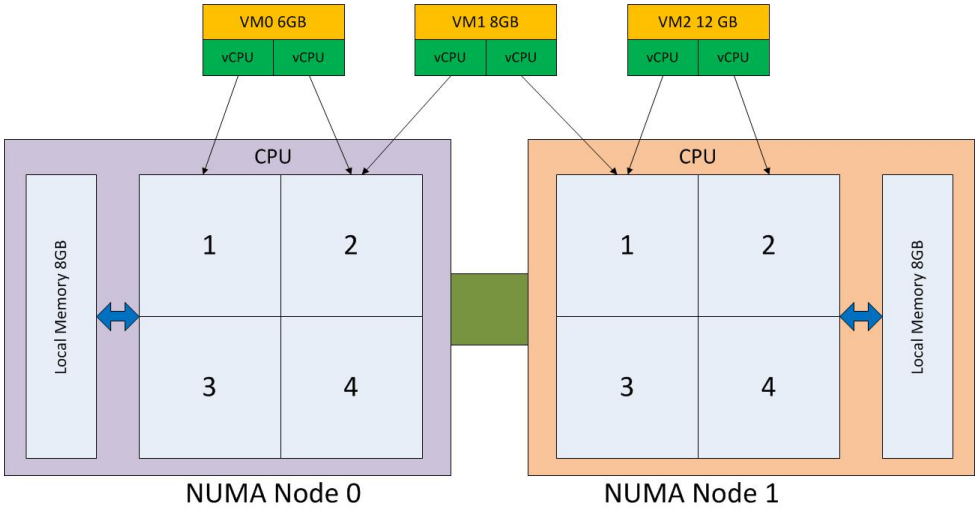
Kao što smo već pomenuli NUMA odstupa od tradicionalnog dizajna sistema i alocira memoriju posebno za svaki procesor. Jedan od sistema koji su doveli do popularizacije

ovog tipa arhitekture je bio Cray Origin 2000 superkompjuter krajem devedesetih godina prošlog veka.

AMD implementacija NUMA-e došla je sa Opteron procesorom (2003.) koji je koristio HyperTransport za svoj interkonekt između procesora. Intel je svoju NUMA implementaciju objavio sa svojom serijom Itanium servera 2007. godine i njihov interconekt se zvao QuickPath.

Virtuelizacija

Zbog činjenice da je u današnjem vremenu veoma česta virtuelizacija važno je naglasiti da kada su u pitanju NUMA sistemi moramo pažljivo alocirati procesorske resurse i rasporediti ih po virtuelnim mašinama. Prilikom kreiranja virtuelnih mašina biramo koje procesore (bilo fizičke ili logilčke) želimo da alociramo kojoj virtuelnoj mašini. Zbog toga što se tajna performansi NUMA sistema zasniva na lokalizovanom pristupu memoriji veoma bi poželjno ako ne i obavezno u mnogim slučajevima alocirati procesore iz istog NUMA domena jednoj virtuelnoj mašini (Slika 2).



Slika 3 Virtuelizacija na NUMA sistemu

Ovde vidimo loš primer alociranja procesora po virtuelnim mašinama. Virtuelne mašine VM0 i VM2 imaju adekvatno alocirane procesore koji se svi nalaze i istom NUMA domenu dok VM1 po jedno od svojih jezgara uzima iz svakog od NUMA domena. Ovakvim pristupom će se umnogome smanjiti performanse te virtuelne mašine jer će neminovno dolaziti do već opisanih zahteva za “remote” memorijom koji su daleko sporiji od “lokalne” memorije. Što se tiče alociranja memorije ovde takođe vidimo da postoji problem sa VM2 koji ima kapacitet memorije koji prevazilazi kapacitet “njenog” NUMA domena što dovodi do fragmentacije memorije i pojave zahteva za “remote” memorijom. O ovom kontekstu tih 4 od tih 12GB će se neminovno naći u drugom NUMA domenu i pozivi ka toj memoriji će biti daleko sporiji.

Multithreading

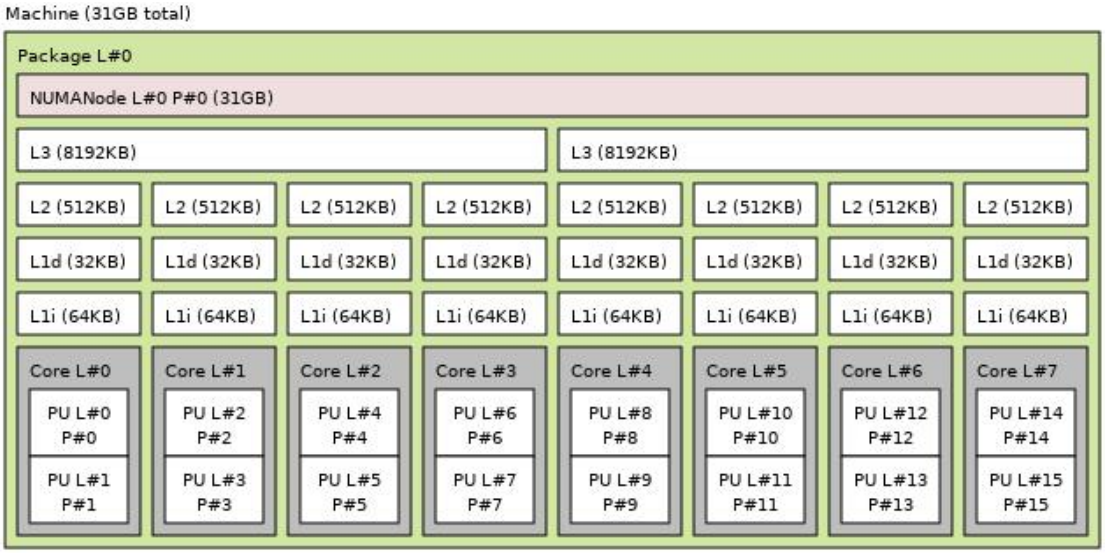
Još jedan od problema NUMA arhitekture je činjenica da operativni sistem može da u bilo kom trenutku premesti nit sa jednog procesora na drugi. Poenta ovog premeštanja je da bi se ravnomernije rasporadio workload po procesorima. Ovaj proces nije poželjan na numa sistemima jer u slučaju premeštanja workload-a sa jednog NUMA domena na drugi odvajaju se podaci od mesta gde su potrebni i zahteva se od procesora da vrši “remote” pristup memoriji. Ovaj problem se može rešiti mehanizmom koji se zove “pining”. Na ovaj način se sprečava migracija niti sa jednog procesora na drugi.

Interconnect

Interkonekt kao što smo pomenuli predstavlja direktnu vezu između dva fizička procesora koja služi za razmenu informacija veoma velikom brzinom. Prelaskom sa UMA na NUMA arhitekturu došlo je do uklanjanja Front-side bus-a (FSB) i do direktne veze procesora sa ostalim kontrolerima. Ovo omogućava veliko povećanje brzine. Na primer poslendnji FSB koji se koristio u pre je imao brzinu od 1.6 GT/s (giga-transfera u sekundi) odnosno 12.8 GB/s. Trenutne brzine Intelovog QuickPath-a dostižu brzine od 25.6 GB/s.

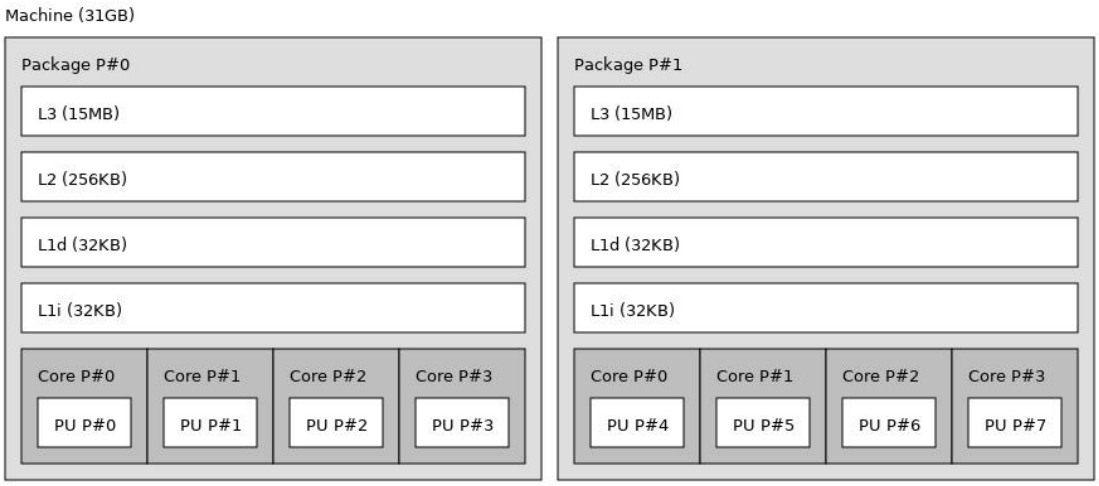
Topologija

Generalno NUMA tehnologija iako prisutna nije bila od preterano velikog značaja za desktop računare sve do pojave AMD Ryzen procesora. Oni kao odgovarajući Intel procesori imaju jedan NUMA node/domen koji je vezan za taj procesor.



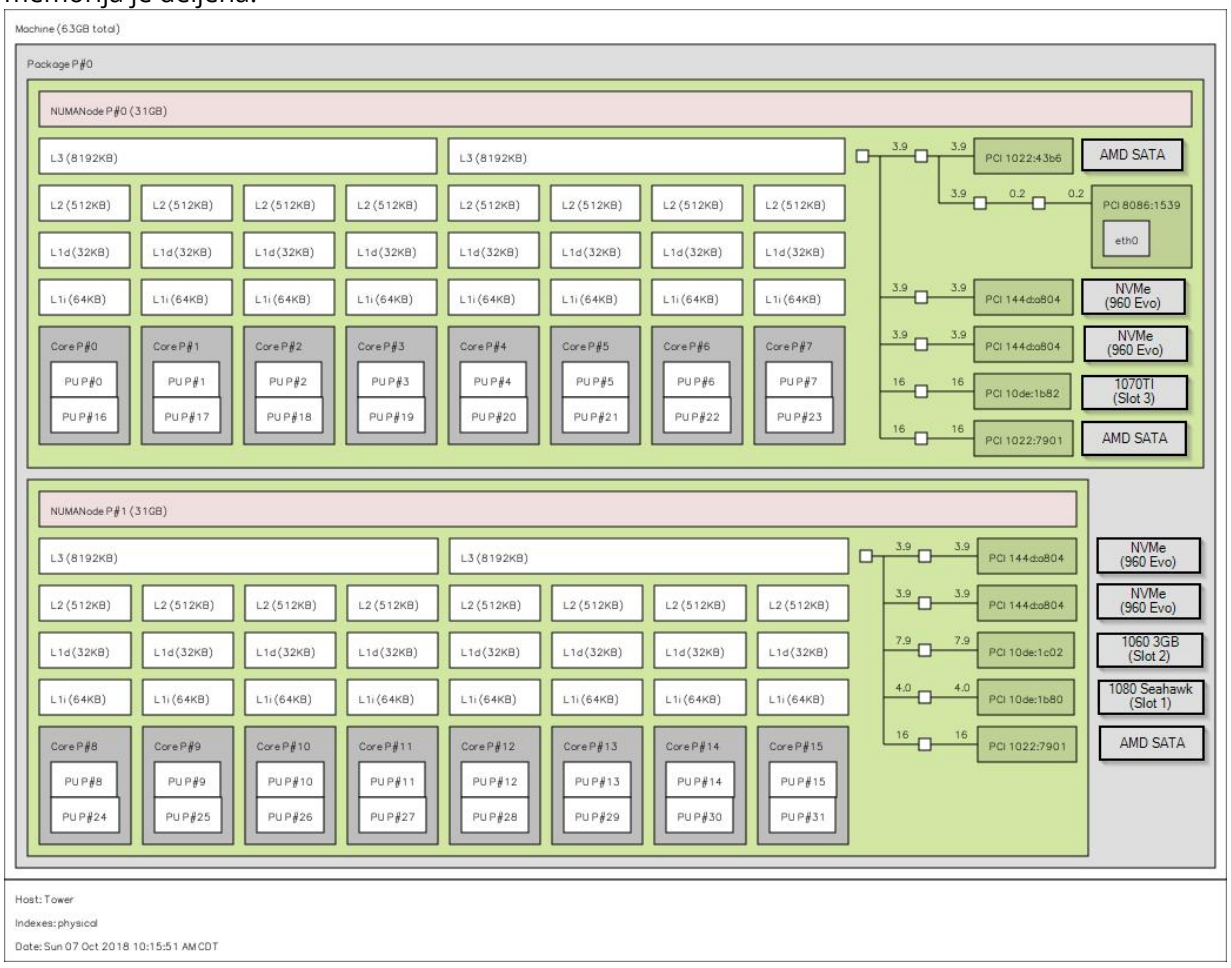
Slika 4 Topolgija Ryzen 7

Na slici iznad vidimo primer topologije Ryzen 7 procesora sa svojim jedim NUMA domenom i 32GB sistemske memorije. Na ovom primeru takođe vidimo da Ryzen procesori imaju deljen L3 keš. Korišćen program za prikaz topologije je **lstopo** na Linux sistemu.



Slika 5 UMA Topologija

Na slici iznad vidimo topolgiju starije serverske mašine koja nema NUMA domen. Radi se o dva fizička quad-core procesora koji oba rade po UMA arhitekturi. Njihova memorija je deljena.



Slika 6 Threadripper Topolgija

Na slici iznad vidimo topologiju Ryzen Threadripper procesora koji je ima dva NUMA domena. Ova dva domena su zapravo dva odvojena procesora u istom pakovanju povezana interkonektom koji AMD zove Infinity Fabric. Na ovaj način tehnički imamo jedan procesor ali naravno imamo mogućnost korišćenja oba NUMA domena. Zanimljivost kod na primer ovakvih konfiguracija je to da možemo specifične aplikacije koje koriste PCIe uređeaje na jednom NUMA domenu alocirati isključivo na taj domen kroz softversku konfiguraciju. Primer bi bilo igranje igara. Sa slike vidimo da su grafičke kartice povezane na NUMA node P#1 što znači da kada grafička karitica želi da komunicira sa procesorom to može da radi direktno u slučaju da aplikacija alocirana na isti NUMA node i podaci ne moraju da prolaze kroz interkonekt. Takođe kod Windows operativnog sistema scheduler uvek daje veći prioritet jezgru 0 prvog procesora tako da raspored PCIe uređaja se može kreirati i sa tim na umu. Ovakve optimizacije daju veoma malo promene u performansama ali mogu biti veoma korisne za kritične sisteme. Threadripper nije naravno jedini desktop procesor sa više od jednog NUMA domena. Neki od Intelovih Xeon procesora imaju više od jednog NUMA domena ali njihova namena nije za desktop računare tako da i samim tim je dostupnost veoma mala.

Reference

Non-uniform memory access, Wikipedia, <https://en.wikipedia.org/wiki/Non-uniform_memory_access>

NUMA Deep Dive, frankdenneman.nl, Frank Denneman, <https://frankdenneman.nl/2016/07/07/numa-deep-dive-part-1-uma-numa>

NUMA, HPC Wiki, <https://hpc-wiki.info/hpc/NUMA>

Binding/Pinning, HPC WIki, <https://hpc-wiki.info/hpc/Binding/Pinning>

What is NUMA, YouTube, Level1Techs <https://www.youtube.com/watch?v=M-Q02b5uvfY>