Aukšto pasiekiamumo (HA) sistemos

13 paskaita

Apsauga nuo gedimų (fault tolerance)

HA sistemos

Pagrindinės sąvokos

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

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

- negendamumu,
- 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 iki susidėvėjimo su pertraukomis remontams ir techninei priežiūrai.

Išsilaikymas – tai objekto savybė išlaikyti savo darbingumą 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 funkcijas.

- Kompiuterių sistemų gedimus įtakoja tokie faktoriai:
  - Aparatūrinė įranga (hardware)
  - Programinė įranga (software)
  - 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 suminį 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), paritetinė ir ECC tipo atmintis
- Laiko pertekliškumas
  - Užlaikymai (timeout), pakartotinės užklausos, siuntimai (retransmit)
- Fizinis pertekliškumas
  - *N-modulinis* pertekliškumas, RAID diskai, rezervinio kopijavimo serveriai, replikuojantys serveriai, aukšto patikimumo serveriai

Kokio lygio apsauga galima?

100 % apsisaugoti 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 pasiekiamumui matuoti panaudojant devintukų (*NINES*) metodą, kuris parodo, kiek laiko procentais sistema buvo pasiekiama ir veiksni.

Pasiekiamumas/patikimumas

Sistemos pasiekiamumą 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 Marcuso – Sterno 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

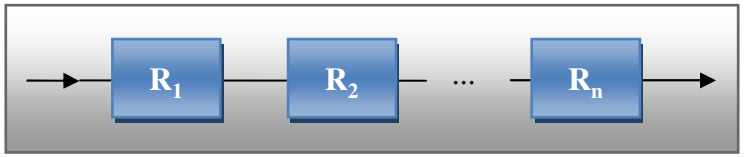
IT sistema – tai sluoksninė struktūra, kurios pasiekiamumas/ patikimumas priklauso nuo atskirų jos sluoksnių patikimumo ir sistemos komponentų sujungimo būdų.

Išskiriami tokie IT sistemos sluoksniai:

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

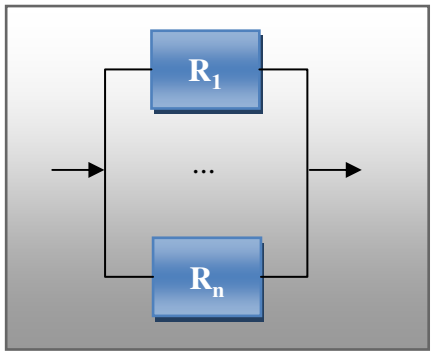
Patikimumo skaičiavimas

Nerezervuota sistema

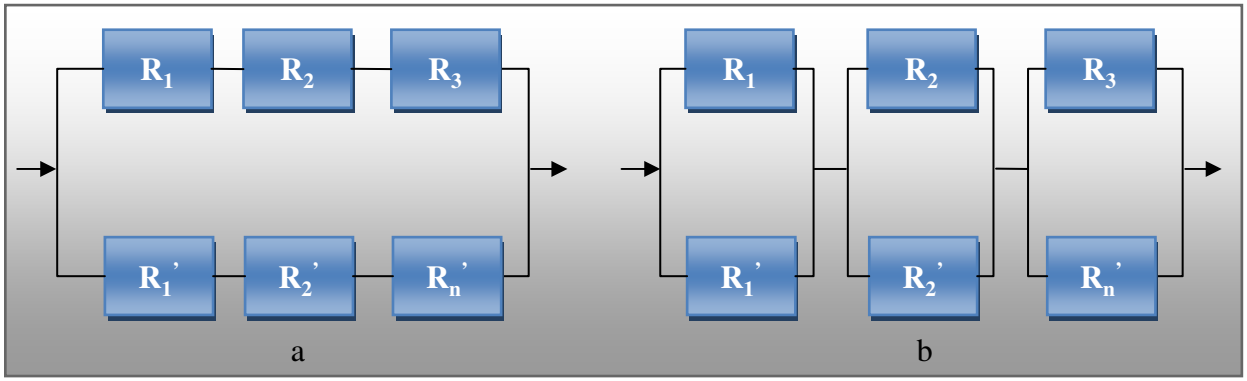


$$R_S(t) = \prod_{i=1}^{n} R_i(t).$$

Rezervuota sistema



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



$$R_{AL} = 2R_a \cdot R_b \cdot R_c - R_a^2 \cdot R_b^2 \cdot R_c^2,$$

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

Klasterių tipai

Siekiant užtikrinti fizinį sistemų patikimumą, kai iš kart apimai visi sistemų sluoskniai, naudojamos **klasteriai** ir **replikavimo serveriai**.

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/pasiekiamumo klasterius (*angl. High-Availabilty clusters – HA*).

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:

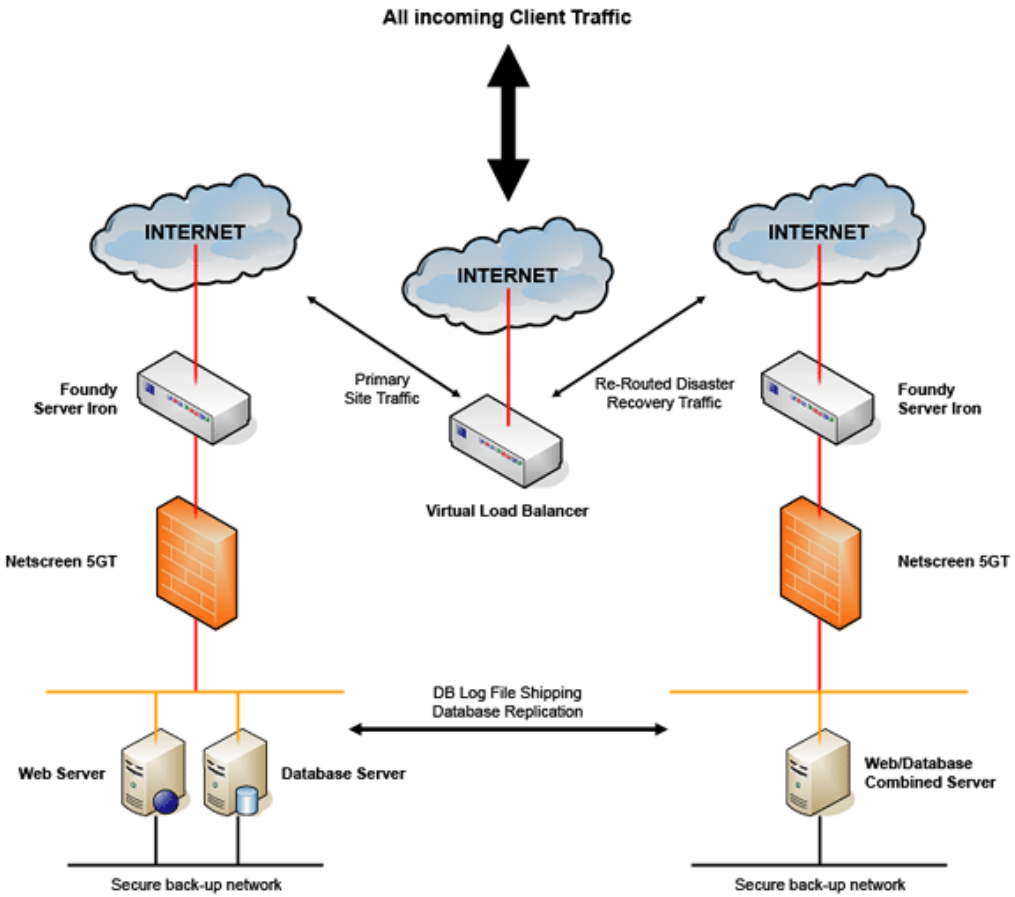
- Uždaviniai su dideliais duomenų kiekiais
- Uždaviniai reikalaujantys daug skaičiavimo laiko
- Lygiagretaus tipo uždaviniai

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 būdai

- Peradresavimas (redirect)
- Persiuntimai (forward)
- Balansavimas pagal apkrovą

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)

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.

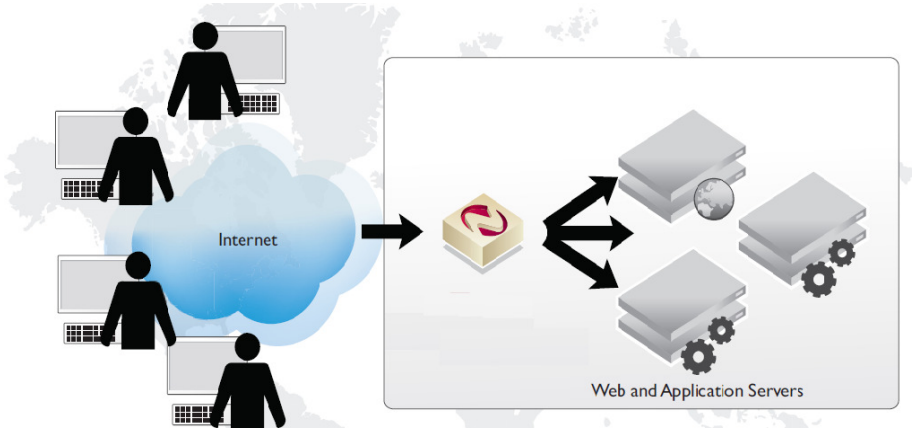
- Priskirimai gali būti atliekami diferencijuojant pagal sąsajos numerius, pvz.
  - visas FTP srautas gali būti priskiriamas vienai mašinai.

Apkrovos balansavimas

Apkrovos balansavimo programinė įranga

BALANCE – tai atviro kodo TCP proxy programa, naudojanti *round-robin* apkrovos skirstymo princip1 ir palaikanti failover. 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)



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 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).
  - 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.

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.

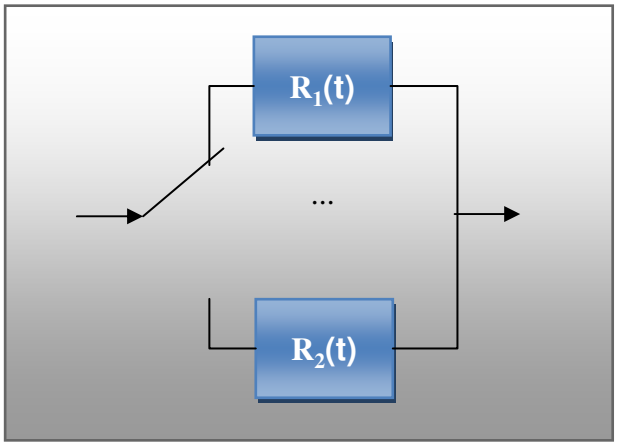
Aktyvus/Aktyvus

- 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;

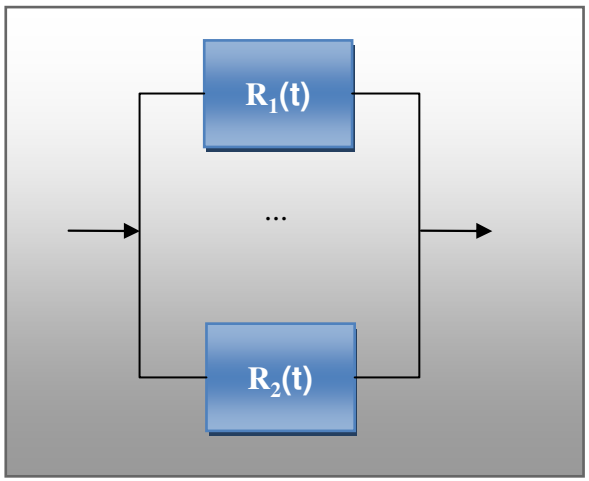
$R_1(t)$ - atitinkamai pirmo ir antro elemento patikimumas;

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:

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

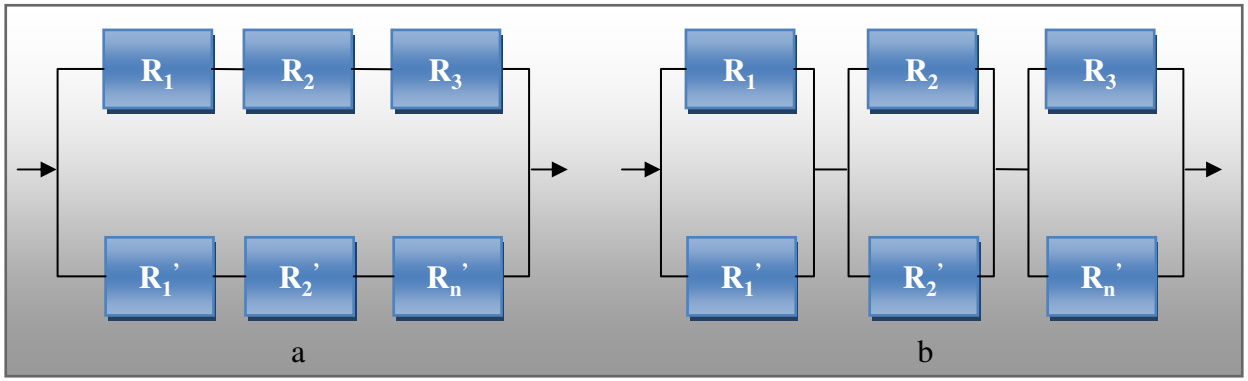


Komponentinis patikimumo modelis

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.



Pavyzdys

Sistema sudaryta iš n = 10 vienodo patikimumo elementų. Kiekvieno elemento patikimumas Ri=0,9. Kiek reikia rezervinių elementų abiem rezervavimo būdais, kad gautume sistemos patikimumą 0,95?

Sprendimas

Rezervinių grandžių skaičių bendrojo rezervavimo atveju galime apskaičiuoti naudodamiesi formule:

$$\left(1-R_i^n\right)^{m+1}=1-R_{sist}(t),$$

Įstatę reikšmes ir apskaičiavę, gauname, kad m = 6 t.y. reikės 6 papildomų rezervinių grandžių po 10 elementų – iš viso 60 elementų.

Tęsinys

Apskaičiuojame rezervinių elementų skaičių dalinio rezervavimo atveju, pasinaudodami formule:

$$1-(1-R_i)^{m+1} = \sqrt[n]{R_{sist.}}$$

Įstatę reikšmes ir apskaičiavę, gauname, kad m = 1 t.y. reikės papildomos 1 rezervinės grandies iš 10 elementų.

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:

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

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

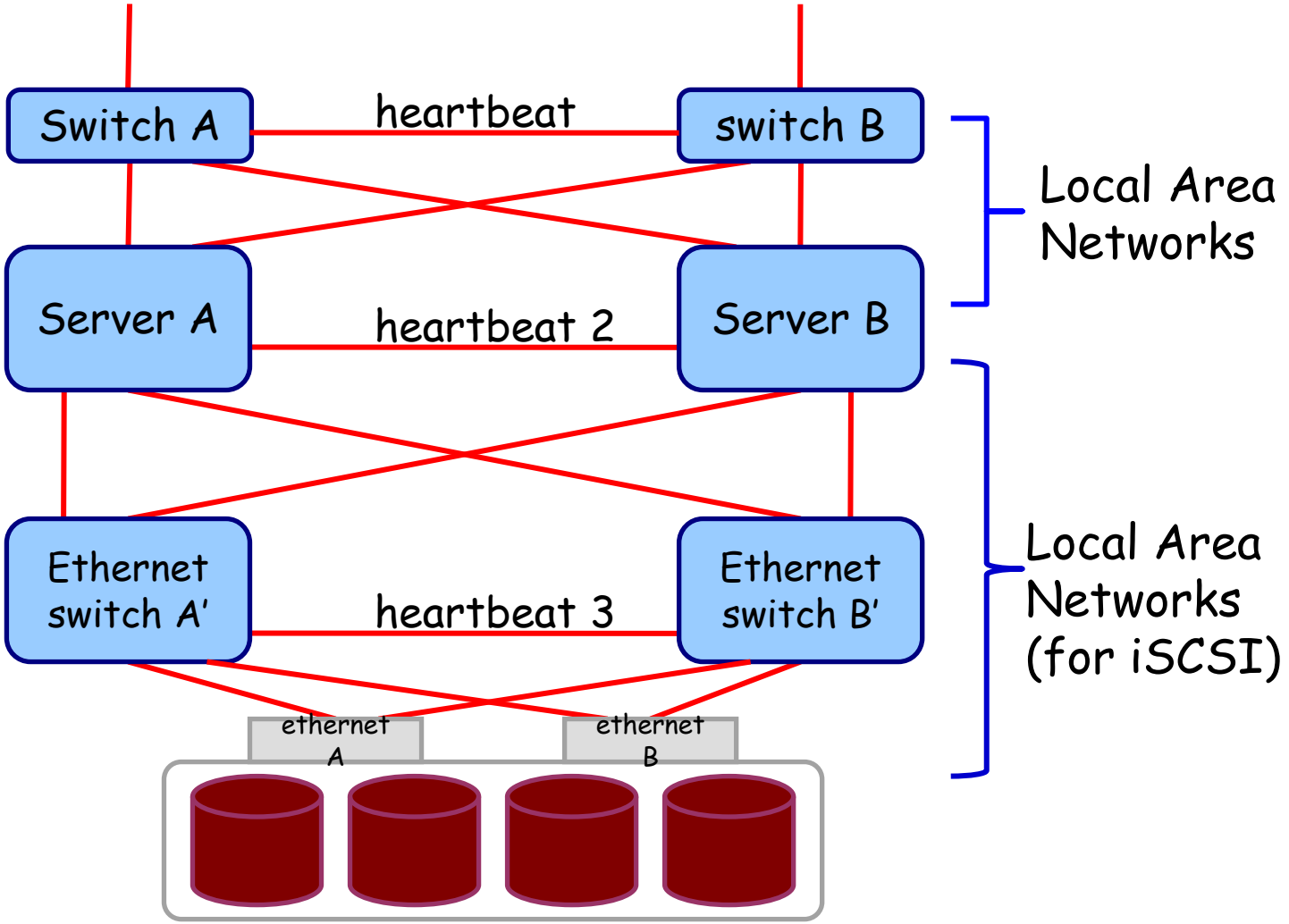
Karštasis *(Hot failover)*

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

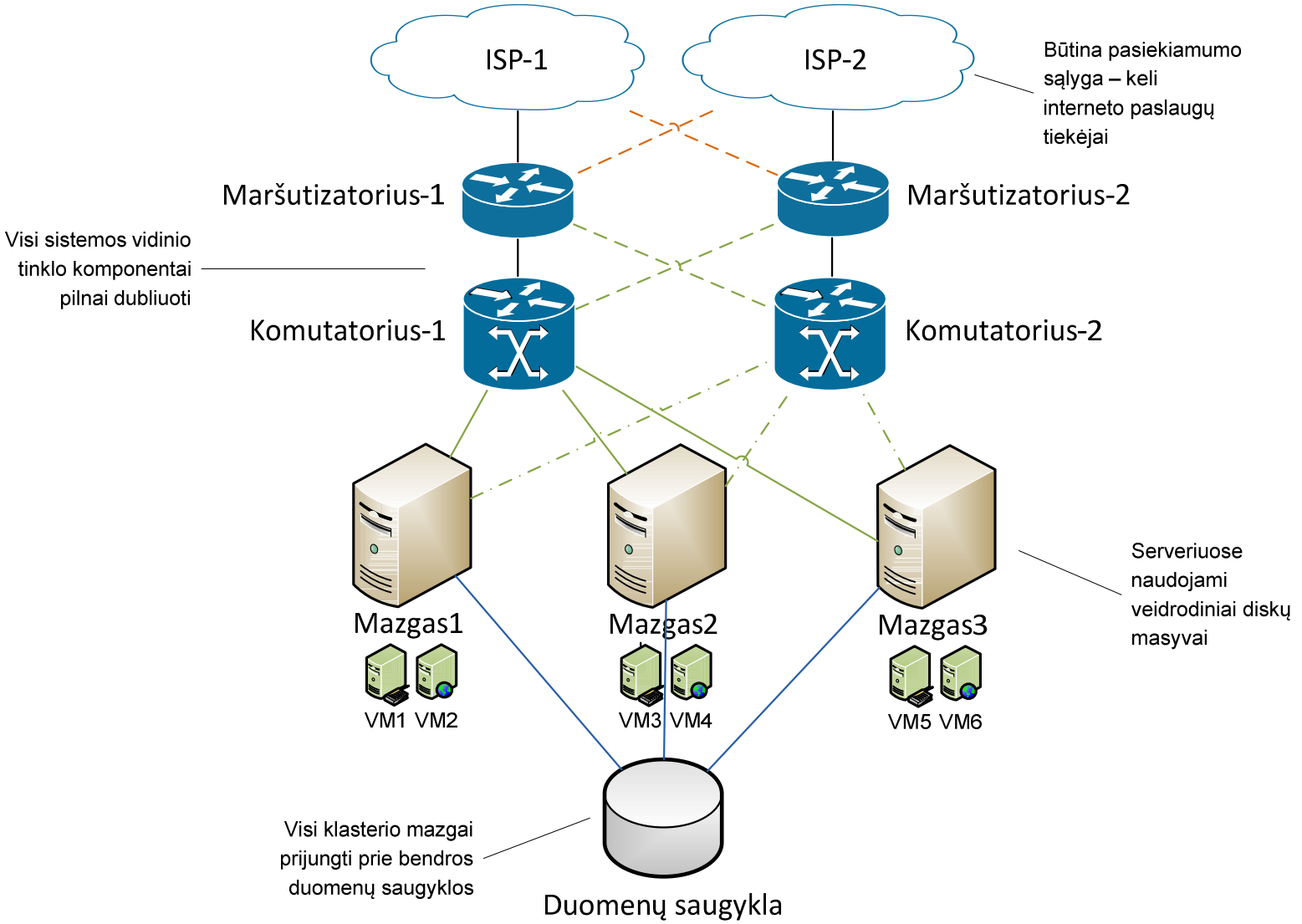
HA sistemų komponentai

- **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 failines sistemas, LUN, kuriuos naudoja kitas serveris.

HA sistemos pavyzdys



HA sistema su VM



Virtualizacija HA klasteriuose