Virtualios infrastruktūros sauga

6 paskaita

Apsauga nuo gedimų (fault tolerance)

Aukšto patikimumo (HA) sistemos

Virtualios infrastruktūros fizinė sauga

Virtuali infrastruktūra – fizinės infrastruktūros atvaizdas virtualioje konsoliduotoje aplinkoje.

Fizinis serverių patikimumas turi būti ne mažesnis nei VM patikimumas.

Virtualios infrastruktūros esminis skirtumas nuo fizinės yra tai, kad patikimumo sprendimai gali būti realizuojami tiek fiziniame, tiek ir VM lygmenyje.

Pagrindinės sąvokos

Patikimumas – tai objekto, dirbančio nustatytu režimu ir nustatytomis darbo bei aptarnavimo sąlygomis, savybė nustatytą laiką atlikti savo funkcijas, išlaikant apibrėžtą funkcionalumą ir eksploatacines charakteristikas.

Patikimumas - kompleksinė objekto savybė, įvertinama tokiomis dalinėmis jo savybėmis:

- negendamumu,
- darbingumu (funkcionalumu),
- pataisomumu,
- ilgaamžiškumu
- išsilaikymu.

Patikimumo sudėtinės dalys

Negendamumas – tai objekto gebėjimas nepertraukiamai išlaikyti savo darbingumą tam tikrą laiką.

Darbingumas – tai objekto būsena, kai jis gali atlikti savo funkcijas. Darbingumo praradimas vadinamas gedimu.

Pataisomumas – tai objekto savybė, leidžianti numatyti, aptikti ir pašalinti jo gedimus, palaikyti ir atkurti darbingumą, atliekant remontą arba techninį aptarnavimą.

Ilgaamžiškumas – tai objekto savybė išlikti darbingam (funkcionaliam) iki susidėvėjimo su pertraukomis remontams ir techninei priežiūrai.

Išsilaikymas – tai objekto savybė išlaikyti savo darbingumą (funkcionalumą) tam tikrą laiką jo nenaudojant.

Gedimai, sutrikimai (*faults*)

Gedimas – tai sistemos nukrypimas nuo darbinės būsenos, kai sistema tam tikrą laiko dalį yra neveiksni arba nepilnai atlieka savo funkcijų t.y. neteikia paslaugos.

- Kompiuterių sistemų gedimus įtakoja tokie faktoriai:
  - Aparatūrinė įranga (hardware)
  - Programinė įranga (software)
  - Kompiuterių tinklas
  - Žmogiškasis faktorius (vartotojai, sistemos administratoriai)

- Gedimai gali būti suskirstyti į tokias kategorijas:
  - Trumpalaikiai gedimai
  - Trumpalaikiai pasikartojantys gedimai
  - Ilgalaikis arba nepataisomas gedimas

Gedimų greitis

Kuo didesnis sistemos patikimumas, tuo rečiau ji genda. Vienas iš gedimus apibūdinantis statistinis rodiklis yra **gedimų intensyvumas (greitis) *λ***.

Jis apskaičiuojamas dalijant sumį gedimų skaičių per stebėjimo laiką iš suminio išdirbio per tą patį laiką.
Skaičiuojant, daroma prielaida, kad vidutinis gedimų intensyvumas yra pastovus per visą stebėjimų laiką.

Elementas	Gedimų greitis [gedimai/10^6 h]
Diodai:	
germanio	0,002–0,678
silicio	0,021–0,452
silicio-karbido	0,002–0,55
seleno	0,11–0,60
Lemputės	0,05
Tranzistoriai:	
germanio	0,6–1,91
silicio	0,27–1,44
Varžos:	
kompozicinės	0,005–0,297
pastoviosios	0,01–0,07
kintamosios	0,02–05
vielinės	0,02–0,807
anglinės	0,005–0,888
Perjungimo kontaktai	0,1
Kabeliai	0,01–0,12
Transformatoriai (įėjimo)	0,12–2,08
Autotransformatoriai	0,06
Ritės	0,001–1,082
Rėlės	0,04–0,3

Apsisaugojimas nuo gedimų (fault tolerence)

Norint apsisaugoti nuo gedimo padarinių reikia taikyti **pertekliškumo principą**, kuris sako, kad sugedus sistemai ar jos komponentui turi jo darbą perimti perteklinis to pačio funcionalumo komponentas.

Pertekliškumas (*redundancy*) gali būti trijų lygių:

- Informacijos pertekliškumas
  - Hamming kodai (atmintis, HDD: paritetinis bitas ir ECC)
- Laiko pertekliškumas
  - Užlaikymai (timeout), pakartotinės užklausos, siuntimai (retransmit)
- Fizinis (virtualus) pertekliškumas
  - *N-modulinis* pertekliškumas: RAID diskai, rezervinio kopijavimo serveriai, replikuojantys serveriai, aukšto patikimumo serveriai

Kokio lygio fizinė sauga galima?

100 % apsauga nuo gedimų neįmanoma.

- Kuo sistemos negendamumo lygmuo artimesnis 100%, tuo ji brangesnė.

Sakoma, kad sistema yra apsaugota nuo **k** gedimų (***k-fault tolerant***), jei ji:

- Turi *k+1* komponentų, iš kurių *k* gali sugesti, bet likęs vienas palaikys sistemos funkcionalumą;
- Turi *2k+1* komponentą su *Byzantine tipo gedimais,* kai *k* komponentų gali sugedę, o *k+1* komponentas palaikys funkcionalumą.

“Devintukų” metodas

Veiksnumas procentais	Neveiksnumas procentais	Neveiksnumas per metus	Neveiksnumas per savaitę
98 %	**2 %**	**7,3 dienos**	**3 val., 22 min.**
99 %	**1 %**	**3,65 dienos**	**1 val., 41 min.**
99,8 %	**0,2 %**	**17 val., 30 min**	**20 min., 10 sek.**
99,9 %	**0,1 %**	**8 val., 45 min.**	**10 min., 5 sek.**
99,99 %	**0,01 %**	**52 min., 30 sek.**	**1 min.**
99,999 %	**0,001 %**	**5,25 min.**	**6 sek.**
99,9999 %	**0,0001 %**	**31,5 sek**	**0,6 sek.**

Sistemos patikimumui (availability) matuoti panaudojant devintukų (*NINES*) metodą, kuris parodo, kiek laiko procentais sistema buvo pasiekiama ir veiksni.

Pasiekiamumas/patikimumas

Sistemos patikimumą taip pat galima įvertinti žinant jos:

- **vidutinį laiką tarp gedimų** (*MTBF – Mean Time Between Failures*)
- **vidutinį gedimų šalinimo laiką** (*MTTR – Maximum Time To Repair*).

Skaičiavimui naudojama Marcus – Stern formulė:

$$A = \frac{MTBF}{MTBF + MTTR}.$$

Iš formulės matome, kad mažėjant gedimų šalinimo laikui, bendras patikimumas artėja prie 100 %. Ir gedimų šalinimo laiko įtaka sistemos patikimumui mažėja, didėjant vidutiniam laikui tarp gedimų.

IT sistemų patikimumas

IT sistema – tai sluoksninė struktūra, kurios pasiekiamumas/ patikimumas priklauso nuo atskirų jos sluoksnių patikimumo ir sistemos komponentų sujungimo būdų.

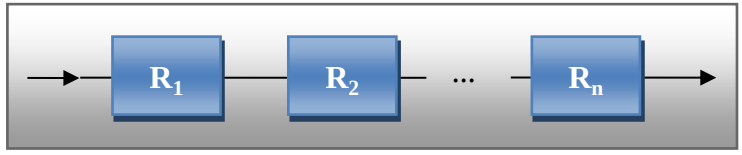
Išskiriami tokie IT sistemos sluoksniai:

- Aparatūrinis
- Tinklo
- Operacinės sistemos
- Programų sistemų - servisų
- Aplikacijų/paslaugų

Virtualios IT infrastruktūros atveju įvedamas papildomas VMM (hypervizoriaus) sluoksnis.

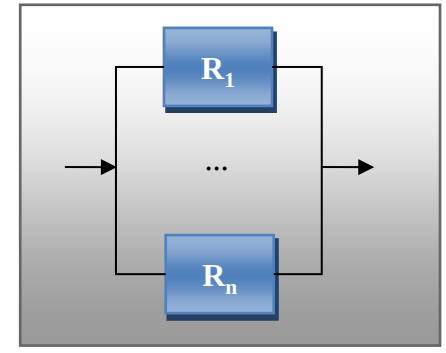
Patikimumo skaičiavimas

Nerezervuota sistema

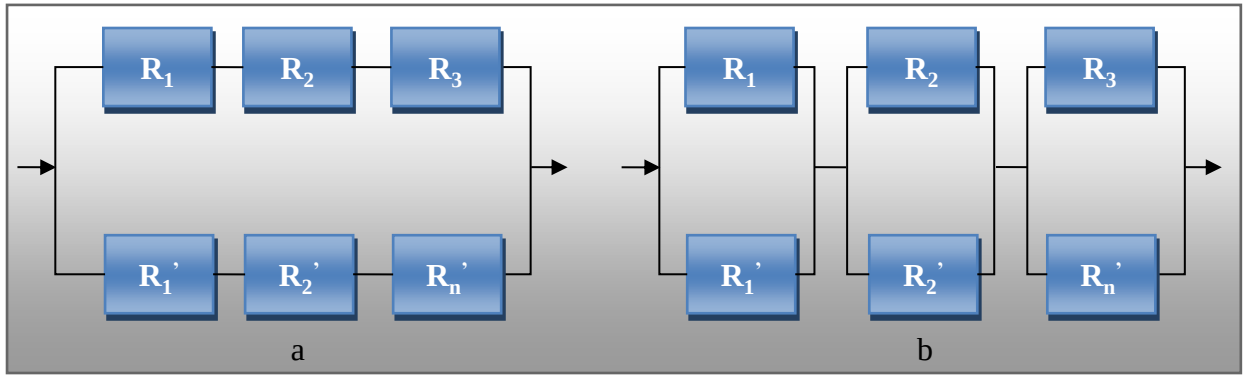


$$R_S(t) = \prod_{i=1}^{n} R_i(t),$$

Rezervuota sistema



$$R_S(t) = 1 - \prod_{i=1}^{n} [(1 - R]_i(t)),$$



$$R_{AL} = 2R_a \cdot R_b \cdot R_c - R_a^2 \cdot R_b^2 \cdot R_c^2, \quad R_{ZL} = \left(2R_a - R_a^2\right)\left(2R_b - R_b^2\right)\left(2R_c - R_c^2\right)$$

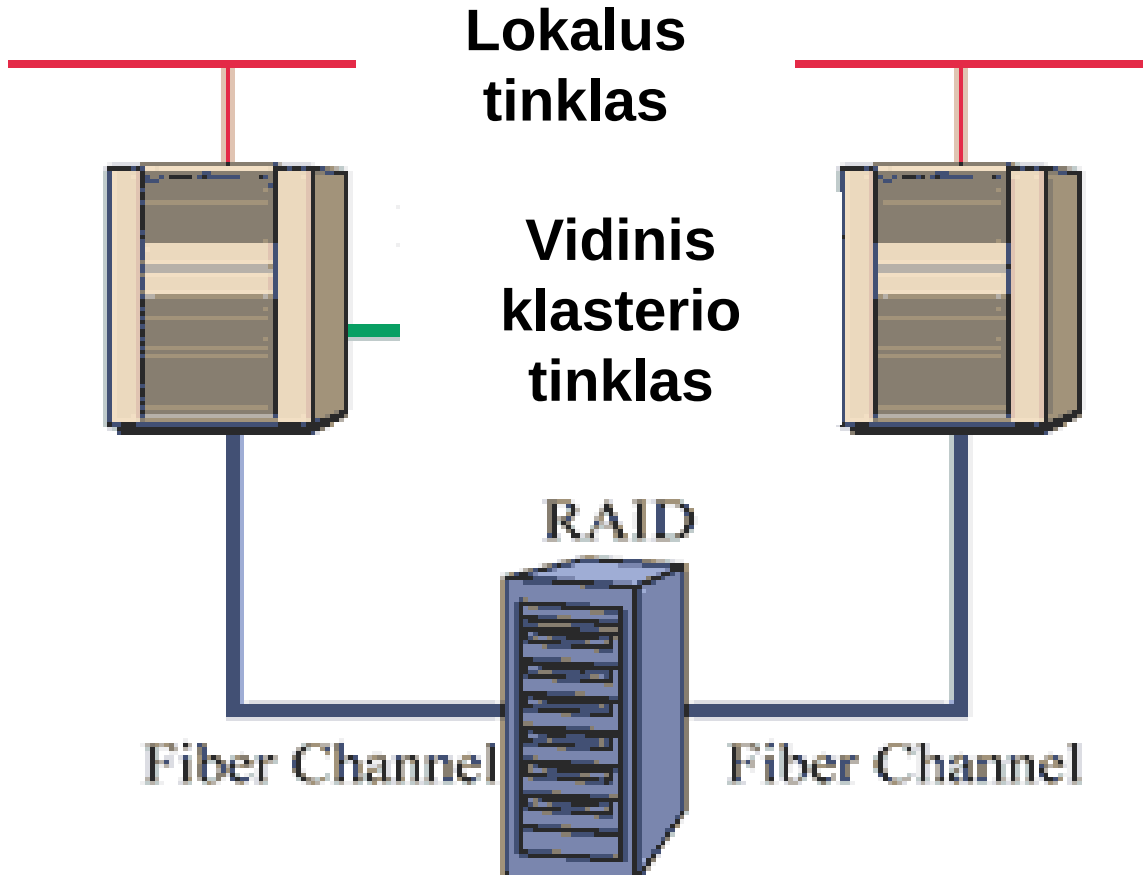
Klasterių tipai

Siekiant užtikrinti sistemų patikimumą, kai iš kart apimami visi sistemų sluoksniai, naudojamos **klasteriai** ir **pertekliniai serveriai (RAIN)**.

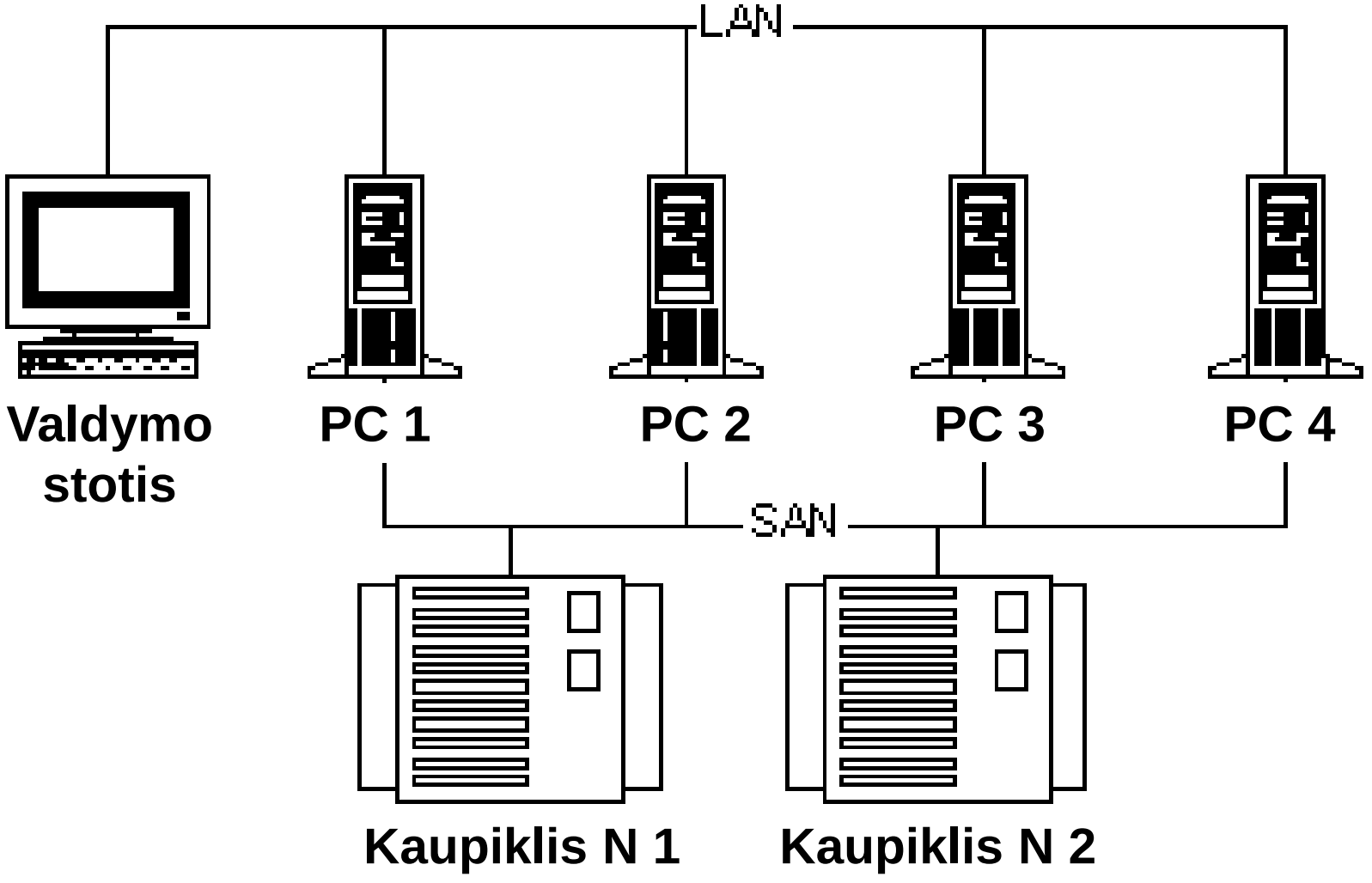
Kompiuterių klasteriai pagal naudojimo sritį skirstomi į:

- Didelio našumo klasterius (*angl. High-Performance Computing clusters – HPC*).
- Apkrovos balanso klasterius (*angl. Load-Balancing clusters – LB*).
- Didelio patikimumo klasterius (*angl. High-Availabilty clusters – HA*).

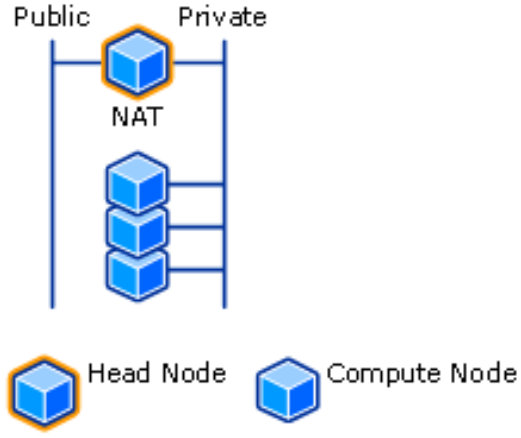
Paprasčiausias klasteris

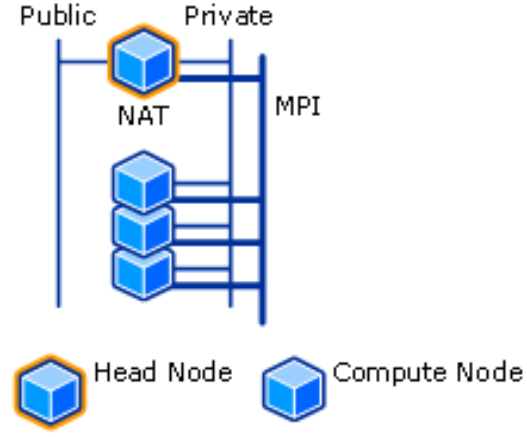


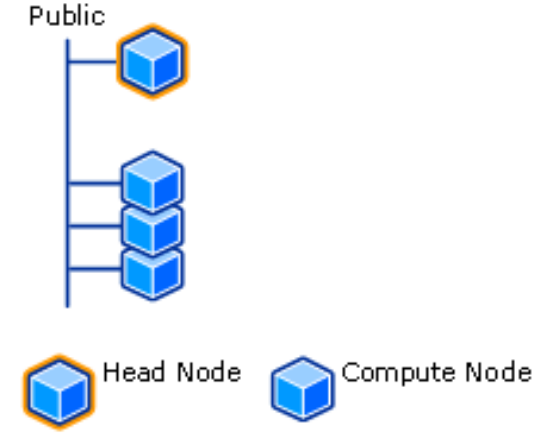
Sudėtingesnis klasteris

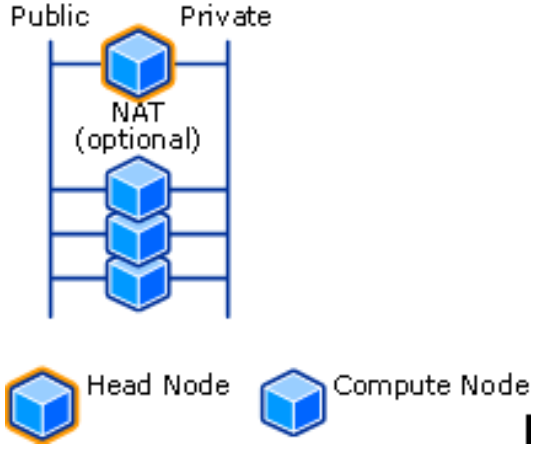


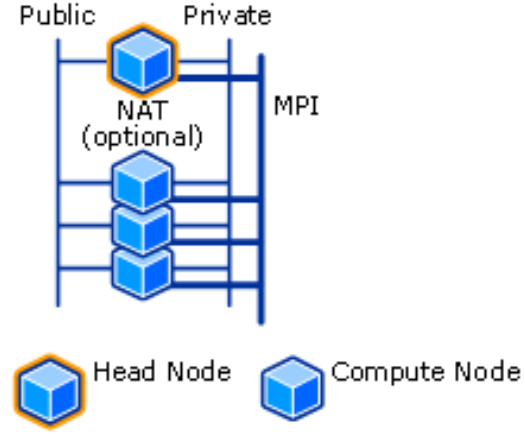
HPC klasterio mazgų tinklas











HPC

Didelio našumo klasteriai – tai brandžiausia ir dažniausiai naudojama klasterių grupė, skirta labai didelių skaičiavimo išteklių reikalaujančių uždavinių sprendimui.

HPC klasteriuose naudojama:

- unifikuoti kompiuteriai, turintys vienodas operacines
- sistemas didelio pralaidumo komunikacijų tinklas.

HPC paskirtis:

- Uždaviniai su dideliais duomenų kiekiais
- Uždaviniai reikalaujantys daug skaičiavimo laiko
- Lygiagretaus tipo uždaviniai

HPC klasteriai

Moksliniams tyrimams sukurti klasteriai yra reitinguojami www.top500.org.

Reitingas skaičiuojami pagal HPL testo rezultatus.

Home / Lists / November 2012

Top500 List - November 2012

R_{max} and R_{peak} values are in TFlops. For more details about other fields, check the TOP500 description.

previous | 1 | 2 | 3 | 4 | 5 | next

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560640	17590.0	27112.5	8209
2	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1572864	16324.8	20132.7	7890
3	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705024	10510.0	11280.4	12660
4	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786432	8162.4	10066.3	3945
5	Forschungszentrum Juelich (FZJ) Germany	**JUQUEEN** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	393216	4141.2	5033.2	1970
6	Leibniz Rechenzentrum	**SuperMUC** - iDataPlex	147456	2897.0	3185.1	3423

Patinka 2.264 people like this.

TOP10 November 2012

1. Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x
2. Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom
3. K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect
4. Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom
5. JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect

Pagreitėjimas

Pagal Amdahl dėsnį **efektyvus pagreitėjimas** (*speedup*) išreiškiamas taip:

$$S = 1 / ((1-f) + f/k),$$

kur **f** - lygiagretaus kodo dalis, **k** - vektorinio procesoriaus santykinis greitis skaliarinio procesoriaus atžvilgiu.

Vektorinio, ypač daugelio procesorių mašinų bendras našumas labai priklauso nuo ryšių, sisteminės programinės įrangos ir kt. faktorių. Todėl galima rašyti, kad lygiagretaus skaičiavimo laikas:

$$T_p(n) = c_p(n) + o_p(n),$$

kur c_p - skaičiavimo laikas, o_p - laiko nuostoliai (*overhead*), **p** - procesorių skaičius, o **n** - duomenis charakterizuojantis dydis (uždavinio dydis).

Pagreitėjimas

Tada pagreitėjimas $\mathbf{S_p}$ rodys, kiek kartų **p** procesorių turinti sistema išspręs uždavinį greičiau už vieną procesorių:

$$\mathbf{S_p = T_1 / T_p.}$$

Paprastai pagreitėjimas

$$\mathbf{1 \leq S_p \leq p.}$$

Buvo pasiūlyta Amdahl dėsnio modifikacija, kai skiriami laiko nuostoliai $\mathbf{T_{po}}$, išlygiagretinama dalis $\mathbf{T_{pp}}$ ir neišlygiagretinama dalis $\mathbf{T_{ps}}$.

Tada **pagreitėjimas** $\mathbf{S_p}$ vieno procesoriaus atžvilgiu išreiškiamas taip:

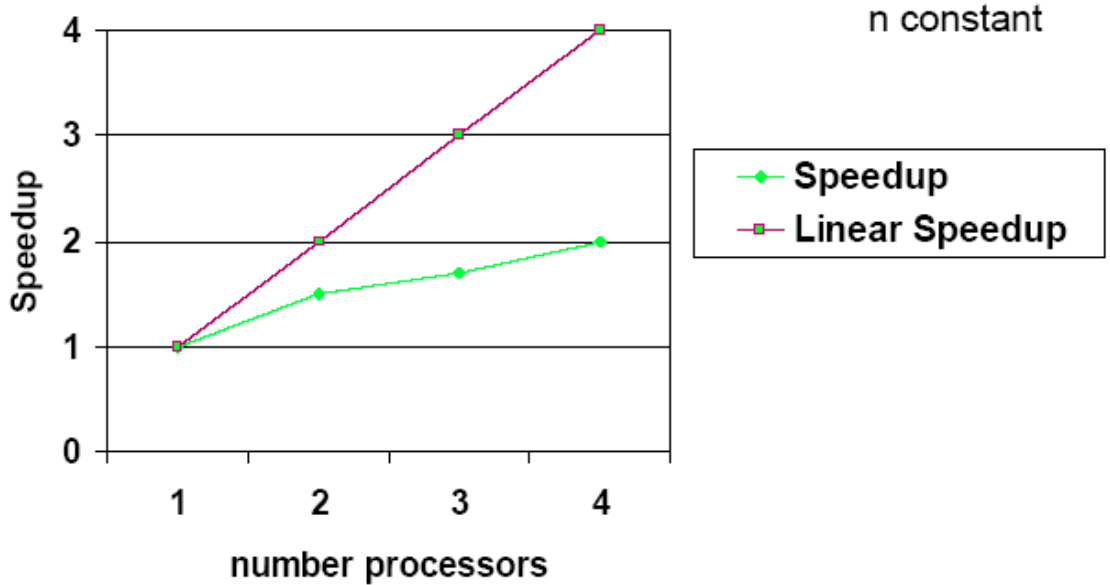
$$\mathbf{S_p = T_1 / (T_{po} + T_{pp} / p + T_{ps}).}$$

Pagreitėjimas

- S(N,P) įvertina pagreitėjimą, kurį pasiekiame spręsdami lygiagrečiai
- Paprastai: S(N, P) ≤ P. (papildomos sąnaudos, overhead).
- Tiesinis (Ideal speedup): S(N, P) = P

T(n,1) = 300 s
T(n,2) = 200 s
T(n,3) = 170 s
T(n,4) = 150 s

S(n,2) = 300 s / 200 s = 1,5
S(n,3) = 300 s / 170 s = 1,7
S(n,4) = 300 s / 150 s = 2



Efektyvumas, pertekliškumas

Efektyvumu E_p vadinsime santykį

$$E_p = S_p / p = T_1 / (p*T_p)$$

rodantį, kiek multiprocesoriuje atskiri procesoriai apkrauti lyginant su vienu procesoriumi. Tai rodo skaičiavimų ekonominį efektyvumą.

Pertekliškumu R_p vadinsime **p** procesoriuose įvykdytų operacijų skaičiaus santykį su operacijų skaičiumi, reikalingu uždaviniui išspręsti paprastame procesoriuje :

$$R_p = O_p / O_1 .$$

R_p visuomet didesnis už vienetą (dėl valdymui reikalingų laiko nuostolių).

LB klasteris

Apkrovos balansavimas – tai kompiuterių tinklų metodika, skirta paskirstyti apkrovą tarp daugelio kompiuterių, kompiuterių klasterių, tinklo mazgų, CPU, diskų ar kitų resursų.

Apkrovos balansavimo tikslas – pasiekti sumažinti užklausų vykdymo laiką, optimizuoti resursų apkrovimą, išvengiant perkrovų.

LB serveriai dažniausiai naudojami internetiniuose serveriuose: web, e-pašto, naujienų, DNS serverių.

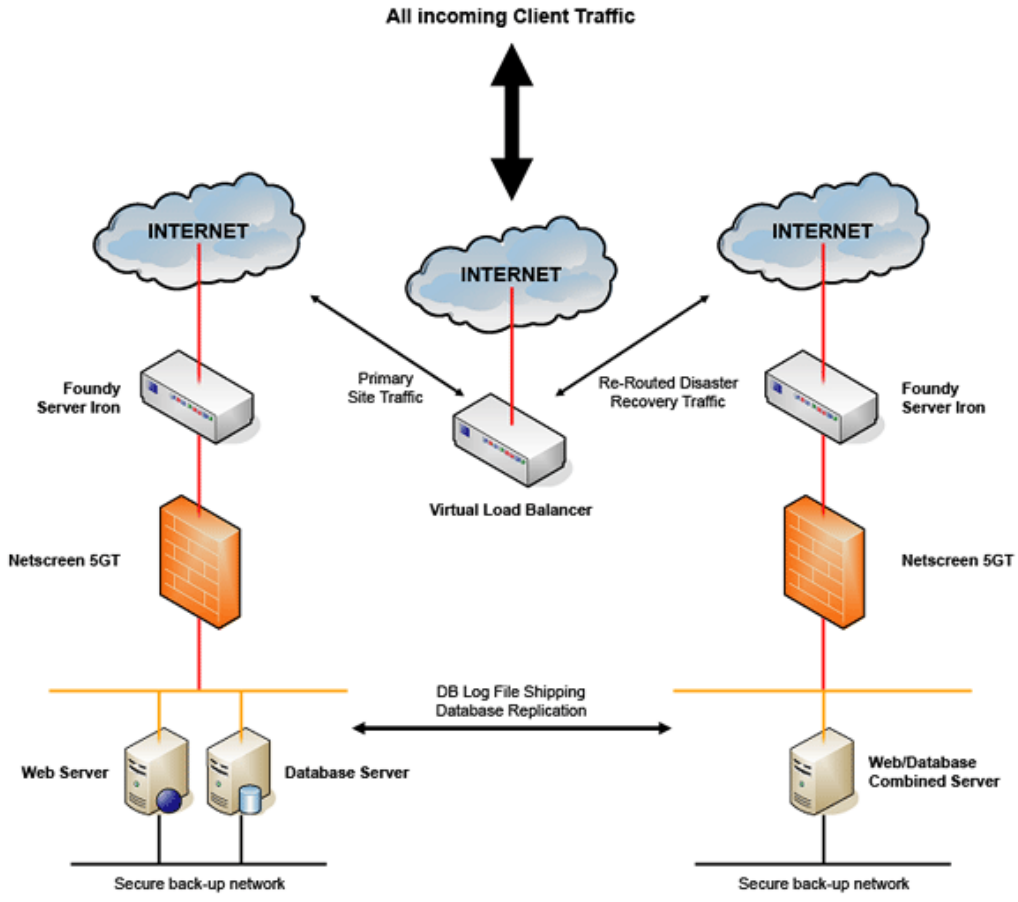
LB taip pat naudojami prieigai prie replikuojančių DB.

LB klasteris

Apkrovos balanso (LB) klasterį sudaro:

- vidiniai (*back-end*)
- išoriniai mazgai (*front-end*).

Išoriniai LB klasterio serveriai komunikuoja su naudotojais, stebi vidinių mazgų apkrovimą ir būseną realiuoju laiku ir pagal iš anksto nustatytas taisykles paskirsto vartotojų užduotis mažiausiai užimtiems vidiniams klasterio mazgams. Šis pirminis LB sistemos elementas dar kitaip vadinamas apkrovos balanso tarnybine stotimi arba tarpininku.



Vidinius klasterio mazgus sudaro serveriai su programine įranga klientų užklausoms apdoroti.

Apkrovos balansavimo būdai

Balansavimo algoritmai:

- Pasirenkamas serveris su mažiausiu TCP sujungimų skaičiumi
- Svorio koeficientų principas
- Parinkimas atliekamas **round-robin** principu.
- Pasirenkamas geriausią ryšį turintis serveris (SYN/ACK time)

Balansavimo būdai

- Peradresavimas (redirect)
- Persiuntimai (forward)

Apkrovos balansavimas

Funkcionalumas

- Suriša vieną/kelis virtualius adresus (IP, MAC) su fiziniais adresais t.y.
  - Įeinančios užklausos surišama su konkrečiu fiziniu adresu, parinkimas atliekamas pagal vieną iš balansavimo algoritmų.

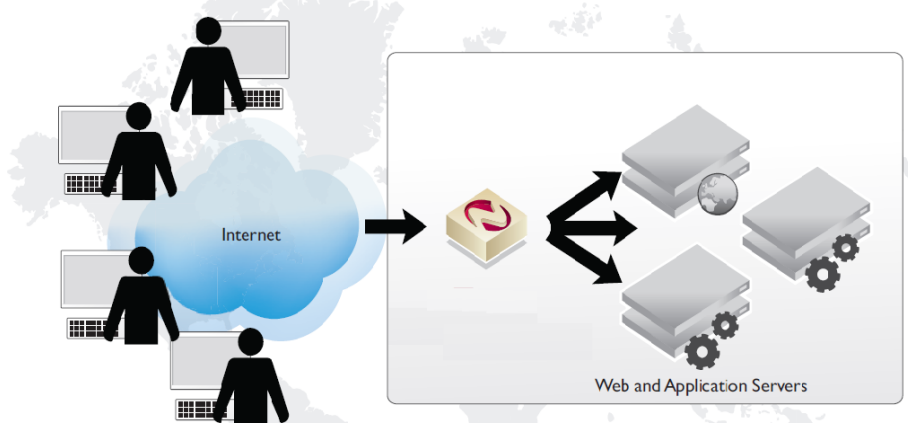
- Priskirimai gali būti atliekami diferencijuojant pagal porto numerius, pvz.
  - visas FTP srautas gali būti priskiriamas vienai mašinai.

Apkrovos balansavimas

Apkrovos balansavimo programinė įranga

BALANCE – tai atviro kodo TCP proxy programa, naudojanti *round-robin* apkrovos skirstymo principą ir palaikanti HA (*failover)* funkcionalumą. Ji skirta TCP/IP sesijų srautams paskirstyti tarp tarnybinių stočių. (www.inlab.de)

ZEUS Load balancer – komercinė apkrovos balansavimo įranga, galinti dirbti su SSL protokolu. Apkrovos balansavimas gali remtis taisyklių principu. (www.zeus.com)



Apkrovos balansavimas

VMware ESX serverio apkrovos balansavimo metodai.

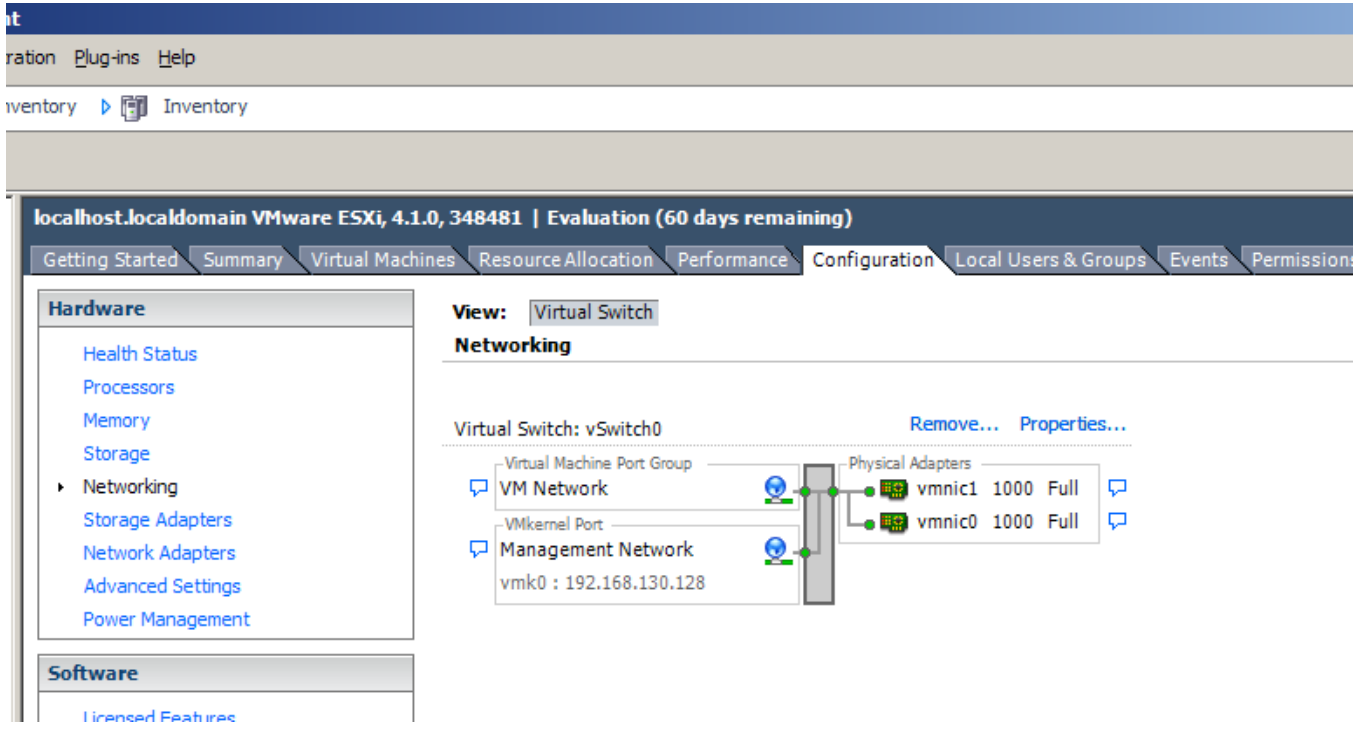
Balansuojant apkrovą, VMkernel nusprendžia per kurią pNIC IP paketas turi būti išsiųstas.

Balanasavimui naudojami tokie maršrutizavimo algoritmai:

- pagal prievado ID (nesudaro CPU apkrovos)
- Pagal siuntėjo MAC hash (vidutinė CPU apkrova)
- Pagal IP hash (imlus CPU apkrovai)

pNIC grupavimas (VMware)

Norint gauti apkrovos balansavimo ar padidinto patikimumo sistemas, keli pNIC priskiriami vienam virtualiam komutatoriui ir apjungiami į grupę (NIC teaming).



HA klasteris

Aukšto patikimumo klasterio (HA) paskirtis – užtikrinti sistemos paslaugų nenutrūkstamą pasiekiamumą. Pasiekiamumo lygmuo apibrėžiamas SLA ir svyruoja nuo 99% iki 99.999 %.

Visi HA sprendimai paremti pertekliškumo principu, t.y. naudojama perteklinė įranga (mazgai, tinklo įranga, saugyklos), siekiant išvengti klasteryje SPOF (*single points of failure*) ir užtikrinti sistemos pasiekiamumą.

Perteklinių komponentų jungimas – lygiagretus.

HA veikimo algoritmas gedimo atveju:

- Detektuojamas gedimas ir izoliuojamas sugedęs mazgas
- Perimami sugedusio mazgo tinkliniai nustatymai (IP adresas, vardas, maršrutizavimo lentelė, MAC adresas ir t.t.)
- Apkrova perskirstoma likusiems mazgams

HA klasterio užduotys

- Kaip detektuoti gedimą ir užtikrinti automatinį jo šalinimą (failover)?
- Per kiek laiko bus detektuotas gedimas?
- Kaip ir kur neveikianti aplikacija bus atstatoma?

Gedimo detektavimas (Heartbeat)

- **Gedimo detektavimo būdas:**

  “ping” mechanizmas t.y. UDP paketų periodinis siuntimas visam tinklui ir programų scenarijų vykdymas klaidų atveju (heartbeat).

  VMware ESX atveju, heartbeat realizuojamas tarp ESX service console.

  Siekiant išvengti tinklo komponentų įtakos detektuojant gedimą, reikia dubliuoti tinklo įrangą (arba naudoti atskirą tinklą – private network) arba naudoti tiesioginį serverių tinklo plokščių sujungimą laidu.

Heartbeat saugumas

Galimos heartbeat saugumo grėsmės:

Tai gali veikti kaip:

- DoS ataka (dirbtinai suklaidinama HA ir VM migruoja į vieną ESX)
- kaip neautorizuoto prisijungimo būdas.

Heartbeat tarnyboje galima konfigūruoti atsako laukimo laiką ir paketų skaičių į kurį neatsako serveris.

HA sistemos modeliai

Patikimos (rezervuotosios) sistemos atveju elementai yra jungiami lygiagrečiai ir sistema veikia tol, kol veikia bent vienas sistemos elementas.

Egzistuoja du rezervuotų sistemų modeliai (*failover configuration models*):

Aktyvus/Pasyvus

- lygiagrečiai sujungtų elementų sistemoje veikia tik pagrindinis elementas ir tik jam sugedus yra įjungiamas rezervinis.

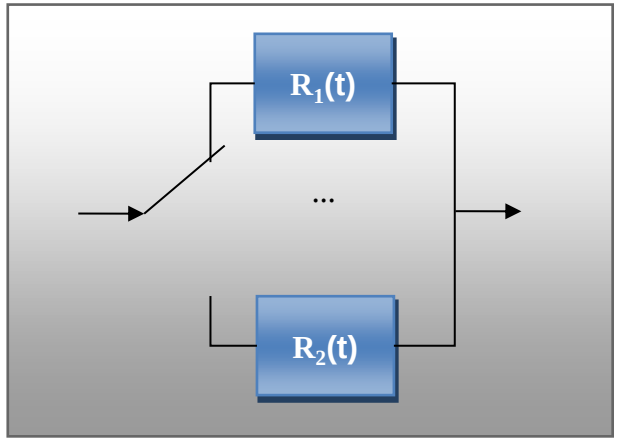
Aktyvus/Aktyvus

- lygiagrečiai sujungtų elementų sistemoje veikia visi elementai vienu metu, o sugedus vienam iš sistemos elementų, kiti elementai perima sugedusio apkrovą ir sistema veikia tol, kol veikia bent vienas elementas.

HA sistemos modeliai

Aktyvus/Pasyvus HA sistemos patikimumas, kai sistema sudaryta iš dviejų elementų, kurių gedimai nepriklauso vienas nuo kito, yra apskaičiuojamas:

$$R_S(t) = R_1(t) - \int_0^t R_2(t - t_2)\frac{d}{dt_2}R_1(t_2)dt_2,$$

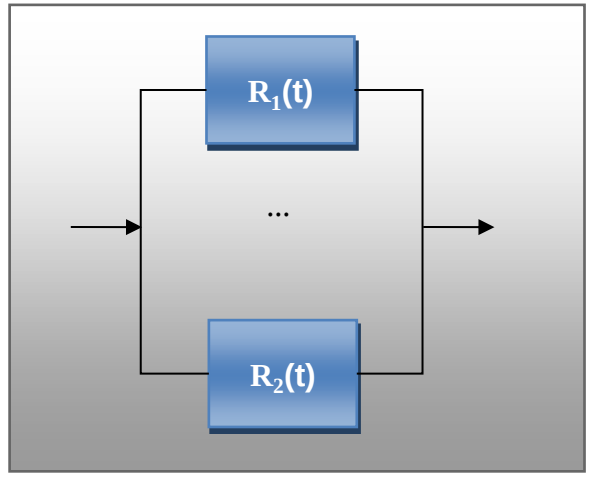


$R_S(t)$ -sistemos patikimumas;
$R_1(t)$ - atitinkamai pirmo ir antro elemento patikimumas;
t_2 - laikas nuo kurio yra aktyvuotas antras pasyvus elementas

HA sistemos modeliai

Aktyvus/Aktyvus HA sistemos patikimumas, kai elementų gedimai nepriklausomi vienas nuo kito, randami:

$$R_S(t) = 1 - \prod_{i=1}^{n}[(1 - R]_i(t)).$$

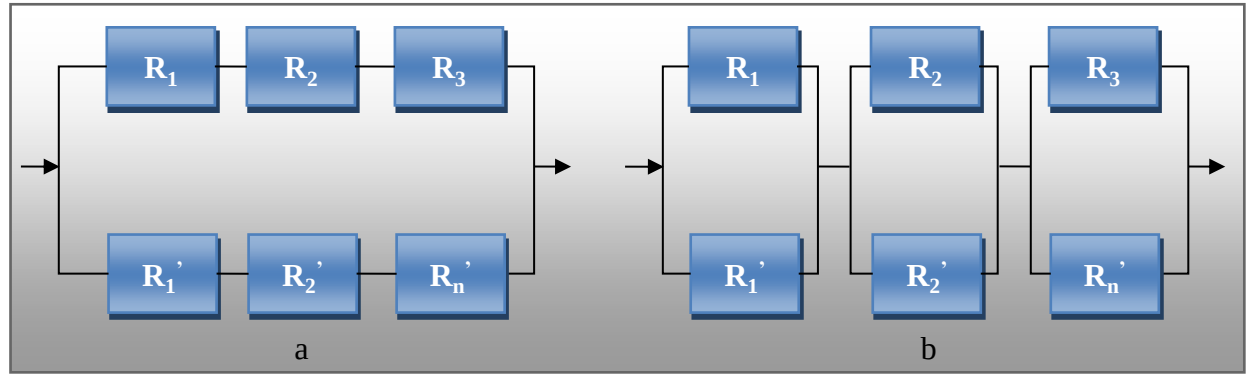


Komponentinis patikimumo modelis

Papildomai išskiriami tokie patikimumo (rezervavimo) atvejai:

- bendrasis sistemos rezervavimas (aukšto lygio rezervavimas)
- dalinis (žemo lygio arba komponentinis) rezervavimas.

Bendrojo rezervavimo atveju visa sistema yra dubliuojama. Tuo tarpu dalinio rezervavimo atveju yra dubliuojamos sistemos atskiros posistemės ar komponentai.



m/N Aktyvus modelis

Tegul sistema turi *N* lygiagrečiai sujungtų elementų. Kad ji reikiamai funkcionuotų, *m* iš *N* elementų turi būti nesugedę. Tokia sistema vadinama *m*/*N* aktyviuoju rezervavimu.

Esant identiškiems komponentams, tokios *m*/*N* aktyviai rezervuotosios sistemos patikimumą galima apskaičiuoti naudojantis formule:

$$R_a = 1 - \sum_{n=N-m+1}^{N} C_n^N \left(1-R\right)^n R^{N-n}, \qquad \text{kur} \quad C_n^N = \frac{N!}{\left(N-n\right)!\,n!}.$$

Gedimų šalinimo tipai

Šaltasis *(Cold failover)*

- Aplikacija perstartuoja įvykus gedimui, dingsta neišsaugota informacija

Šiltasis *(Warm failover)*

- Aplikacija periodiškai naudoja kontrolinius taškus (*checkpoints*)
- Aplikacija perstartuoja į paskutinę kontrolinio taško būseną

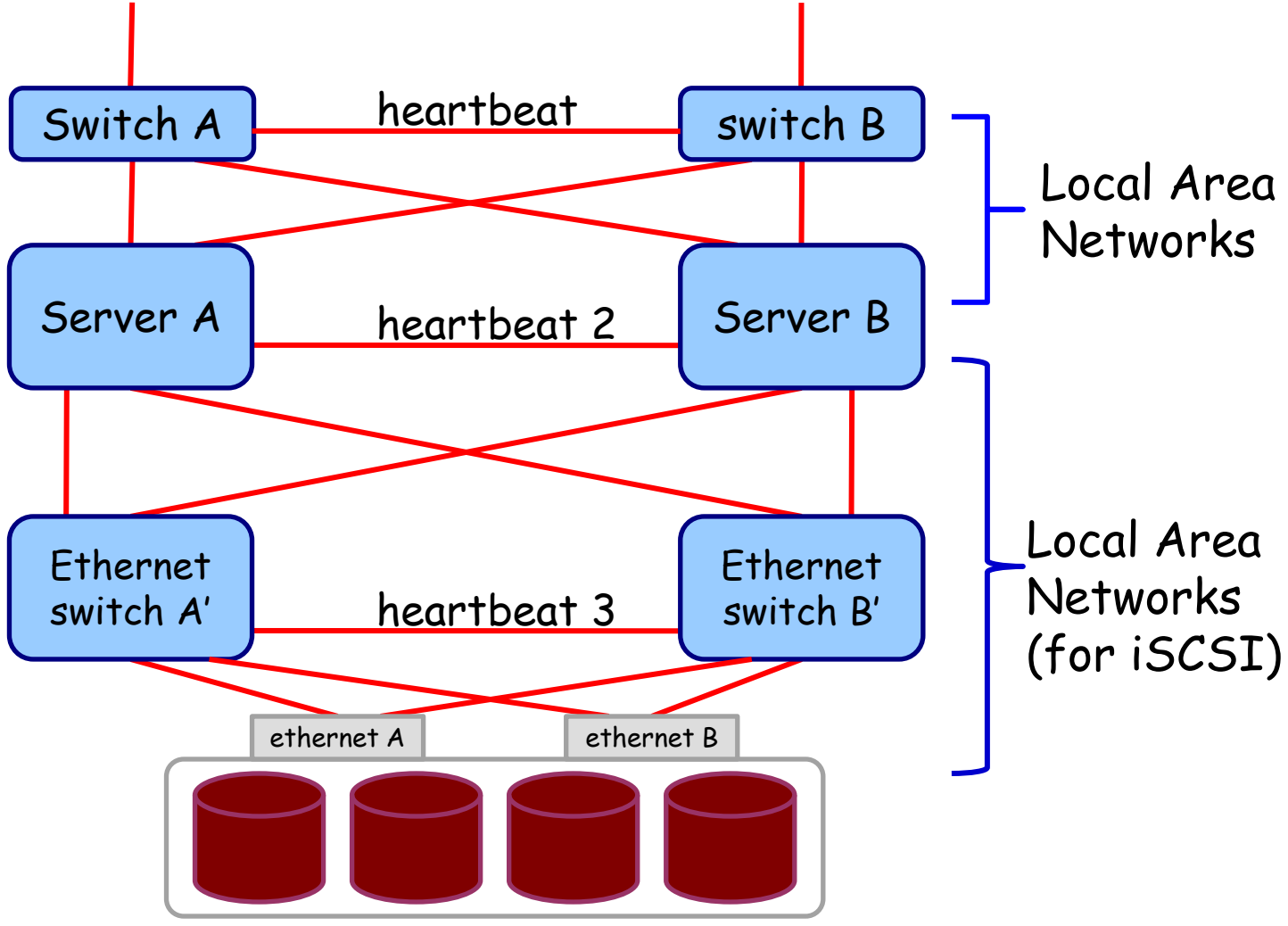
Karštasis *(Hot failover)*

- Aplikacijos būsena sinchronizuojama su jos kopija po kiekvieno pakeitimo. Gedimo atveju veikia aplikacijos kopija. Neveiksnumo laikas artimas *0* s.

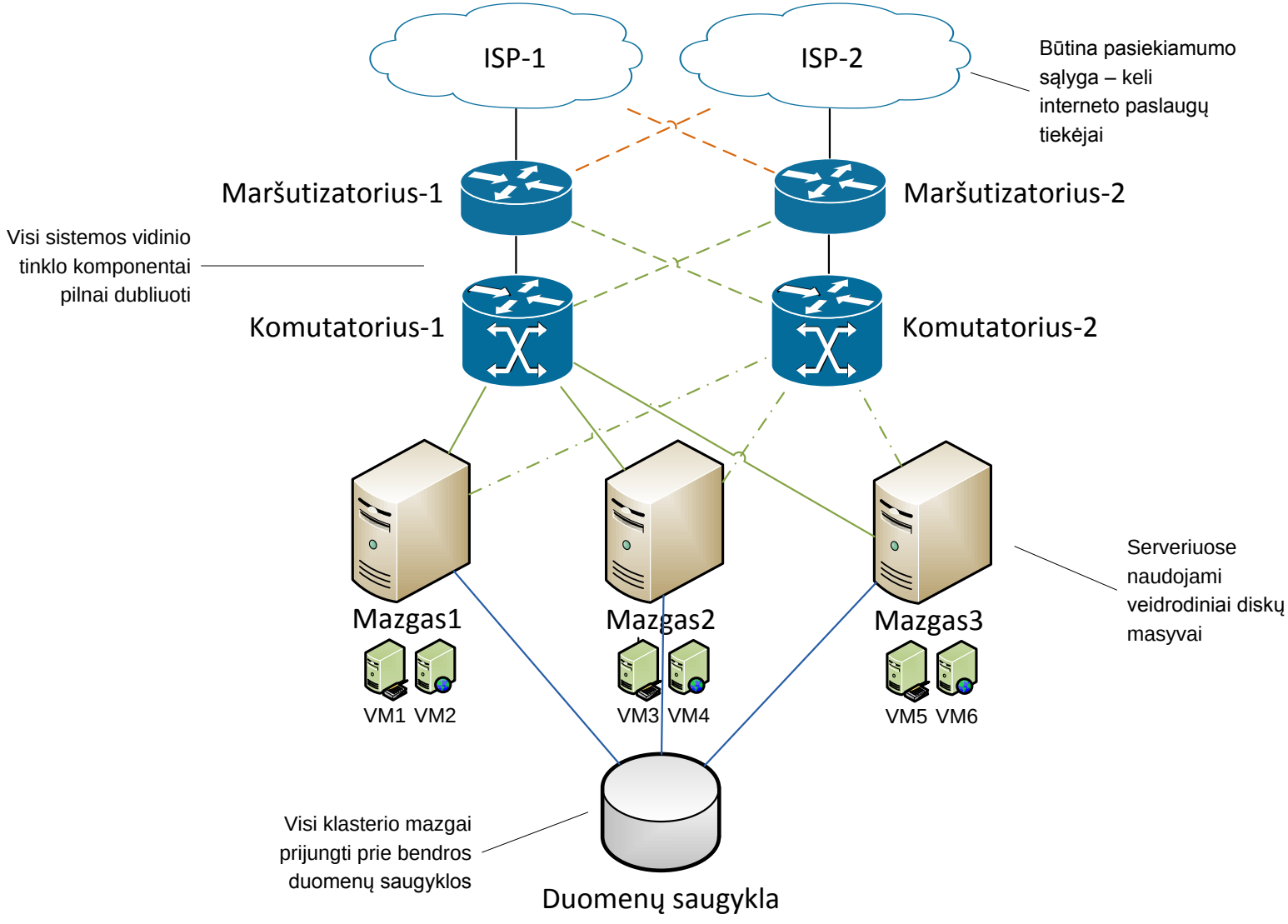
HA sistemų komponentai

- **Karšto keitimo įrenginiai**
  - Minimizuojamas prastovos laikas keičiant įrenginį.
- **Pertekliniai, dubliuoti įrenginiai**
  - Maitinimo blokai, ventiliatoriai
  - Atmintis su paritetu ir ECC
  - RAID diskų masyvai
  - Automatiškai persijungiantys komponentai (elektros tiekimo linijos, interneto tiekėjai ir t.t.)
- **Bendro naudojimo saugyklos**
  - Serveriai jungiami prie vienos saugyklos. Užtikrinama galimybė prisijungti kito serverio failines sistemas, LUN, kuriuos naudoja kitas serveris.

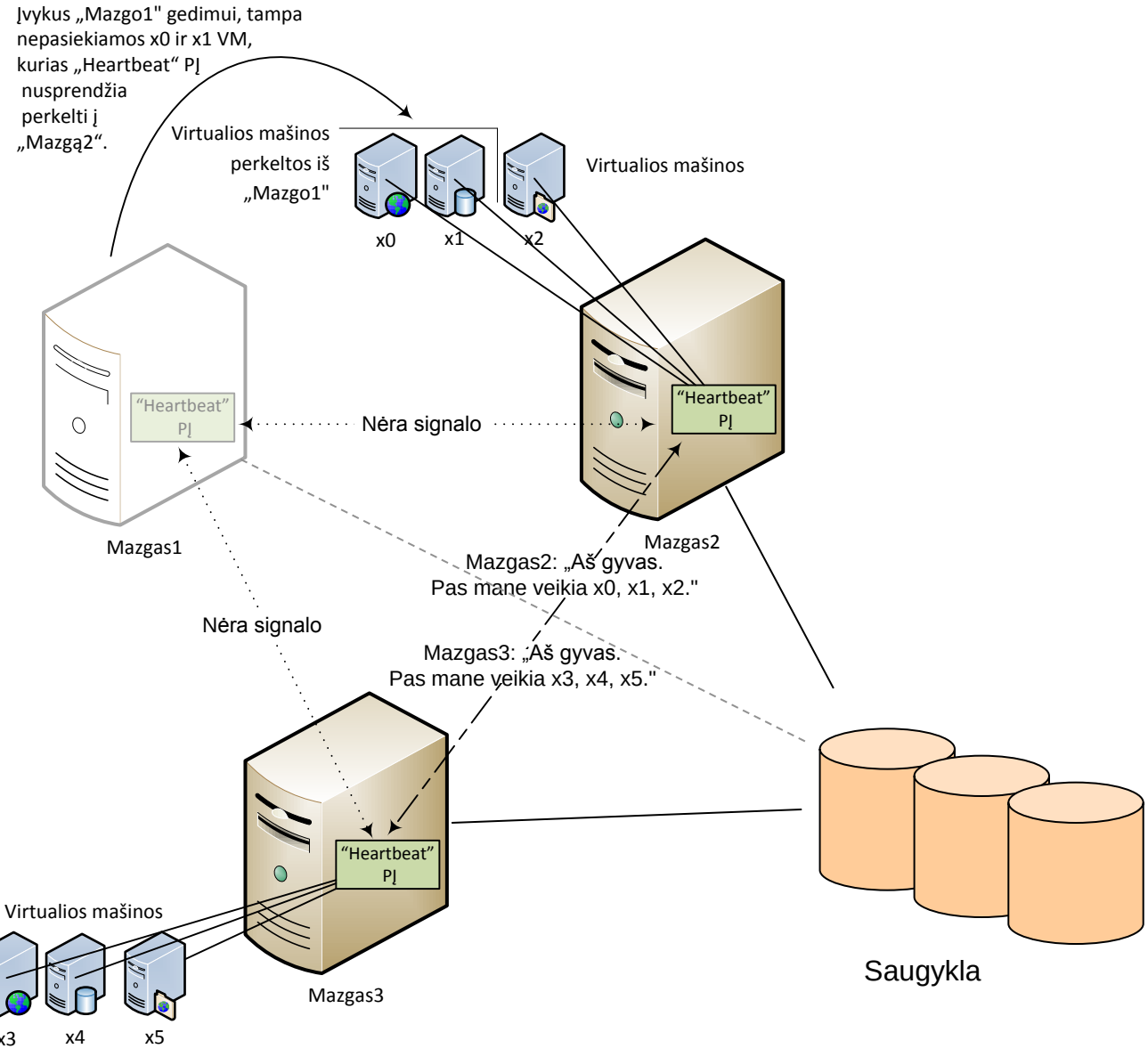
HA sistemos pavyzdys



HA sistema su VM



Virtualizacija HA klasteriuose



VMware ESX klasteris

VMware klasterį sudaro 2 ar daugiau ESX serverių.

Klasteris leidžia apjungti fizinių serverių resursus (CPU ir RAM) į bendrą virtualių resursų grupę ir dalintis jais.

ESX lokalių diskų ir tinklo resursai nedalinami.

Klasteryje realizuojami tokie VM funkcionalumai:

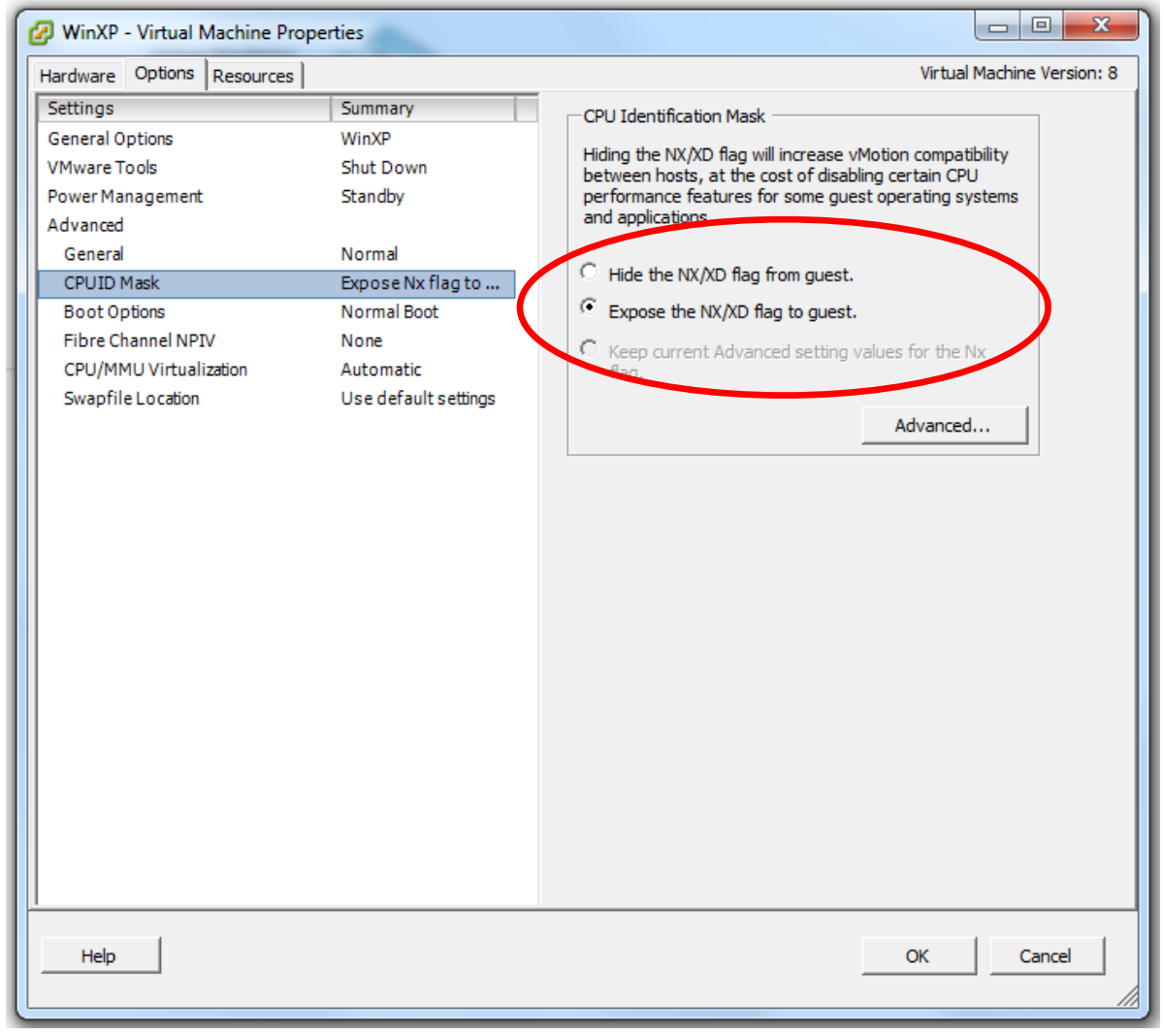
- vMotion – t.y. VM automatinis perkėlimas iš vieno ESX serverio į kitą
- VM perstartavimas
- Dinaminis ESX serverio išjungimas nepiko metu.

Sąlygos VM migravimui

Norint realizuoti migravimą tarp ESX serverių keliamos tokios sąlygos:

- Abu serveriai pasiekia tą patį SAN LUN arba NAS įrenginį
- Ne mažesnis nei 1Gbps tinklo pralaidumas
- CPU ID kaukė turi būti išjungta dėl didesnio suderinamumo
- Prieiga prie to pačio fizinio tinklo
- Virtualių komutatorių konfigūracija tokia:
  - portų grupių pavadinimas sutampa,
  - vSwitch pavadinimai nesutampa,
  - vmnic pavadinimai nesutampa.

CPU ID kaukė


WinXP - Virtual Machine Properties
Hardware
Options
Resources
Virtual Machine Version: 8
Settings
Summary
General Options
WinXP
VMware Tools
Shut Down
Power Management
Standby
Advanced
General
Normal
CPUID Mask
Expose Nx flag to ...
Boot Options
Normal Boot
Fibre Channel NPIV
None
CPU/MMU Virtualization
Automatic
Swapfile Location
Use default settings
CPU Identification Mask
Hiding the NX/XD flag will increase vMotion compatibility between hosts, at the cost of disabling certain CPU performance features for some guest operating systems and applications.
Hide the NX/XD flag from guest.
Expose the NX/XD flag to guest.
Keep current Advanced setting values for the Nx flag
Advanced...
Help
OK
Cancel

VMware ESX klasteris

VMware klasterio savybės:

HA – didelis patikimumas realizuojamas migruojant VM

RDS (Dynamic Resource Scheduling) – VM migravimas pagal išskirtų resursų kiekį.

DPM (Distributed Power Management) – VM migravimas ir ESX išjungimas nepiko metu.

FT (fault tolerant) – VM gyvas atvaizdas (live shadow instance), kuris daromas pažingsniui su CPU operacijomis su pirminiu atvaizdu. Esant aparatūriniam gedimui, FT eliminuoja net ir mažiausią duomenų pakeitimo praradimą.

Distributed virtual switch – tai klasterio virtualių komutatorių centralizuoto konfigūravimo funkcija.

HA klasterio ribojimai

Konfigūruojant HA klasterį, atliekami tokie darbai:

- Nustatomo klasterio patikimumo politika – t.y. Leidžiamas serverio gedimų skaičius (paprastai 1...4)
- Nustatomi pirminiai (primary) ir antriniai (secondary) mazgai
- Nustatomi galimos resursų atsargos VM perkėlimui
- Kada ir kokiu eiliškumu bus atstatytos VM jas numigravus į kitą serverį.

į

Dingo tinklo ryšys

Kas įvyksta kai dingsta tinklo ryšys?

Atliekami tikrinimai:

- ping į visus klasterio mazgus
- ping į gateway

Jei nei vienas iš ICMP paketų negrįžta, serveris pereina į izoliuotą rėžimą. Šiame rėžime VM išjungiamos (pagal nutylėjimą).