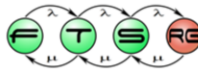


Rendszermonitorozás

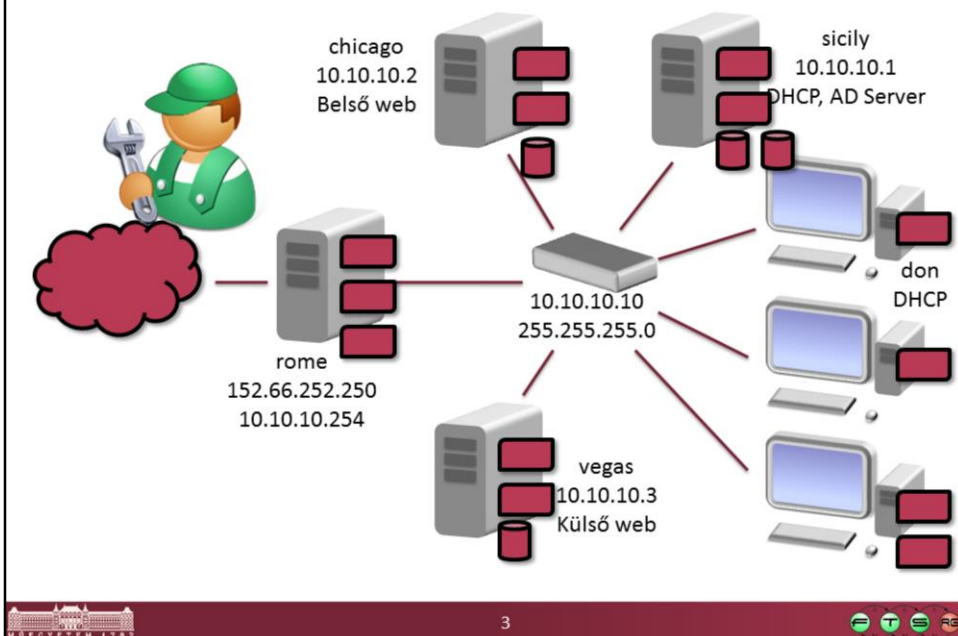
Tóth Dániel, Kocsis Imre,
Salánki Ágnes, Micskei Zoltán



„When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge of it is of a meager and unsatisfactory kind”

Lord Kelvin

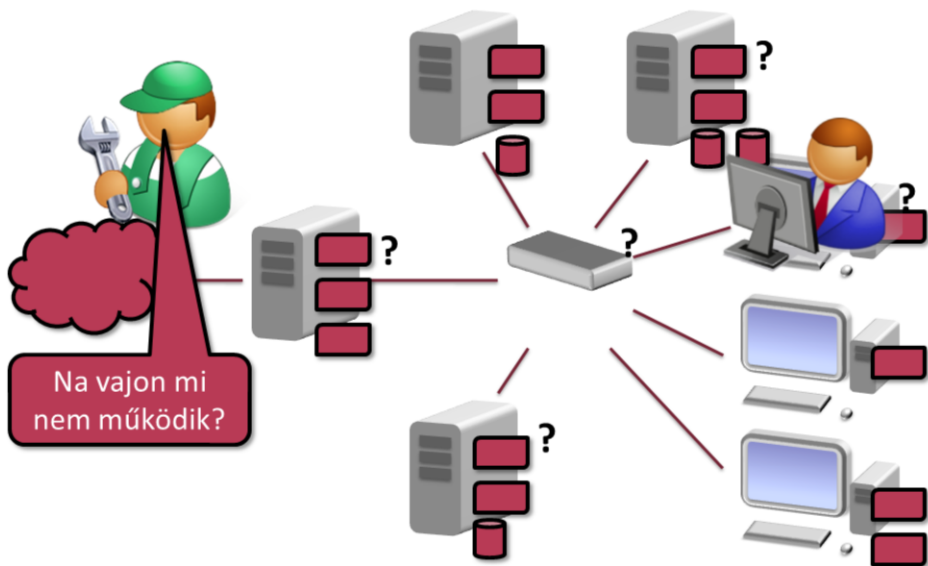
„Kézbentartott” rendszer



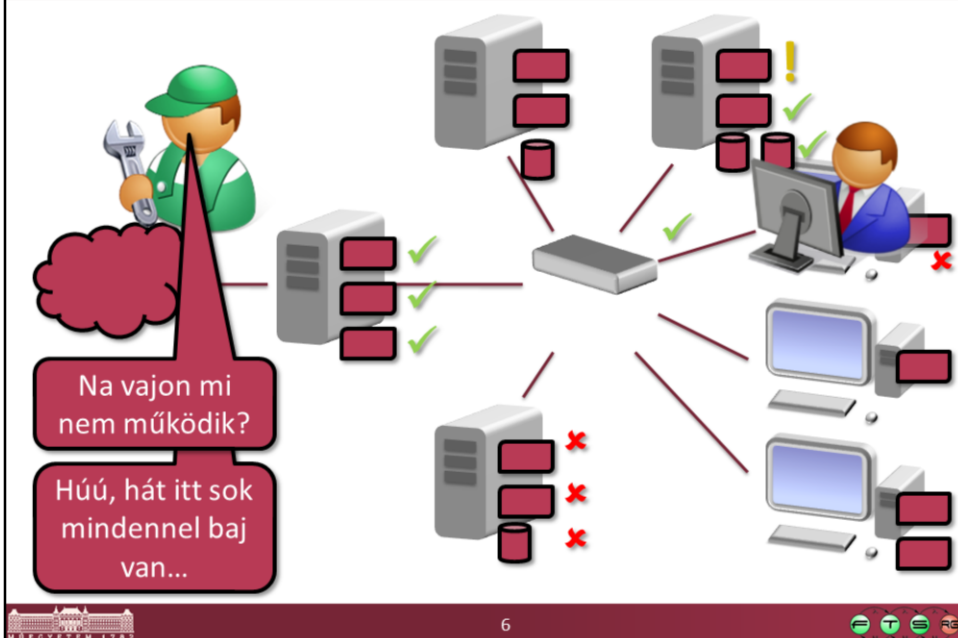
Reakció hibajelenség esetén



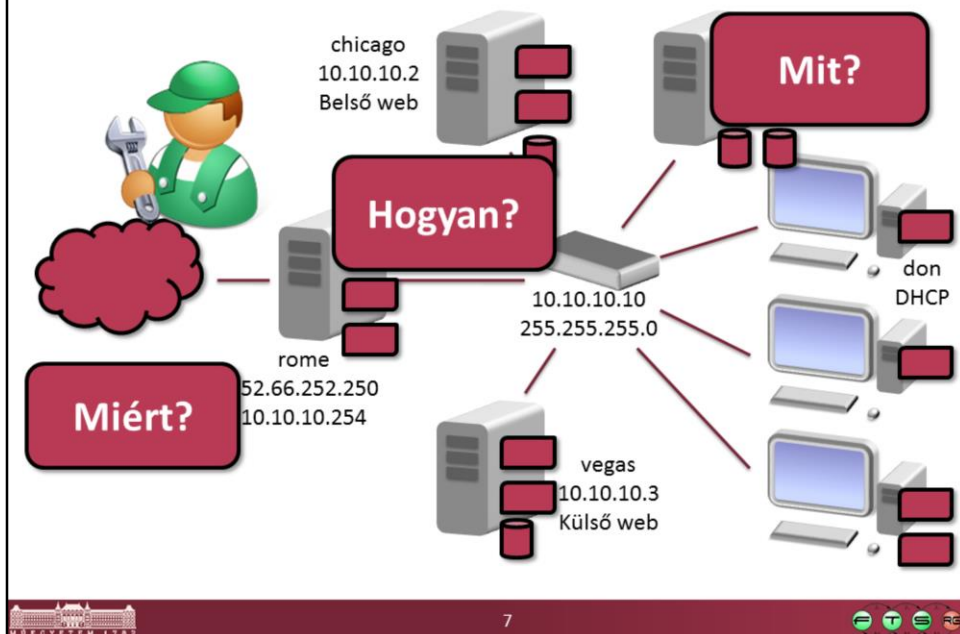
Reakció hibajelenség esetén



Reakció hibajelenség esetén



„Kézbentartott” rendszer



Miért?

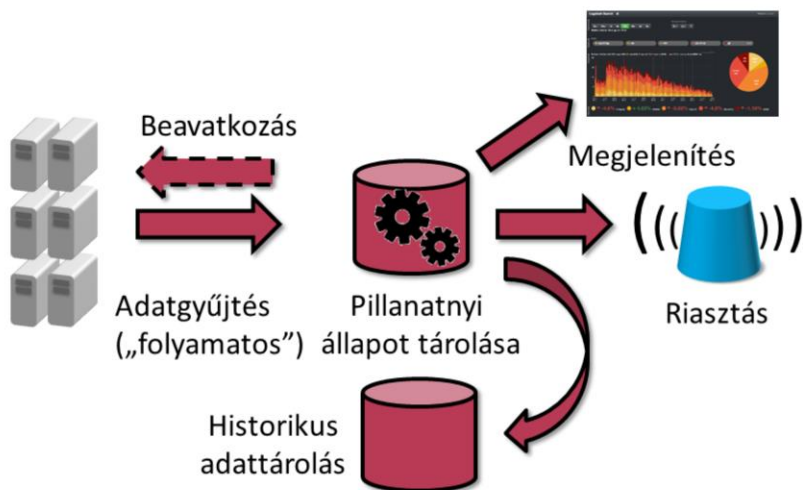
Monitorozás fogalma, részfeladatai, céljai

Alapfogalmak (ITIL)

*„**Monitoring** refers to the activity of observing a situation to detect changes that happen over time.”*

A **monitorozás** valamely „helyzet” megfigyelése, mely során az időbeni változásokat kívánjuk érzékelni.

Rendszermonitorozás részei



Monitorozás jellemzői (ITIL) ((()))

- Fontos CI-k és tevékenységek megfigyelése
- Meghatározott feltételek teljesülése → riasztás
- Megfelelőség ellenőrzése:
 - Rsz.-komponensek teljesítménye/kihasználtsága
 - Normálistól eltérő tevékenységek/tevékenységi szintek
 - Nem engedélyezett változtatások
 - Eljárásrendek
 - „Szolgáltatások” minősége
 - KPI-k



*„**Reporting** refers to the analysis, production and distribution of the output of the monitoring activity.”*

A **jelentéskészítés** a monitorozás kimenetének analízisét, „eredményének” előállítását és az eredmények megfelelő terítését fedí.

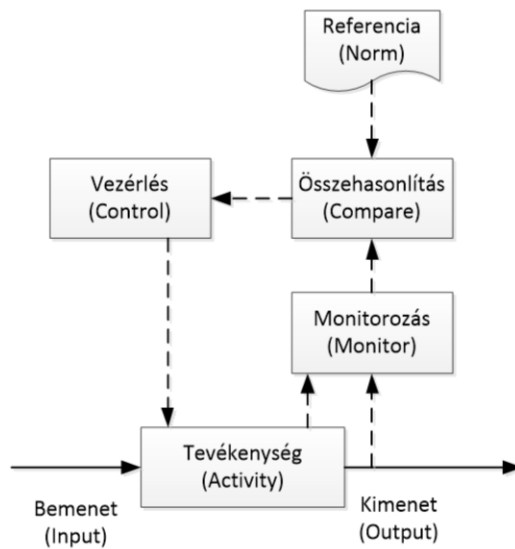


*„**Control** refers to the process of managing the utilization or behaviour of a device, system or service. [...]”*

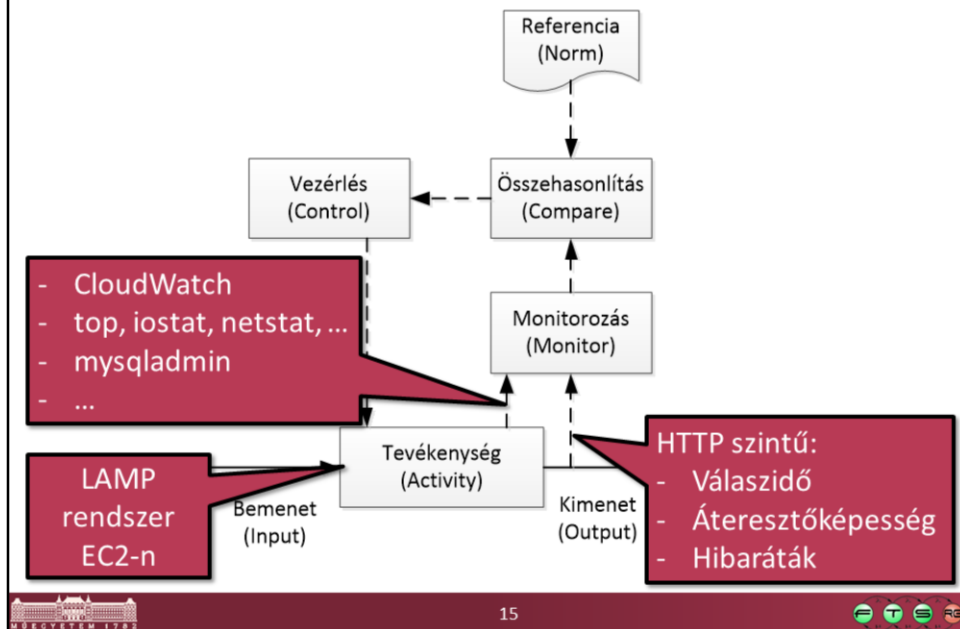
A **vezérlés** egy eszköz, rendszer vagy szolgáltatás kihasználtsága vagy viselkedése menedzselésének a folyamata.

Ugyanaz a kör, mint szabályozástechnikában: a tanult megfigyelhető és vezérelhető jelekről tanultakat itt analóg módon lehet alkalmazni.

„Monitor Control Loop” (ITIL)

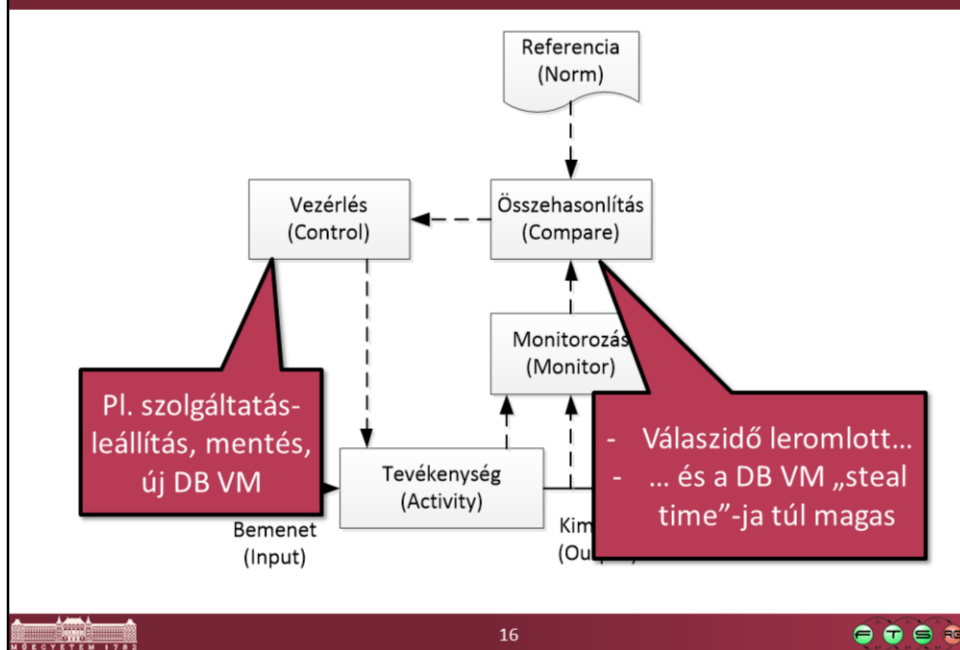


Példa: „Monitor Control Loop”



„LAMP – development stack combines the power of Debian Linux, Apache HTTP Server, MySQL, and PHP to provide an industry-standard open-source solution for rapid development of web applications.”

Példa: „Monitor Control Loop”



Honnan van a referencia?

1. Ökölszabály (környezetből, szakterületből származó tudás), pl. „a CPU_Ready ne menjen 5% fölé”
2. Azt is a monitorozott adatokból találjuk ki, pl. az átlagos CPU használat 30%, akkor jöjjön riasztás mondjuk már 60%-nál.

Monitorozás és egyéb folyamatok

- Érezhető a kapcsolat egyéb folyamatokkal
 - Kapacitástervezés
 - Eseménykezelés
 - Konfiguráció-menedzsment
 - „Proaktív Probléma-Menedzsment”
 - ...
- (De ezekkel most nem foglalkozunk)

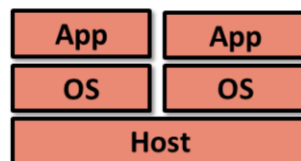
Rendszermonitorozás: állapotkép fenntartása

- Infrastrukturális komponensek és szolgáltatások működőképességéről
- Terhelésről, erőforrások kihasználtságáról
- Topológiáról, konfigurációról
 - Kapcsolat a konfiguráció-menedzsmenttel!
- (Elosztott) feladat-végrehajtás állapotáról
- (Adat)biztonságról

Mit?

Tipikus metrikák

Felügyelt metrikák: példák



Hypervisor (ESX)

CPU Statistics

Level	Counter name in API	Description	Unit
1	cpu.ready.summation	Ready time is the time spend waiting for CPU(s) to become available in the past update interval.	millisecond
1	cpu.usagemhz.average	The CPU utilization. The maximum possible value here is the frequency of the processors times the number of cores. As an example, a VM using 4000 MHz on a system with four 2 GHz processors is using 50% of the CPU ($4000 / (4 * 2000) = 0.5$)	megaHertz
1	cpu.usage.average	The CPU utilization. This value is reported with 100% representing all processor cores on the system. As an example, a 2-way VM using 50% of a four-core system is completely using two cores.	percent
2	cpu.reservedCapacity.average	CPU Reserved Capacity	megaHertz
2	cpu.idle.summation	CPU Idle	millisecond
2	cpu.swapwait.summation	Swap wait time is time that the world spent waiting for memory to be swapped in. When the VM is waiting for memory, it is not doing work.	millisecond
3	cpu.system.summation	System time is the time spent in VMkernel during the last update interval. This does not include guest code execution.	millisecond
3	cpu.waits.summation	Wait time is the time spent waiting for hardware or VMkernel lock thread locks during the last update interval.	millisecond
3	cpu.extra.summation	CPU extra is the time above the statically calculated entitlement. Entitlement is the share of processing time that a VM should get as a result of its vCPU count and assigned shares. <i>You should not use or care about this counter in any of your own analysis.</i>	millisecond
3	cpu.used.summation	CPU Used	millisecond
3	cpu.guaranteed.latest	Guaranteed time is reported as the amount of the reservation time that the VM used in the past update interval. As an example, if 2000 MHz have been reserved for the VM on an four-way, 2 GHz host, that's 25% of the CPU resource. In a 20s update interval, there are 80,000 ms available on this four-way system. That means 20,000 ms of time has been reserved. If a VM used only half of its available cycles, the guaranteed time is 10,000 ms.	millisecond
4	cpu.usage.none	CPU Usage (None)	percent
4	cpu.usage.minimum	CPU Usage (Minimum)	
4	cpu.usage.maximum	CPU Usage (Maximum)	
4	cpu.usagemhz.none	CPU Usage in MHz (None)	
4	cpu.usagemhz.minimum	CPU Usage in MHz (Minimum)	
4	cpu.usagemhz.maximum	CPU Usage in MHz (Maximum)	

És ez csak a CPU, csak a hoszt

21

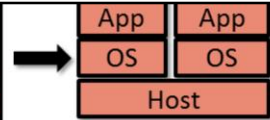
<https://communities.vmware.com/docs/DOC-5600>

Hypervisor CPU (ESX)

- Mintavételezési ablak!
 - Élő adatok, alapértelmezett: 20s
 - Aggregáló függvények!

- Néhány példa
 - usage/usagemhz: AVG, %/MHz
 - idle: SUM, ms
 - swapwait: SUM, ms
 - system (VMkernel!): SUM, ms
 - ready: SUM, ms

Operációs rendszer



ikocsis@morgo:~

```

top - 00:34:26 up 21 days, 13:54, 4 users, load average: 0.00, 0.01, 0.00
Tasks: 178 total, 1 running, 177 sleeping, 0 stopped, 0 zombie
Cpu(s):  0.1%us,  0.1%sy,  0.0%ni, 99.8%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4053280k total, 3741816k used, 311464k free, 92980k buffers
Swap:  2097144k total,  0k used, 2097144k free, 3313664k cached
    
```

FID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
9	root	20	0	0	0	0	S	5.3	0.0	92:59.03	ksftirqd/1
1266	root	20	0	13544	808	652	S	0.3	0.0	78:01.95	lldpad
31226	ikocsis	20	0	17116	1348	976	R	0.3	0.0	0:00.21	top
1	root	20	0	21320	1404	1124	S	0.0	0.0	1:56.05	init
2	root	20	0	0	0	0	S	0.0	0.0	0:04.99	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:09.89	...
4	root	20	0	0	0	0	S	0.0	0.0	512:56.67	...
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.39	...
6	root	RT	0	0	0	0	S	0.0	0.0	0:11.45	...
7	root	RT	0	0	0	0	S	0.0	0.0	0:23.90	...
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.39	...
10	root	RT	0	0	0	0	S	0.0	0.0	0:11.80	...
11	root	RT	0	0	0	0	S	0.0	0.0	0:23.22	...
12	root	RT	0	0	0	0	S	0.0	0.0	0:00.39	...
13	root	20	0	0	0	0	S	0.0	0.0	686:50.41	...
14	root	RT	0	0	0	0	S	0.0	0.0	0:10.13	...
15	root	RT	0	0	0	0	S	0.0	0.0	0:07.29	...

Task Manager

File Options View

Processes Performance App history Start-up Users Details Services

CPU Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz

7% Utilization

7% 2.39 GHz

Memory 47/11.9 GB (39%)

Disk 0 (C:) 0%

Ethernet Not connected

Bluetooth Not connected

WiFi Not connected

Ethernet 0 B/s 0 Kbps

60 seconds

Utilization Speed Maximum speed: 2.39 GHz

7% 2,39 GHz Sockets: 1

Processes Threads Handles Logical processors: 4

89 1198 35504 Virtualizations: Enabled

Up time 5:22:50:37

L1 cache: 128 KB

L2 cache: 512 KB

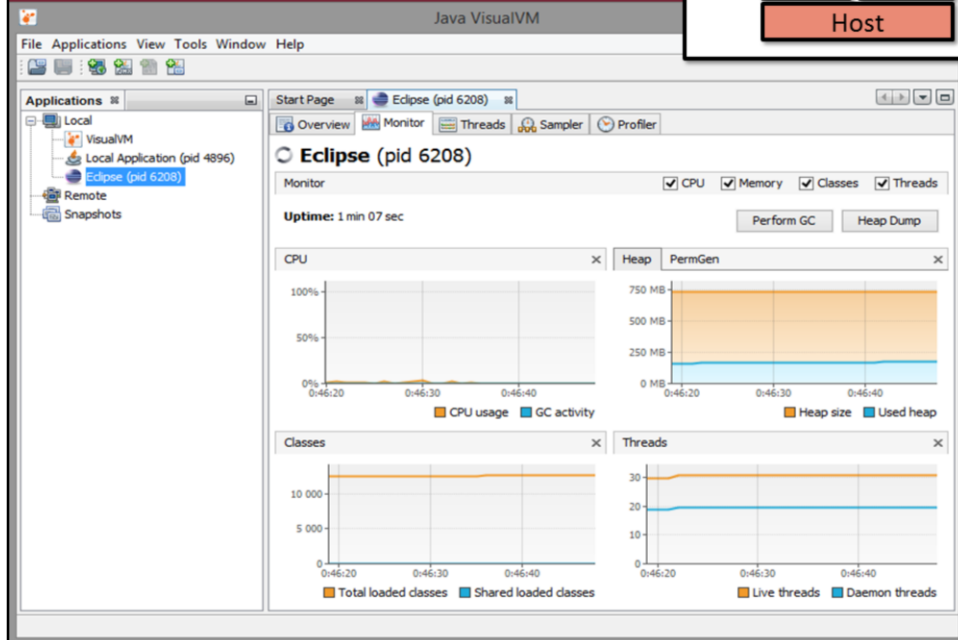
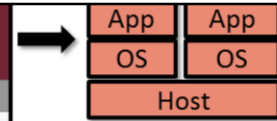
L3 cache: 3.0 MB

Feuer details Open Resource Monitor

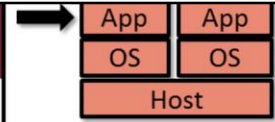
MSÉGYETEM 1789

23

middleware – példa: JVM

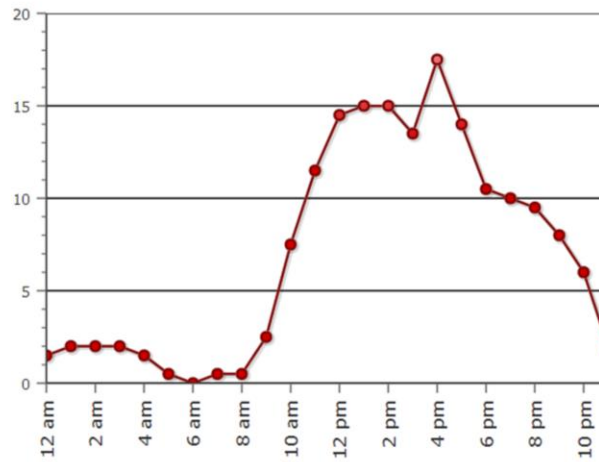


Alkalmazási szint – példa: VCL



Reservations by Hour

(Active reservations during given hour averaged over selected dates)



Hogyan?

Monitorozó rendszerek fajtái

Monitorozás típusai (ITIL)

- **Aktív vagy passzív**

- Beavatkozik-e a monitorozó a rendszerbe, vagy csak megfigyeli annak működését?

- **Reaktív vagy proaktív**

- Reakció a hibák után vagy előtt
- Nem mindenképp a monitorozás alá tartozik



Hogy ezt pontosan milyen technikával valósítjuk meg, az például szoftvertechnikák anyag, az összes kommunikációs mintára (observer, publish-subscriber stb.) itt mind megvan az analógia.

Monitorozás típusai (ITIL)

■ Folyamatos vagy kivétel-alapú mérés

- Folyamatos, valós idejű ellenőrzés vagy detektálás és jelentés „kivételes helyzetek” esetén
- A folyamatos lehet jóval költségesebb
- Hibrid: triggerelt monitorozás → adott kivétel után folyamatos



„Continuous vs. Exception-Based Measurement”

Adatgyűjtés



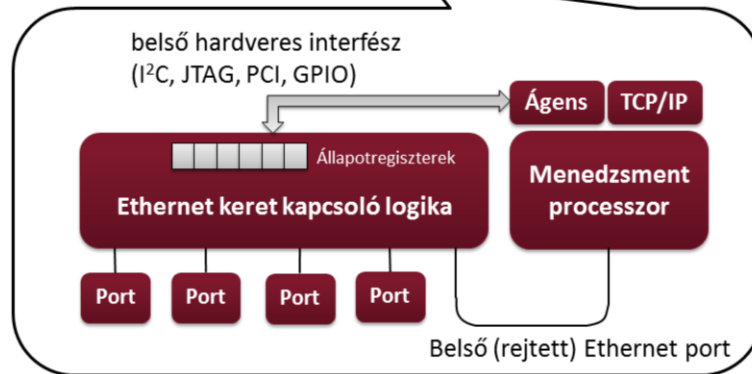
Adatgyűjtés megvalósítása

- Jellegzetes követelmény:
 - A rendszerünk nagy, sok különálló elemből áll
 - Az adatokat hálózaton keresztül olvassuk le
- Kulcselem: **ágens**
 - Kis beépülő komponens minden berendezésbe, célja:
 - adatszolgáltatás valamilyen (hálózati) interfészen
 - értesítés különféle események bekövetkezéséről
 - egyszerű beavatkozások elvégzése

Adatgyűjtés megvalósítása hardverben

Lásd még: IPMI, Intel vPro,
IBM BladeCenter
Management Module, ...

Berendezés
pl.: Ethernet switch



Adatgyűjtés megvalósítása szoftverben I.

Jellemző alapesetek:

- **Olyan szoftver komponenst akarunk megfigyelni, ami nincs erre felkészítve**
 - Az ágens külön folyamat az operációs rendszeren
 - Olyan hívásokat végezhet el, ami csak egy gépen futó folyamatok között lehetséges (de a belső adatszerkezetekhez többnyire nem férünk hozzá)
 - Az operációs rendszer segítségével követi a megfigyelt folyamatot (futási állapot, létrehozott állományok tartalma, erőforráshasználat, stb.)
- Az ágens integrált része a szoftvernek

Adatgyűjtés megvalósítása szoftverben I.

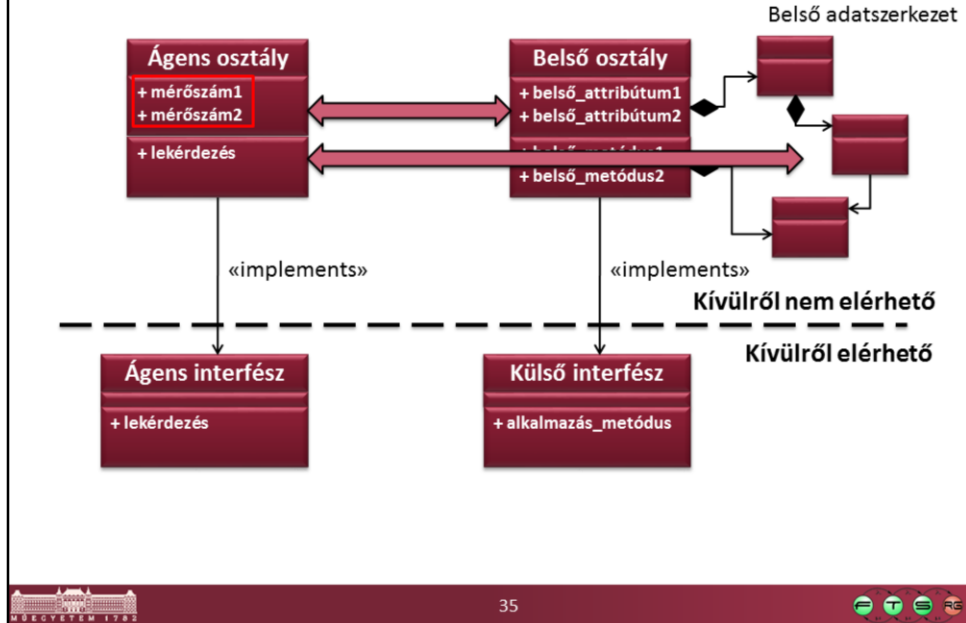


Adatgyűjtés megvalósítása szoftverben II.

Jellemző alapesetek:

- Olyan szoftver komponensst akarunk megfigyelni, ami nincs erre felkészítve
- **Az ágens integrált része a szoftvernek**
 - Hozzáférünk a belső adatszerkezetekhez
 - Közvetlenül végezhetünk függvényhívásokat
 - Forráskód *instrumentálás* (mérő, adatgyűjtő hívások elhelyezése a forráskódban) lehetséges
 - A lényeg: a belső mérési lehetőségeket kívülről is elérhetővé kell tenni

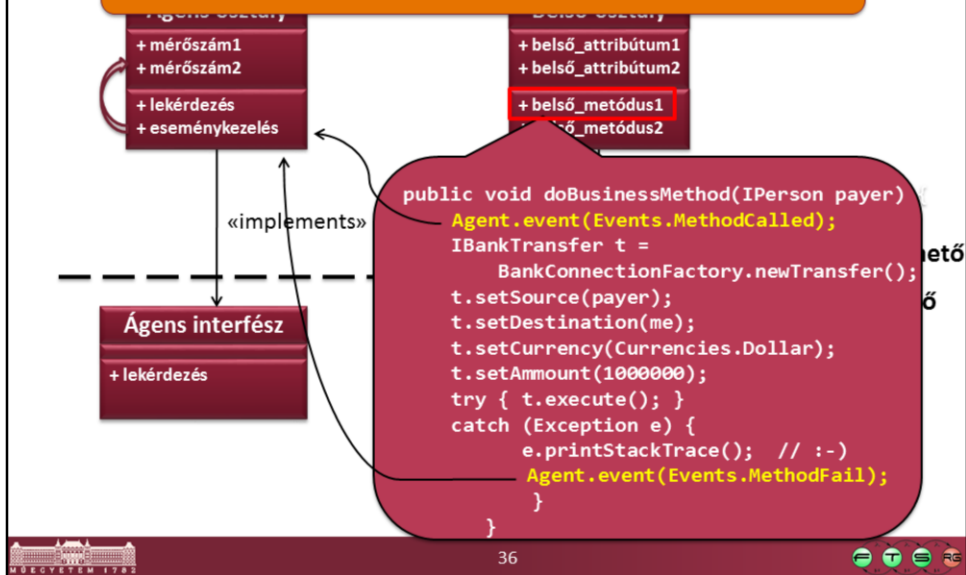
Hozzáférés belső adatszerkezethez



(Demonstrációs ábra, ékezet és szóköz valódi modellben ne legyen osztály- és attribútumnévben!)

Forráskód instrumentáció

Bővebben: felügyeletre tervezés előadás



Ágens lekérdezési interfész

- Hogyan kérdezzük le az ágenstől a mért adatokat?
- Jó lenne...
 - hálózaton keresztül
 - szabványos interfész, protokoll
 - Egységesen: gyártók, készülékek, szoftver/hardver
 - Adatok széles skálájának támogatása
 - ha azt is le tudnánk kérdezni, hogy pontosan miket lehet lekérdezni az ágenstől

Konfigurációmenedzsment: hasonlóság!

Jellegzetes alapfunkciók

■ Pillanatnyi értékek

- Skálár mennyiség: CPU kihasználtság, RAM, tárhely telítettség, ...
- Diszkrét értékkészlet: Kiszolgáló-folyamat UP/DOWN/ERROR, ...

■ Összegyűjtött mérési adatok

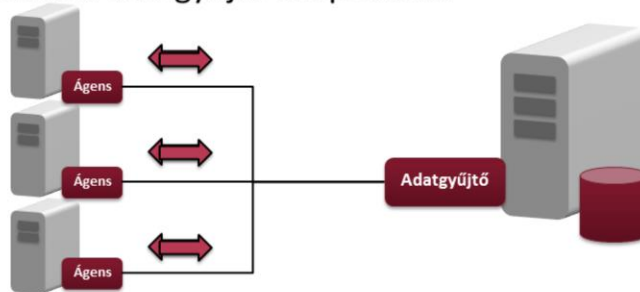
- Skálár mennyiség (pl. kumulatív hálózati forgalom)
 - Eloszlás – pl. kérések válaszidejéről csak „hisztogram”-szintű tudás
 - Utolsó érték/átlag/minimum/maximum/delta
- Napló bejegyzések

■ Értesítés eseményekről

- Diszkrét állapotváltozás (ok → down)
- Határérték túllépés (diszk telítettség >90%)

Ágens lekérdezési interfész

- Ágens interfészek működési elv szerint
 - **Pull** – a központi adatgyűjtő kezdeményezi az ágensok lekérdezést
 - **Push** – az ágens kezdeményezi az adatok elküldését a feliratkozott adatgyűjtő központnak



Egyetemen pl. az előadás push, a ZH előtti konzultáció pull jellegű (a hallgató az adatgyűjtő, az oktató az ágens)

Szabványos protokollok

SNMP

WSDM

Netflow/IPFIX

...

Syslog

ICMP

RMON

CIM-XML

Netconf

JMX

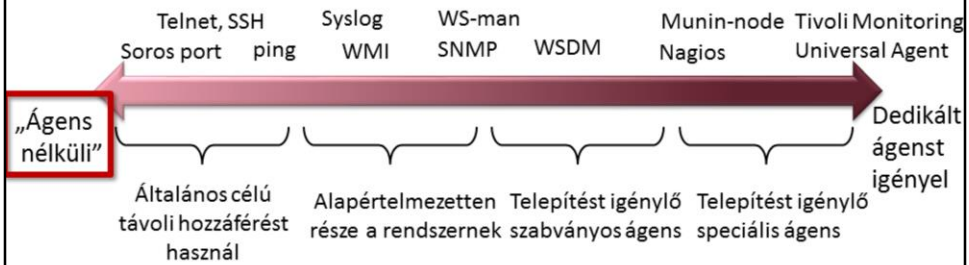
SFlow

WS-Management

„Ágens alapú” és „ágens nélküli” technológiák

Igazából nincs olyan, hogy ágens nélküli

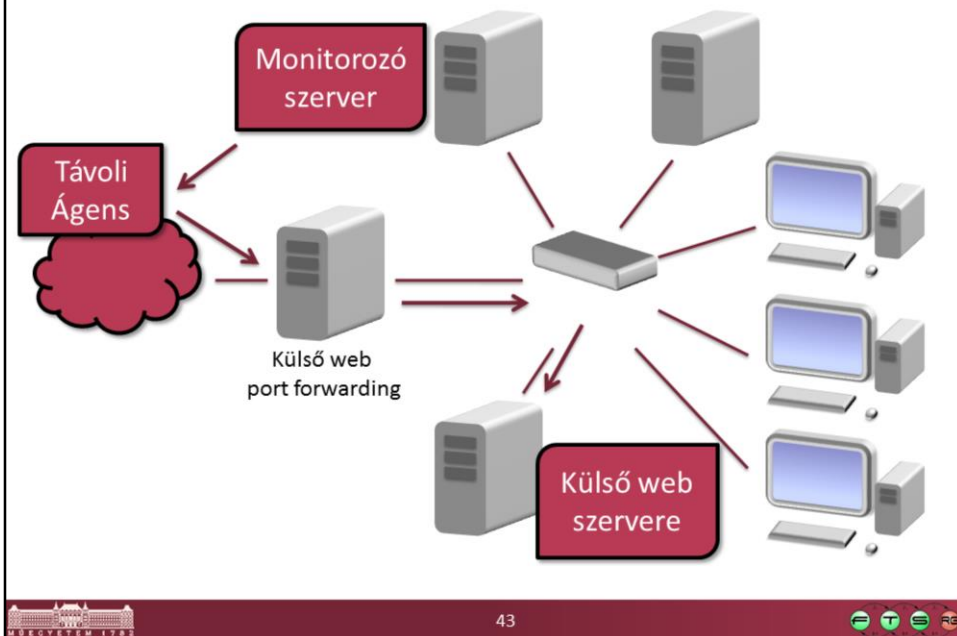
- Parancssoros belépés és értéklekérdezés: távoli hozzáférés kiszolgáló az „ágens”
- Inkább: specializáltság alapján



Szondázás

- Szondázás (probing)
 - Tipikusan „ágens nélküli”: nem „belenézni” akarunk a célrendszerbe, hanem a távolról elérhető szolgáltatását kipróbálni
 - A monitorozó rendszer hálózati kliens szerepben
 - Ilyenkor is kellhet ágens
 - **Szolgáltatás elérési pontról** (Service Access Point) nézve akarunk képet kapni a szolgáltatásról
- **Mellékhatás: hibajelzés több elem hibájára!**

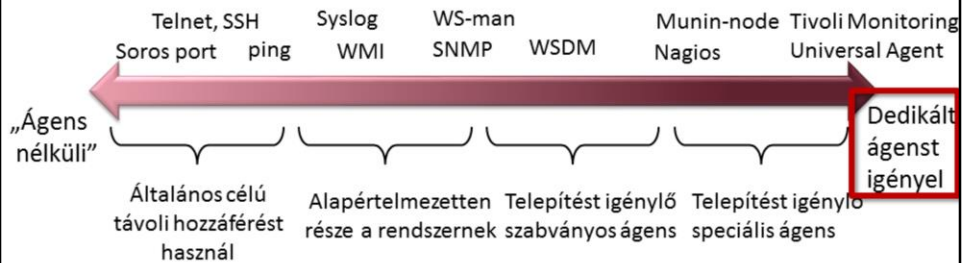
Szondázás példa



A külső webnek, mint szolgáltatásnak a megcélzott szolgáltatás elérési pontja a tűzfalon kívül van -> akkor van „jó” állapotban a külső web, ha kívülről nézve működik. A monitorozó szerver viszont belül van, ezért kell egy távoli ágens, amit megkérhet, hogy kívülről is megnézze a szolgáltatást.

„Ágens alapú” és „ágens nélküli” technológiák

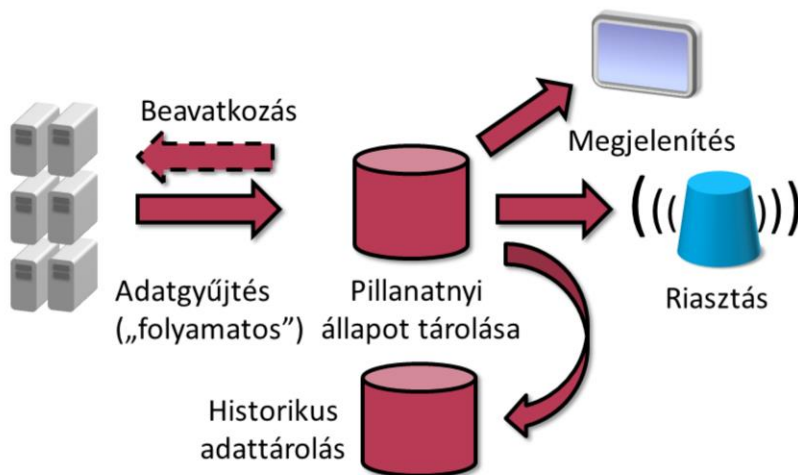
- Igazából nincs olyan, hogy ágens nélküli
 - Parancssoros belépés és értéklekérdezés: távoli hozzáférés kiszolgáló az „ágens”
 - Inkább: specializáltság alapján



Példa monitorozó rendszerre

Nagios

Rendszermonitorozás részei



Monitorozó rendszer példa

- Nagios
 - Free, open source
 - <http://www.nagios.org/>
 - „Kevés” (<100) gépre javasolt
 - Főleg:
 - állapot áttekintésére
 - automatikus riasztásra
- Tactical overview
 - Monitorozott szolgáltatások
 - Grafikus megjelenítés

Monitorozó rendszer példa: Nagios

- Rendelkezésre állás és teljesítmény jelentése
- Naplók és riasztások
- Főleg aktív szondázás
 - kézi konfigurálás...
- Saját ágens protokoll
 - Egyszerű, szöveges, bővíthető shell szkriptekkel
 - Támogat szabványos protokollokat is

Nagios: tactical overview

Nagios®

Tactical Monitoring Overview
 Last Updated: Tue Mar 27 02:48:48 CEST 2012
 Updated every 90 seconds
 Nagios® Core™ 3.2.0 - www.nagios.org
 Logged in as ikocsis

General

- Home
- Documentation

Monitoring

- Tactical Overview
- Service Detail
- Host Detail
- Hostgroup Overview
- Hostgroup Summary
- Hostgroup Grid
- Servicegroup Overview
- Servicegroup Summary
- Servicegroup Grid
- Status Map
- 3-D Status Map

- Service Problems
 - Unhandled
- Host Problems
 - Unhandled
- Network Outages

Show Host:

- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue

Reporting

- Trends
- Availability
- Alert Histogram
- Alert History

Network Outages

0 Outages

Hosts

3 Down 0 Unreachable 27 Up 0 Pending

1 Unhandled Problems

Services

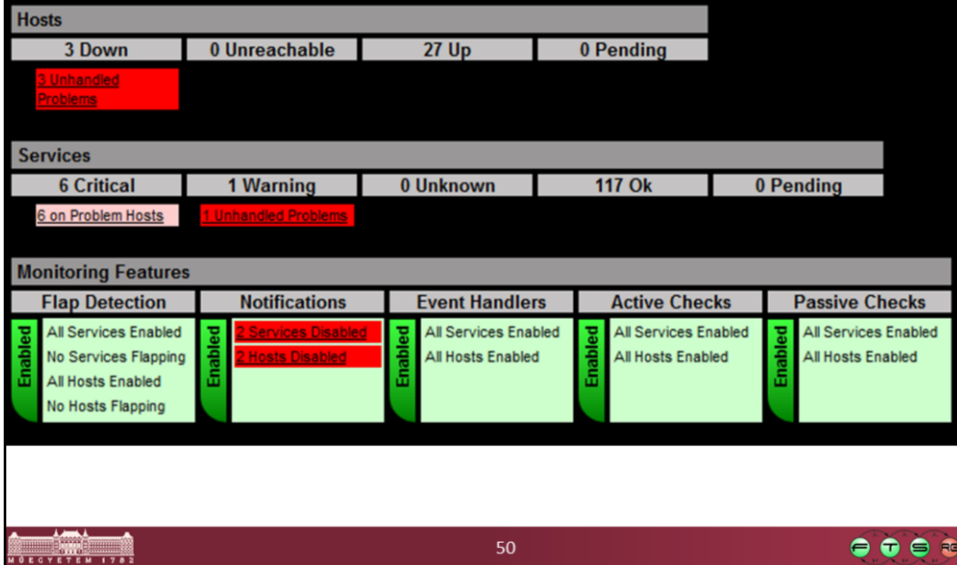
6 Critical 1 Warning 0 Unknown 117 Ok 0 Pending

6 on Problem Hosts 1 Unhandled Problems

Monitoring Features

Flap Detection	Notifications	Event Handlers	Active Checks	Passive Checks
Enabled All Services Enabled No Services Flapping All Hosts Enabled No Hosts Flapping	Enabled Services Disabled 2 Hosts Disabled	Enabled All Services Enabled All Hosts Enabled	Enabled All Services Enabled All Hosts Enabled	Enabled All Services Enabled All Hosts Enabled

Nagios: tactical overview



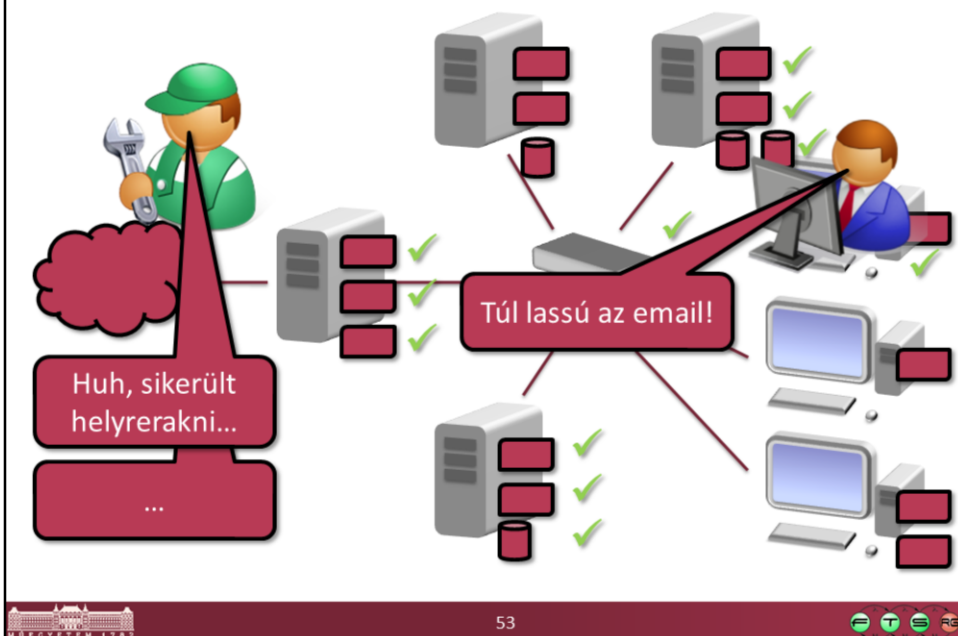
Nagios: service detail

	NRPE PING	OK	2012-03-27 02:49:14	3d 9h 1m 47s	1/4	PING OK - Packet loss = 0%, RTA = 0.47 ms
	PING	OK	2012-03-27 02:49:15	31d 10h 37m 10s	1/4	PING OK - Packet loss = 0%, RTA = 0.23 ms
	ESX Management Console	OK	2012-03-27 02:50:16	27d 10h 40m 46s	1/4	TCP OK - 0.000 second response time on p
	HTTP	OK	2012-03-27 02:46:30	55d 17h 27m 29s	1/4	HTTP OK: HTTP/1.1 301 Moved Permanently
	HTTPS	OK	2012-03-27 02:48:00	55d 17h 26m 35s	1/4	HTTP OK: HTTP/1.1 200 OK - 5293 bytes in
	PING	OK	2012-03-27 02:48:17	55d 17h 23m 12s	1/4	PING OK - Packet loss = 0%, RTA = 0.66 ms
	ESX Management Console	CRITICAL	2012-03-27 02:49:17	128d 2h 29m 24s	1/4	No route to host
	HTTP	CRITICAL	2012-03-27 02:50:19	128d 2h 29m 24s	1/4	No route to host
	HTTPS	CRITICAL	2012-03-27 02:46:30	128d 2h 29m 24s	1/4	No route to host
	PING	CRITICAL	2012-03-27 02:48:00	128d 2h 29m 24s	1/4	CRITICAL - Host Unreachable (152.66.253.1
	ITIM	OK	2012-03-27 02:48:19	53d 10h 46m 30s	1/4	HTTP OK: Status line output matched "HTTP
	PING	OK	2012-03-27 02:49:24	51d 4h 40m 35s	1/4	PING OK - Packet loss = 0%, RTA = 0.27 ms
	SSH	OK	2012-03-27 02:46:30	23d 21h 19m 41s	1/4	SSH OK - OpenSSH_3.9p1 (protocol 1.99)



Adatgyűjtéstől a diagnosztikáig: szondázás

Reakció hibajelenség esetén



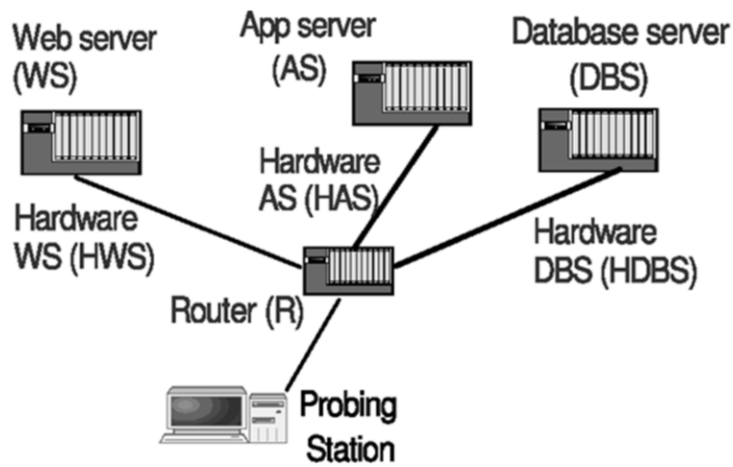
Diagnosztika

- Nem megy a webkiszolgáló. De **miért** nem?
 - Megfelelő megfigyelések kellenek
- Adott hibahatás okának felderítéséhez mit figyeljünk?
 - Pl. egy ESX hoszt több száz valós idejű metrikát definiál magán + VM-ek metrikái
 - Egy operációs rendszer még bonyolultabb lehet
- Hogyan következtessünk a hibaokra?

Diagnosztika

- Hibahatás-detektálás (failure detection): van-e hibahatást okozó jelenség a rendszerben
- Hibaok-lokalizáció (fault localization): a hibahatást kiváltó pontos hibaokok meghatározása
- Szondázás: olyan teszttranzakció, melynek kimenetele több komponens állapotától is függhet
 - Gondoljuk végig: VM-ben futó Apache-re wget távolról
- I. Rish et al. (2005). Adaptive diagnosis in distributed systems. *IEEE transactions on neural networks*, 16(5), 1088-1109.

Függőségek



(Kiterjesztett) függőségi mátrix

Problem Probe	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Egyszeres
hibaok-
feltételezésnél a
hibaaktivációs
kombinációk
NF – No Failure

Szonda
hibaérzékenysége

Probes:

pWS - Web Page access, **pAS** -
pDBS - Database query, **pingR** -
pingWS - ping Web Server, **pingAS** -
server, **pingDBS** - ping Database

!!! Implicit tudás:

- topológia-modell
- Szolgáltatás-függőségi modell
- (Egyszerű) hiba(terjedési) modell

Detektálás/lokalizálás

- Minimális hibadetektáló szondahalmaz választása?

Detektálás/lokalizálás

	W S	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	W S	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	W S	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	W S	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

■ Minimális hibadetektáló szondahalmaz választása?

- Az a minimális szondahalmaz, amire minden oszlopösszeg > 0

	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0

- minimális halmazfedés! 😊
- a „minimum set cover” NP-nehéz ☹️
- De: igen jó heurisztikák



Amit ebből tanultunk algelen: az eldöntési változata a problémának, az ott van a tanult NP-teljesek között.

Detektálás/lokalizálás

- Minimális hibalokalizáló szondahalmaz választása?

Detektálás/lokalizálás

	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Detektálás/lokalizálás

	WS	AS	DBS	R	HWS	HAS	HDBS	NF
pWS	1	1	1	1	1	1	1	0
pAS	0	1	1	1	0	1	1	0
pDBS	0	0	1	1	0	0	1	0
pingR	0	0	0	1	0	0	0	0
pingWS	0	0	0	1	1	0	0	0
pingAS	0	0	0	1	0	1	0	0
pingDBS	0	0	0	1	0	0	1	0

Figyelem: ehhez kell az
egyszeres hibaok
feltételezés!

Detektálás/lokalizálás

- Minimális hibalokalizáló szondahalmaz választása?
 - Az a minimális szondahalmaz, ahol minden hibaok-párt meg tudunk még különböztetni → páronként különböző oszlopok
 - NP-nehéz ☹
 - Szintén jó heurisztikák

Összefoglalás: rendszermonitorozás

