HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Cluster sustavi

CCERT-PUBDOC-2006-12-176

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument, koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Cluster sustav je skupina računala koja tijesno surađuju u izvršavanju određenih zadataka tako da funkcioniraju kao jedno računalo. Čvorovi *cluster* sustava često su povezani brzom LAN mrežom, a namjena im je povećati performanse i/ili dostupnost u usporedbi s jednim računalom koje izvršava iste zadatke. Pri tome su ovi sustavi daleko jeftiniji od pojedinačnih računala usporedivih performansi.

Cluster sustavi mogu se sastojati od samo dva poslužitelja od kojih jedan aktivno odgovara na klijentske zahtjeve dok drugi pasivno čeka kvar prvoga kako bi preuzeo njegovu ulogu. Ovo je najjednostavniji primjer *cluster* sustava koji povećava raspoloživost usluge. Ako bi dva spomenuta poslužitelja preuzimala klijentske zahtjeve u ovisnosti o vlastitim i raspoloživim resursima drugog čvora, osim povećanja raspoloživosti takav sustav implementirao bi upravljanje opterećenjem. U slučaju da se *cluster* sustav koristi za provođenje matematičkih proračuna koje je moguće paralelizirati, dva poslužitelja bi proračun izvršila dvostruko brže od jednog računala i tada bi oni činili *cluster* sustav visokih performansi.

Ovo su jednostavni primjeri, a stvarne implementacije *cluster* sustava mogu uključivati do nekoliko tisuća računala, poslužitelje smještene na suprotnim stranama svijeta ili složene algoritme za raspodjeljivanje i migraciju zadataka među čvorovima.

U nastavku dokumenta opisan je povijesni razvoj *cluster* sustava, zahtjevi koji su postavljeni pred *cluster* sustave, klasifikacije *cluster* sustava prema načinu obrade mrežnog prometa i prema namjeni, poznatije implementacije *cluster* sustava te prednosti i nedostaci korištenja *cluster* sustava.

2. Povijesni razvoj *cluster* sustava

Prve *cluster* (hrv. grozd) sustave izgradili su korisnici kojima je za izvršavanje radnih zadataka bilo potrebno više od jednog računala ili koji su imali potrebu stvaranja pričuvnih kopija. Prvi takvi sustavi javljaju se sredinom prošlog stoljeća.

Načela na kojima se temelje *cluster* sustavi prvi puta formalno je oblikovao IBM-ov inženjer Gene Amdahl 1967. godine u članku naslova „*Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities*". U ovom članku matematički je opisao ubrzanje koje je moguće postići razlaganjem zadatka na više elemenata i njihovim izvođenjem na računalnom sustavu paralelne arhitekture te je time postavio, po njemu nazvan, Amdahalov zakon na kojem se temelje *cluster* sustavi visoke dostupnosti (eng. HA – *High Availability*) i višeprocesorsko računarstvo.

Rani razvoj *cluster* sustava usko je vezan uz razvoj računalnih mreža. Zbog toga su osmišljavanje računalne mreže s razmjenom paketa (eng. *packet switching network*) od strane tvrtke RAND 1962. godine i implementacija takve mreže 1969. godine pod nazivom ARPANET, koja se kasnije razvila u Internet, ključni događaji u povijesti razvoja *cluster* sustava.

Tijekom ranih 1970ih razvoj *cluster* sustava teče usporedno s razvojem računalnih mreža, obilježenim TCP/IP i Xerox PARC projektima, i s razvojem Unix operacijskog sustava. 1971. godine razvijen je Hydra operacijski sustav namijenjen *cluster* sustavu DEC PDP-11 mikroračunala, ali tek oko 1983. godine su *cluster* sustavi postali široko dostupni uslijed definiranja protokola i alata za udaljenu raspodjelu poslova i dijeljenje datoteka, uglavnom u sklopu razvoja BSD Unix operacijskog sustava.

Prvi komercijalni *cluster* sustav ARCnet razvila je 1977. godine tvrtka Datapoint. Ovaj sustav nije zaživio na tržištu, a prvi uspješan *cluster* sustav razvila je 1983. godine tvrtka *DEC*. Radi se o VAXcluster sustavu namijenjenom VAX/VMS operacijskom sustavu. Oba navedena *cluster* sustava pored paralelnog izvođenja zadataka podržavaju dijeljene datotečne sustave i vanjske jedinice. Dva značajna *cluster* sustava razvijena su 1994. godine: Tandem Himalaya sustav visoke raspoloživosti i IBM S/390 Parallel Sysplex sustav namijenjen komercijalnoj uporabi.

Značajnu ulogu u razvoju *cluster* sustava imao je PVM (eng. *Parallel Virtual Machine*) programski sustav otvorenog koda namijenjen stvaranju *cluster* sustava temeljenih na TCP/IP komunikaciji. Na ovom modelu i uz pomoć rasta raširenosti jeftinih PC računala uskoro su izgrađeni *cluster* sustavi koji su po broju operacija u sekundi (eng. FLOPS – *Floating Point Operations Per Second*) nadišli najnaprednija superračunala. Time je potaknut i razvoj *grid* računarstva.

3. Zahtjevi postavljeni pred *cluster* sustave

Ovisno o primjeni, *cluster* sustavi moraju zadovoljiti neke ili sve od sljedećih zahtjeva:

- nadogradivost,
- dostupnost,
- cjelovitost sustava,
- brza komunikacija,
- programibilnost i
- primjenivost.

3.1. Nadogradivost

Cluster sustav treba imati mogućnost dinamičke i transparentne nadogradnje. To znači da dodavanje ili uklanjanja čvora iz *cluster* sustava ne smije biti vidljivo korisniku niti smije utjecati na izvođenje aplikacija.

3.2. Dostupnost

Ako pojedini čvor *cluster* sustava postane privremeno ili trajno nedostupan, na primjer zbog sklopovskog kvara, njegove zadatke preuzimaju ostali dostupni čvorovi. Zahtjevi se automatski prosljeđuju aktivnim poslužiteljima tako da korisnik i aplikacije koje se izvode na sustavu ne zamjećuju kvar.

3.3. Cjelovitost sustava

Cjelovitost sustava (eng. SSI – *Single System Image*) je predodžba, stvorena na programskoj ili sklopovskoj razini, kojom se skupina resursa predstavlja kao jedan snažniji resurs. Svaki *cluster* sustav koji ima ovo svojstvo, korisniku, aplikacijama i računalnoj mreži djeluje kao jedno računalo.

3.4. Brza komunikacija

Čvorovi *cluster* sustava moraju biti povezani brzom računalnom mrežom i koristiti komunikacijske protokole koji omogućuju brzu komunikaciju. U suprotnom, komunikacija između čvorova postaje usko grlo sustava.

3.5. Programibilnost

Jednostavno programsko sučelje (eng. API – *Application Programming Interface*) olakšava stvaranje učinkovitih aplikacija namijenjenih *cluster* sustavima. Ono ujedno skriva arhitekturu pojedinog *cluster* sustava od programera i tako omogućuje stvaranje prenosivih aplikacija.

3.6. Primjenjivost

Tijekom izgradnje *cluster* sustava potrebno je izbjegavati dizajn koji bi mogao ograničiti njegovu primjenu i time osigurati primjenjivost na širokom spektru problema. Aplikacije namijenjene izvođenju na *cluster* sustavima, ali i ostale aplikacije, trebale bi se znatno brže izvoditi na takvom sustavu.

4. Podjela *cluster* sustava prema načinu obrade mrežnog prometa

U sklopu nastojanja da se standardiziraju programske biblioteke koje omogućuju paralelno rješavanje zadataka razvijena su dva sustava za paralelno programiranje: *PVM* (eng. *Parallel Virtual Machines*) i *MPI* (eng. *Message Parsing Interface*).

4.1. *Parallel Virtual Machines*

PVM sustav namijenjen je heterogenim *cluster* sustavima. Heterogeni *cluster* sustavi sastavljeni su od različitih računala i dijele se u dvije skupine: sustavi sastavljeni od raznorodnih računala (npr. SUN SPARCStation IPX, DEC Alpha i PC računala) i sustavi sastavljeni od istorodnih računala različitih performansi (npr. PC računala s procesorima različitih generacija).
PVM je besplatna, prenosiva, programska biblioteka za obradu poruka. Podržava jednoprocesorska i SMP (eng. *Symmetric Multiprocessor*) računala, *cluster* sustave sastavljene od računala s Unix/Linux i Windows operacijskim sustavima, a omogućuje čak i organiziranje raznorodnih računala povezanih putem Interneta u *cluster* sustav. Obrada poruka od strane PVM biblioteke unosi dodatno opterećenje na mrežni promet (eng. *overhead*).
Ovaj sustav omogućuje podešavanje kontrolne radne stanice za raspodjeljivanje zadatka ostalim čvorovima *cluster* sustava te stvaranje virtualnog okruženja za izvođenje paralelnih aplikacija koje omogućuje njihovu prenosivost između različitih sklopovskih platformi.

4.2. *Message Parsing Interface*

MPI sučelje namijenjeno je homogenim *cluster* sustavima. To su sustavi sastavljeni od identičnih čvorova: jednakih računala s identičnim matičnim pločama, procesorima, memorijom, diskovima i mrežnim karticama. Osim kod *cluster* sustava ovo sučelje koristi se i na klasičnim superračunalima (npr. Cray T3G i IBM SP2I).
Dvije najpopularnije implementacije MPI sučelja su:

- LAM (eng. *Local Area Multicomputer*) - potpuna implementacija MPI standarda koja omogućuje pokretanje aplikacija na pojedinim računalima ili na skupinama računala povezanih računalnom mrežom temeljenom na UDP/TCP komunikacijskim protokolima.
- MPICH (eng. *MPI CHameleon*) - implementacija MPI standarda s visokim stupnjem prenosivosti, slična LAM sustavu. Glavna odlika ovog sustava je podržavanje MPICH ADI (eng.

Abstract Device Interface) sučelja koje omogućuje jednostavno i učinkovito prenošenje sustava između različitih sklopovskih platformi.

5. Podjela *cluster* sustava prema namjeni

Cluster sustave je po namjeni moguće podijeliti u tri skupine:

- sustavi visoke dostupnosti (eng. HA - *High Availability*),
- sustavi za raspoređivanje opterećenja (eng. LB - *Load Balancing*) i
- sustavi visokih performansi (eng. HP - *High Performance*).

5.1. *Cluster* sustavi visoke dostupnosti

Cluster sustavi visoke dostupnosti (eng. HA - *High Availability*) namijenjeni su osiguravanju stalne dostupnosti određene usluge. To se postiže korištenjem redundantnih čvorova koji preuzimaju ulogu nedostupnih elemenata sustava.

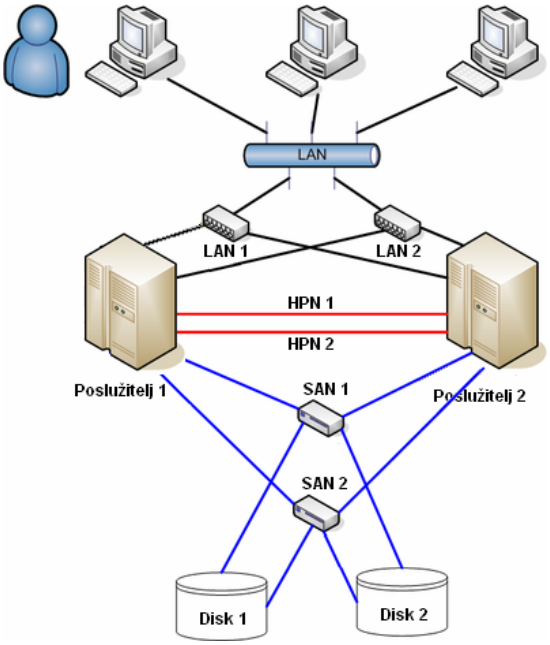
U slučaju rušenja poslužitelja, aplikacija koja se na njemu izvodi najčešće prestaje biti dostupna korisnicima dok netko ne ukloni kvar koji je uzrokovao rušenje. Kod HA *cluster* sustava sklopovski i programski kvarovi se otkrivaju i aplikacija se automatski pokreće na pričuvnom računalu (*Slika 1*). Prije pokretanja aplikacija programska podrška *cluster* sustava može, u slučaju potrebe, konfigurirati pričuvni čvor. Npr. dohvatiti odgovarajući datotečni sustav, podesiti mrežno sklopovlje ili pokrenuti određene pomoćne aplikacije.

Tijekom izgradnje HA *cluster* sustava potrebno je ukloniti ranjive točke ugradnjom višestrukih mrežnih veza i višestrukih spremišta podataka povezanih višestrukim SAN (eng. *Storage Area Network*) mrežnim vezama.

HA *cluster* sustavi često posjeduju unutrašnju HPN (eng. *Heartbeat Private Network*) privatnu mrežu za nadzor stanja sustava. Ako se dogodi kvar ove mreže dok čvorovi HA *cluster* sustava ispravno funkcioniraju, tzv. '*split-brain*' situacija, postoji opasnost da svaki čvor pretpostavi kako su svi ostali u kvaru i pokuša preuzeti njihove zadatke. Višestruko izvođenje istih zadataka može uzrokovati oštećenje spremljenih podataka i zbog toga je potrebno implementirati nadzor ispravnosti HPN mreže.

HA *cluster* sustavi koriste se za izvođenje sustava za upravljanje bazama podataka, poslovnih aplikacija, web stranice tvrtki koje posluju putem Interneta te za dijeljenje datoteka na računalnoj mreži. Svaki zadatak se izvodi samo na jednom računalu tako da je brzina izvođenja ograničena performansama pojedinog čvora.

Najčešće sadrže samo dva čvora, a jako su rijetke implementacije s više od osam računala. Sva računala u ovakvom sustavu moraju imati jednake kartice za pristup zajedničkom dijeljenom diskovnom podsustavu, jednake mrežne kartice te identične operacijske sustave.

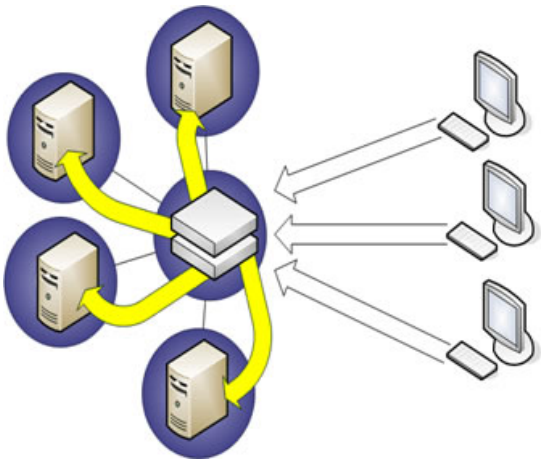


Slika 1: Dijagram HA *cluster* sustava s dva čvora

5.2. *Cluster* sustavi za raspoređivanje opterećenja

Cluster sustavi za raspoređivanje opterećenja (eng. LB - *Load Balancing*) omogućuju raspodjeljivanje zadataka između više računala sa ciljem optimalnog iskorištavanja raspoloživih resursa i skraćivanja vremena izvođenja.

Virtualni poslužitelj, sa svojom IP adresom i portom, predstavlja sučelje LB *cluster* sustava prema korisniku. Kada klijent pošalje zahtjev virtualnom poslužitelju, virtualni poslužitelj odabire jednog od fizičkih poslužitelja s kojima je povezan i prosljeđuje mu spomenuti zahtjev (Slika 2).



Slika 2: Shematski prikaz LB *cluster* sustava

Očuvanje podataka nastalih izvođenjem aplikacije nakon njezina završetka (eng. *persistence*), također se podešava na virtualnom poslužitelju. Zbog opasnosti od kvara pojedinih čvorova *cluster* sustava, ovakvi podaci najčešće se pohranjuju u dijeljenu bazu podataka kojoj pristup imaju svi fizički poslužitelji.

LB *cluster* sustavi mogu se koristiti kao:

- poslužitelji za priručne podatke (eng. *cache server*),

- vatrozidi,
- tzv. farme poslužitelja,
- sustavi za uočavanje neovlaštenog pristupa i
- implementacija raznih protokola, npr. TCP, UDP, HTTP, FTP, SSL, SSL BRIDGE, SSL TCP, NNTP, SIP ili DNS protokola.

Raspoređivanjem čvorova LB *cluster* sustava preko većih zemljopisnih udaljenosti postiže se otpornost sustava na nepogode koje mogu intenzivno i/ili trajno pogoditi određeno područje i onemogućiti rad većeg broja čvorova, kao što su prirodne katastrofe i teroristički napadi. Ovakvi sustavi nazivaju se GSLB (eng. *Global Server Load Balancing*) sustavima.
Kod ovih sustava, kao i kod HA *cluster* sustava, sav posao vezan uz određeni korisnikov zahtjev izvodi se na jednom računalu.

5.2.1. Algoritmi raspoređivanja zahtjeva

LB *cluster* sustavi za raspoređivanje zahtjeva po poslužiteljima, koriste neki od sljedećih algoritama:

- *least connections* – prosljeđivanje zahtjeva poslužitelju s najmanjim brojem aktivnih mrežnih veza,
- *round robin* – koristi se kod DNS poslužitelja kada se na zahtjeve za pristup računalu koje posjeduje više IP adresa, odgovara vraćanjem različitih adresa,
- *least response time* – procjenjuje se vrijeme potrebno za izvršavanje zadatka na raspoloživim poslužiteljima i odabire se najbrži,
- *least bandwidth* i *least packets* – odabire se poslužitelj s najmanjom potrošnjom mrežnih resursa,
- *token* – zadaci se poslužiteljima raspoređuju kružno,
- *URL hashing* – poslužitelj se odabire na temelju tzv. 'hash' vrijednosti izračunate iz cijele URL adrese ili iz njezinog dijela,
- *domain name hashing* – poslužitelj se odabire na temelju tzv. 'hash' vrijednosti izračunate iz imena domene,
- *source IP address* – zahtjev se prosljeđuje poslužitelju ovisno o IP adresi pošiljaoca,
- *destination IP address* – zahtjev se prosljeđuje poslužitelju ovisno o IP adresi na koju je poslan,
- *source IP – destination* – kombinacija prethodna dva algoritma,
- *RTT (Round Trip Time)* – koristi se kod GSLB sustava, a odnosi se na odabiranje poslužitelja s najmanjim kružnim kašnjenjem, odnosno poslužitelja s najkraćim vremenom putovanja korisničkog zahtjeva i poslužiteljevog odgovora na isti zahtjev,
- *static proximity* i *network proximity* – algoritmi koji se također koriste kod GSLB sustava.

5.3. *Cluster* sustavi visokih performansi

Cluster sustavi visokih performansi (eng. HP - *High Performance*) koriste se za rješavanje problema koje je moguće razložiti na više manjih problema koji se zatim paralelno rješavaju na više računala. Najčešće su to složeni matematički proračuni kojima se na ovaj način višestruko skraćuje vrijeme izvođenja.
Za postavljanje HP *cluster* sustava potrebna su minimalno dva računala, a uobičajene su implementacije od nekoliko stotina do nekoliko tisuća čvorova. Najčešće su to jednaka računala na istoj lokaciji povezana brzom mrežom s Unix/Linux operacijskim sustavima, ali prisutne su i implementacije s MacOS ili Windows Compute Cluster Server 2003 operacijskim sustavima.

5.3.1. *Grid* sustavi

Pod nazivom *grid* podrazumijeva se niz tehnologija koje omogućuju povezivanje nezavisnih računala i *cluster* sustava putem računalne mreže. *Grid* sustavi imaju namjenu jednaku onoj HP *cluster* sustava, a osnovna razlika je u načinu upravljanja resursima. *Grid* sustavi funkcioniraju u različitim administrativnim domenama i građeni su od heterogenih sustava s različitim mrežama, različitom programskom podrškom za *cluster* sustave, različitim operacijskim sustavima, itd.

Koriste se za probleme presložene za pojedine HP *cluster* sustave kao što su simulacije potresa, modeliranje savijanja proteina, predviđanje vremenskih uvjeta i sl. *Grid* sustavi se prema namjeni dijele na računske (eng. *Computational Grid*), podatkovne (eng. *Data Grid*) i sustave opće namjene (eng. *General Purpose Grid*).

5.3.2. Paralelizam izvođenja

Paralelno izvođenje je postupak ubrzavanja izvođenja programskog zadatka njegovim razlaganjem na više manjih dijelova koji se zatim simultano izvode na većem broju procesora. Izvođenjem zadatka na *N* procesora moguće je njegovo izvođenje ubrzati maksimalno *N* puta u odnosu na izvođenje na jednom procesoru.

Paralelizam je moguće postići na više razina:

- na programskoj razini i
- na sklopovskoj razini unutar koje je izvršavanje zadataka moguće paralelizirati:
  - na razini procesora i
  - na razini računalnih sustava.

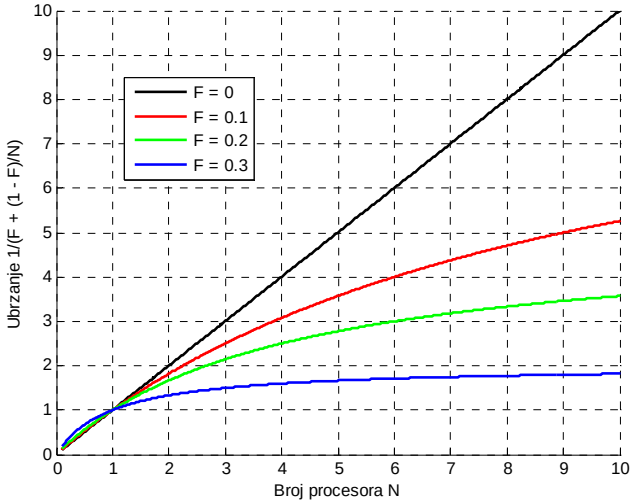
Paralelizam je moguće ostvariti na više načina koji se razlikuju po učinkovitosti primjene na određene vrste problema:

- SMP (eng. *Symmetric Multiprocessor*) – svaki procesor ima pristup sistemskoj memoriji i svim vanjskim jedinicama, podržavaju višeprocesorske i višedretvene aplikacije,
- NUMA (eng. *Non-Uniform Memory Access*) – u ovim sustavima razlikuje se spora dijeljena memorija i brza memorija pridijeljena pojedinim procesorima,
- UMA (eng. *Uniform Memory Access*) – svi procesori imaju jednak prioritet pristupa memoriji,
- SIMD (eng. *Single Instruction Multiple Data*) – svi procesori izvode jednak programski kod, ali nad različitim podacima,
- MIMD (eng. *Multiple Instruction Multiple Data*) – ovakvi sustavi sastoje se od samostalnih računala koja, najčešće pomoću posebnih programskih biblioteka, surađuju u rješavanju zadataka.

Amdahlov zakon formalizira ubrzanje u izvođenju programskih zadataka upotrebom većeg broja procesora. Ako je *F* sekvencijalni dio zadatka (udio programskog zadatka koji nije moguće paralelizirati) i *(1-F)* dio zadatka koji je moguće paralelizirati onda je maksimalno ubrzanje ostvarivo pomoću *N* procesora:

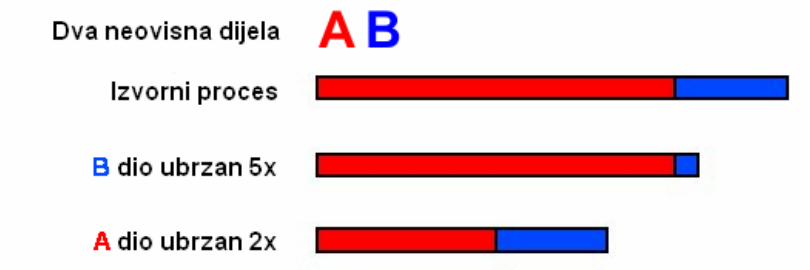
$$\frac{1}{F + \frac{(1-F)}{N}}$$

Ako *N* teži u beskonačno, maksimalno moguće ubrzanje iznosi *1/F*. U praksi se omjer cijene i performansi značajno smanjuje s povećanjem broj procesora kada omjer *(1 - F)/N* postane malen u usporedbi s udjelom *F*. Zbog toga paralelno rješavanje programskih zadataka ima smisla u slučaju korištenja manjeg broja procesora ili kod zadataka s malim udjelom *F*.



Slika 3: Ubrzanje izvođenja programskog zadatka u ovisnosti o udjelu sekvencijalnog dijela

Osim uvođenja većeg broja procesora, značajno ubrzanje moguće je postići pažljivim odabirom dijela zadatka za optimiziranje. Ako se programski zadatak sastoji od dva neovisna sekvencijalna dijela (Slika 4) mnogo veće ubrzanje moguće je postići optimiziranjem dijela zadatka čije izvođenje duže traje i koji predstavlja usko grlo paralelnog izvođenja cjelokupnog procesa.



Slika 4: Ubrzanje izvođenja programskog zadatka ovisno o dijelu koji je optimiziran

6. *Cluster* sustavi za bazne mrežne servise

Cluster sustavi koji implementiraju bazne mrežne servise jednostavni su za implementaciju i ne postavljaju visoke zahtjeve na upotrijebljeno sklopovlje. To su sustavi koji objedinjuju funkcionalne karakteristike HA i LB *cluster* sustava.

6.1. DNS

DNS (eng. *Domain Name System*) je sustav za upravljanje imenima domena, a glavna zadaća mu je njihovo prevođenje u IP adrese. Dva DNS poslužitelja s aktivnim repliciranjem podataka čine DNS *cluster* sustav. Ako je jedan poslužitelj nedostupan klijente opslužuje drugi dostupni poslužitelj, a ako se žele povećati performanse sustava moguće je podesiti svaki od poslužitelja da odgovara na upite polovice klijenata. U slučaju kvara jednog od poslužitelja smanjuje se funkcionalnost jer tada nije moguće dodavati nove zapise.

6.2. DHCP

DHCP (eng. *Dynamic Host Configuration Protocol*) je protokol za automatsko dodjeljivanje IP adresa. Implementiranje ovog protokola pomoću *cluster* sustava s dva čvora zahtjeva korištenje dvostruko veće adresne sheme od one potrebne u slučaju korištenja jednog poslužitelja. Podešavanjem svakog poslužitelja tako da dodjeljuje polovicu adresa, postiže se raspodjela klijenata te se time ujedno povećavaju performanse sustava.

6.3. AD i DFS

AD (eng. *Active Directory*) je Microsoftova implementacija LDAP (eng. *Lightweight Directory Access Protocol*) protokola namijenjena Windows operacijskim sustavima. *Cluster* sustav koji implementira AD sustav mora imati minimalno dva poslužitelja za svaku domenu. Podaci o korisnicima i raspoloživim mrežnim servisima umnažaju se među poslužiteljima.
Redundantnost dijeljenja datoteka postiže se korištenjem *Microsoft Cluster Server* poslužitelja ili DFS (eng. *Distributed File System*) sustava. Klijentski zahtjevi se u DFS *cluster* sustavima ravnomjerno raspodjeljuju među poslužiteljima, a period umnažanja podataka među njima je izvorno podešen na 1.8 sekundi i moguće ga je podešavati. Pri tome treba imati u vidu dodatno opterećenje mrežnih resursa koje donosi učestalo umnažanje podataka.

6.4. NIS

NIS (eng. *Network Information Service*) je servis za rukovanje podacima o korisnicima i mrežnim resursima tvrtke *Sun Microsystems*. Ovaj sustav omogućuje raspodjeljivanje zadataka među svim umreženim računalima pretvarajući tako svaku računalnu mrežu na kojoj je aktivan u *cluster* sustav.

7. Implementacije *cluster* sustava

Cluster sustave moguće je implementirati na Windows i Linux platformama. Windows *cluster* sustavi temelje se na Windows Server 2003 operacijskom sustavu.
Programska podrška namijenjena Linux *cluster* sustavima može se podijeliti na operacijske sustave, međuprograme namijenjene isključivo *cluster* sustavima i one općenite namjene, te cijeli niz popratnih aplikacija. Uz opise Windows rješenja, u nastavku ovog poglavlja ukratko su opisana tri sustava koji omogućuju izgradnju Linux *cluster* sustava.

7.1. Windows Server 2003 Clustering

Mogućnosti stvaranja *cluster* sustava pomoću Windows 2003 Server Clustering sustava temelje se na dvije tehnologije: SC (eng. *Server Cluster*) i NLB (eng. *Network Load Balancing*).

7.1.1. *Server Cluster*

Server Cluster (SC) je unaprijeđena inačica MCS (eng. *Microsoft Cluster Server*) sustava dostupna u *Advanced* i *Datacenter* inačicama Windows 2003 Server operacijskog sustava. Podržava *cluster* sustave s dva do osam čvorova i koristi se najčešće za izvođenje sustava za upravljanje bazama podataka, servisa za elektroničku poštu, LOB (eng. *Line Of Business*) ili vlastitih aplikacija.
Većina aplikacija se izvodi samo na jednom čvoru *Server Cluster* sustava, dok pričuvni čvor čeka ispad aktivnog čvora kako bi preuzeo izvođenje aplikacija. Zbog veće iskoristivosti sklopovskih resursa moguće je istovremeno izvoditi više aplikacija, a neke se mogu, kao npr. *Microsoft SQL Server*, izvoditi paralelno na više čvorova.
Čvorovi *Server Cluster* sustava koriste tzv. kvorum za određivanje vlasništva nad aplikacijama. Kvorum predstavlja podatkovni medij koji mora biti pod kontrolom čvora vlasnika aplikacije. Kvorumom može istovremeno upravljati samo jedan čvor, a ako on postane nedostupan pričuvni čvor preuzima upravljanje kvorumom i vlasništvo nad aplikacijom.
Ako su svi čvorovi povezani sa samo jednim podatkovnim medijem na kojemu je stvoren kvorum, višestruko je pojednostavljeno migriranje aplikacije između čvorova. Nedostatak ovakvog *cluster* sustava proizlazi iz mogućnosti kvara podatkovnog medija ili SAN mreže. Korištenjem redundancije moguće je ukloniti ovaj nedostatak, ali i onda ostaje ograničenje takvog *cluster* sustava na međusobno bliske čvorove što ih čini ranjivima na prirodne katastrofe ili dulji nestanak električne energije.
Kod MNS (eng. *Majority Node Set*) *Server Cluster* sustava poslužitelji kvorume spremaju na lokalne podatkovne medije. Zbog toga u slučaju migracije aplikacije pričuvni čvor mora imati kopiju podataka smještenih u kvorumu što se postiže repliciranjem tih podataka putem računalne mreže. Čvorovi ovakvog *cluster* sustava moraju biti povezani računalnom mrežom koja može biti tipa LAN, WAN ili VPN čime je omogućeno njihovo raspoređivanje na većim zemljopisnim udaljenostima. MNS *cluster* sustav

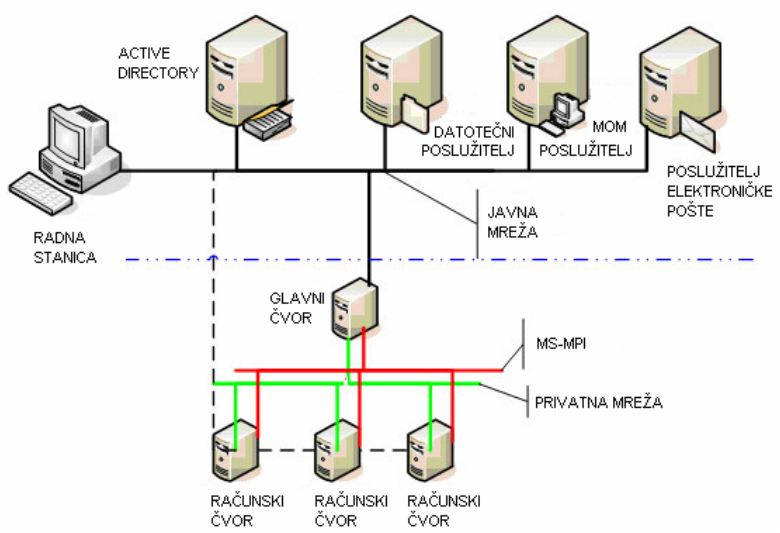
mora imati najmanje tri čvora te više od pola čvorova mora biti dostupno kako bi sustav ispravno funkcionirao.

7.1.2. *Network Load Balancing*

Network Load Balancing (NLB) sustav koristi se za web servise i poslužitelje, te za vatrozide, a dostupan je kod svih inačica *Windows 2003 Server* operacijskih sustava. Ne koristi kvorume pa nema posebnih zahtjeva na podatkovne medije, a podržava sustave s do 32 čvora. U slučaju nedostupnosti pojedinog čvora, zahtjevi se automatski prosljeđuju aktivnim čvorovima. Prije isključivanja čvora zbog redovnog održavanja moguće mu je naložiti izvršavanje preuzetih zadataka.

7.2. Windows Compute Cluster Server 2003

Windows Compute Cluster Server 2003 (CCS) omogućuje implementaciju HP *cluster* sustava ujedinjujući procesorsku snagu 64-bitnih PC računala i jednostavnost korištenja *Active Directory* sustava za upravljanje mrežnim resursima (*Slika 5*).



Slika 5: Karakteristični CCS sustav

Glavne odlike CCS sustava su:

- jednostavno dodavanje i upravljanje čvorovima,
- automatska integracija s postojećom infrastrukturom: RIS (eng. *Remote Installation Services*), MSMS (eng. *Microsoft System Management Server*), MOM (eng. *Microsoft Operations Manager*) i MMC (eng. *Microsoft Management Console*) servisima,
- podržavanje velikog broja aplikacija,
- prepoznatljivo Windows razvojno okruženje.

7.3. Scyld Beowuf Cluster OS

Scyld Beowuf Cluster OS (SBC) je komercijalna inačica Beowulf distribucije koja se prvotno pojavila 1994. godine kao jedno od prvih rješenja za izgradnju *cluster* sustava od običnih stolnih računala. Potpunu distribuciju SBC sustava potrebno je postaviti samo na jednom čvoru s kojega se svi ostali čvorovi pokreću pomoću PXE/Etherboot alata te potom koriste dijelove distribucije kroz dijeljeni datotečni sustav. Glavi čvor omogućuje centralizirano nadgledanje i upravljanje SBC *cluster* sustavom. Prednost SBC sustava je uređena i testirana okolina sa svim potrebnim bibliotekama i razvojnim alatima, kao što su MPI i PVM programske biblioteke, GNU skup prevodilaca za C, C++ i *Fortran* programske jezike, PVFS datotečni sustav te Ganglia alat za nadzor.
Značajna komponenta SBC sustava je TaskMaster, web orijentirano sučelje za raspoređivanje poslova i opterećenja na *cluster* sustavu, koje omogućuje slanje poslova kroz formu portala.

7.4. openMosix

openMosix je SSI sustav otvorenog programskog koda nastao odvajanjem od komercijalnog MOSX (eng. *Multicomputer Operating System for UniX*) projekta. Karakteristike ovog sustava su visoke performanse postignute adaptivnim algoritmima koji, između ostalog, omogućuju automatsko prepoznavanje čvorova i automigraciju procesa.
U sklopu ovog projekta prirađeno je nekoliko distribucija Linux operacijskog sustava, od kojih su najpopularnije:

- ClusterKnoppix – baziran na Knoppix operacijskom sustavu, omogućuje podizanje cijelog *cluster* sustava s jednog medija,
- Quantin – uključuje brojne matematičke i znanstvene aplikacije,
- CHAOS – izuzetno mala (6 Mb) i sigurna distribucija.

Jezgra openMosix sustava sama donosi odluke o migraciji procesa s opterećenijeg na manje opterećen čvor čak i u složenim situacijama kada se performanse i raspoloživi resursi pojedinih čvorova razlikuju. Zbog migracije procesa nužno je korištenje oMFS (eng. *openMosix File System*) ili GFS datotečnoga sustava.

7.5. OpenSSI

OpenSSI sustav nasljednik je Locus distribucije nastale 80-ih godina prošlog stoljeća na UCLA sveučilištu, a temelji se na integriranju dostupnih rješenja otvorenog koda. Predstavlja izravnu konkurenciju openMosix sustavu čije unaprijeđene migracijske algoritme koristi u radu. Tako se prilikom migracije seli cijeli adresni prostor čime se eliminira potreba za naknadnom komunikacijom s matičnim čvorom.
U ovaj sustav ugrađeni su Lustre i CFS datotečni sustavi. OpenSSI sustav posjeduje napredne mogućnosti upravljanja procesima. Opterećenjem se balansira prilikom pokretanja procesa, ali se i tijekom njihova izvođenja migriraju dretve, procesi i grupe procesa s ponovnim otvaranjem datoteka, cjevovoda i uređaja. Moguće je mijenjati prioritete, slati signale svim procesima, a integriran je i HA-LVS (eng. *HA - Linux Virtual Server*) sustav visoke raspoloživosti.
OpenSSI dostupan je za nekoliko Linux distribucija: Fedora, Debian, Red Hat, Suse i Knoppix Live.

8. Prednosti i nedostaci *cluster* sustava

Cluster sustavi omogućuju prevladavanje ograničenja konvencionalnijih računalnih sustava. Neka od spomenutih ograničenja su:

- sekvencijalni računalni sustavi dosegli su fizičke granice svoga razvoja,
- sklopovska unaprjeđenja, kakva su protočna struktura (eng. *pipeline*) ili paralelizam na razini instrukcija (eng. *superscalar*), nisu skalabilna i zahtijevaju primjenu naprednih tehnologija,
- procesori za vektorsku obradu podataka (eng. *vector processing*) prikladni su samo za određene vrste problema, itd.

Osim nadilaženja navedenih ograničenja *cluster* sustavi imaju sljedeće prednosti:

- operacijski sustavi sa sposobnošću istovremenog obavljanja više zadataka (eng. *multitasking*) omogućuju jednostavno paraleliziranje izvršavanja zadataka na više procesora,
- brzina rada dostupnih procesora udvostručuje se svakih 18 mjeseci, ali brzine rada RAM memorija i tvrdih diskova ne prate taj trend pa je stoga izvođenjem aplikacija koje zahtijevaju pristup memoriji i/ili tvrdom disku paralelno na više računala, moguće izbjeći ovo usko grlo,
- ovisno o vrsti problema, paralelnim izvođenjem moguće je ostvariti ubrzanje od 2 do 500 puta, a ovakva ubrzanja trenutno nisu ostvariva korištenjem jednog procesora,
- razvojni alati namijenjeni radnim stanicama mnogo su napredniji od komercijalnih alata namijenjenih višeprocesorskim sustavima, prije svega zbog specifičnih karakteristika pojedinih višeprocesorskih računala,
- *cluster* sustave moguće je postepeno izgrađivati i naknadno nadograđivati, itd.

Nedostaci i mogući razlozi protiv implementacije *cluster* sustava obuhvaćaju:

- održavanje *cluster* sustava može zahtijevati dodatan angažman zaposlenika,

- postavljanje *cluster* sustava može koštati više od mogućih ušteda i dodatnih profita koje bi takav sustav donio,
- izvođenje sustava za upravljanje bazom podataka na *cluster* sustavu nema smisla ako se za pristupanje istoj koristi isključivo vlastita aplikacija koju nije moguće i/ili isplativo prilagoditi izvođenju na istom sustavu,
- složenost *cluster* sustava može otežati otkrivanje i uklanjanje pogrešaka, itd.

9. Zaključak

Ispravno postavljen *cluster* sustav može povećanjem raspoloživosti i upravljanjem opterećenjem uštedjeti značajna financijska sredstva organizaciji koja svoje poslovanje temelji na računalnim tehnologijama (npr. web). Znanstvenim ustanovama korištenje *cluster* sustava omogućuje provođenje složenih proračuna nad velikim količinama podataka za koje su prije pojave ovih sustava bila potrebna superračunala, često nedostupna nekomercijalnim institucijama. Prednosti *cluster* sustava višestruko nadilaze moguće nedostatke i njihova primjena može s vremenom samo rasti i širiti se na nova područja.

Jedna od najčešćih i najstarijih primjena Linux operacijskih sustava je izgradnja *cluster* sustava. Otvorenost operacijskog sustava koja je pridonijela njegovom razvoju, zaslužna je ujedno i za mnoštvo naprednih mogućnosti *cluster* sustava na njemu temeljenih. Duga tradicija i velika razvojna zajednica rezultirale su brojnim distribucijama i pouzdanošću koja može proizaći samo iz dugotrajnog i opsežnog testiranja.

Microsoft je novim inačicama *Windows Server 2003* omogućio jednostavno i brzo stvaranje *cluster* sustava temeljenih na najrasprostranjenijem operacijskom sustavu. To je dodatan pokazatelj važnosti *cluster* sustava i potencijala koje to tržište ima.

10. Reference

[1] Low Cost Super-Computing Using Linux Clustering, http://www.angelfire.com/linux/linuxclusters/index.htm, studeni 2006.

[2] A Gentle Introduction on How to use Cluster Effectively, http://www.tjhsst.edu/~edanaher/cluster.html, studeni 2006.

[3] Amdahl's law, http://en.wikipedia.org/wiki/Amdahl%27s_Law, studeni 2006.

[4] Dragan Jelčić, Dinko Korunić, Anđelko Iharoš: Clusteri pod Windowsima i Linuxom, Mreža, br. 12, prosinac 2005.

[5] Windows Server 2003 Clustering, Microsoft Corporation, http://download.microsoft.com/download/d/d/7/dd75ece7-83de-45da-8bb1-cb233decf595/BDMTDM.doc, prosinac 2006.

[6] Windows Compute Cluster Server 2003 Product Overview, http://www.microsoft.com/windowsserver2003/ccs/overview.mspx, prosinac 2006.

[7] OpenSSI (Single System Image) Clusters for Linux, http://openssi.org, prosinac 2006.

[8] openMosix Project, http://openmosix.sourceforge.net/, prosinac 2006.