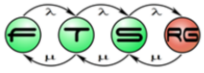
Intelligens rendszerfelügyelet (VIMIA370)

Rendszermonitorozás

Tóth Dániel, Kocsis Imre,
Salánki Ágnes, Micskei Zoltán



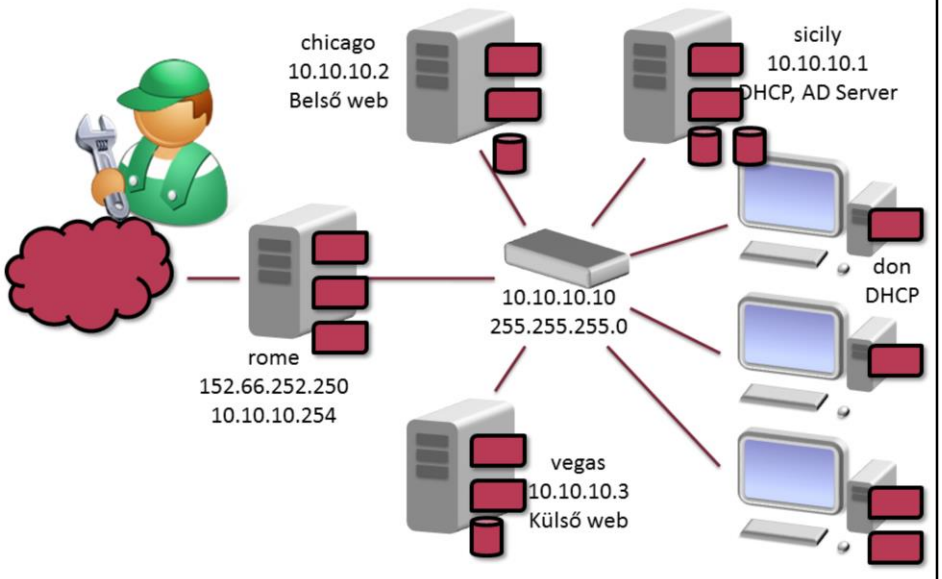
Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Utolsó módosítás: 2016. 04. 10.

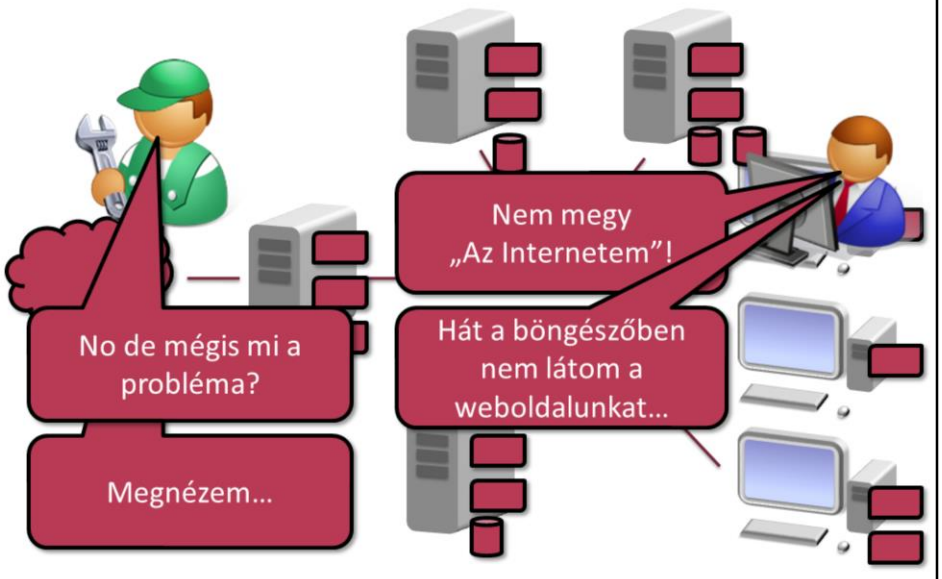
„When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge of it is of a meager and unsatisfactory kind"

Lord Kelvin

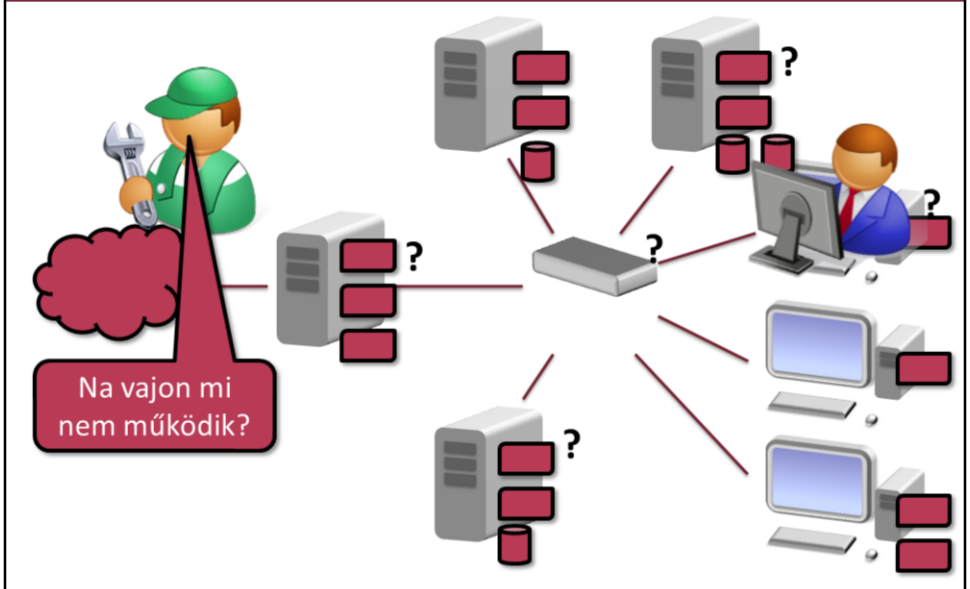
„Kézbentartott” rendszer



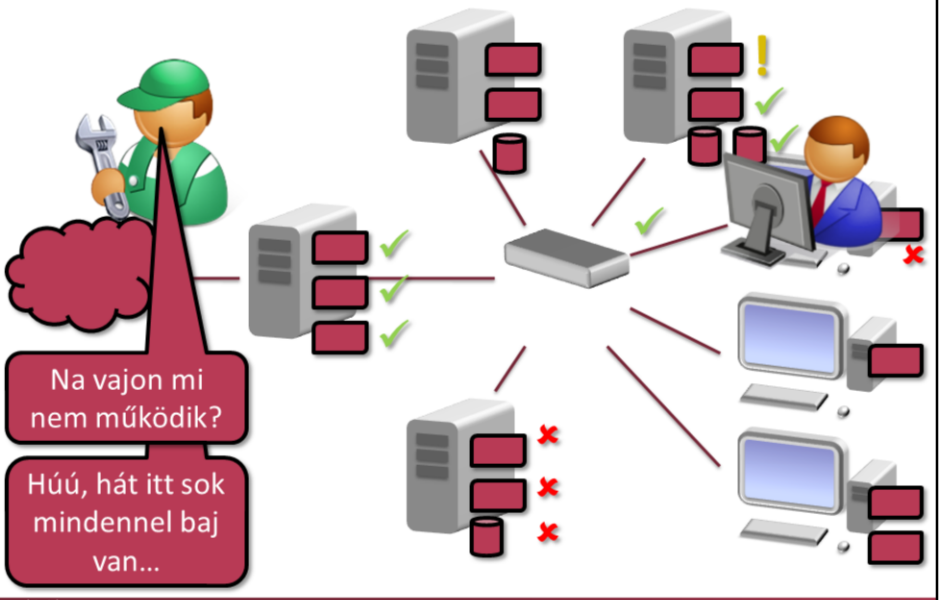
Reakció hibajelenség esetén



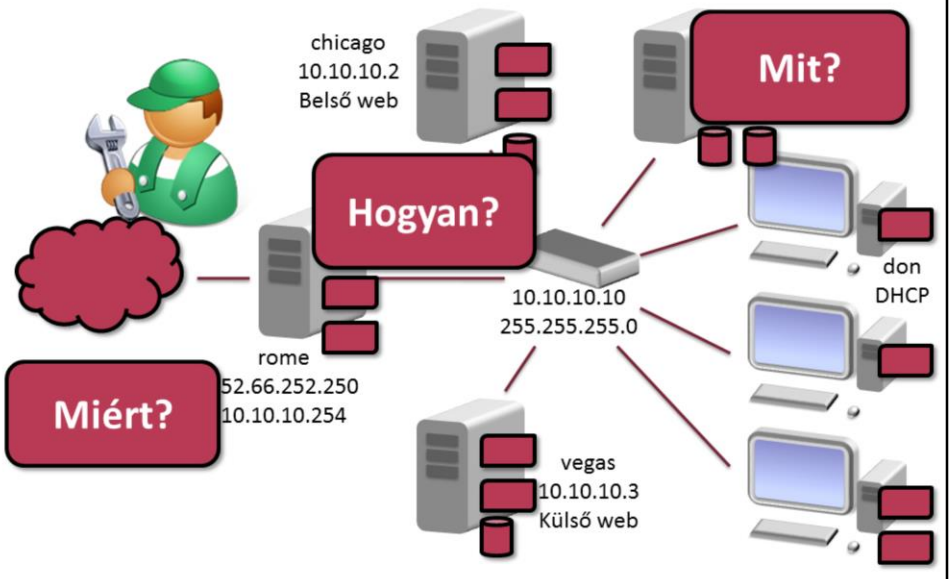
Reakció hibajelenség esetén



Reakció hibajelenség esetén



„Kézbentartott” rendszer



Miért?

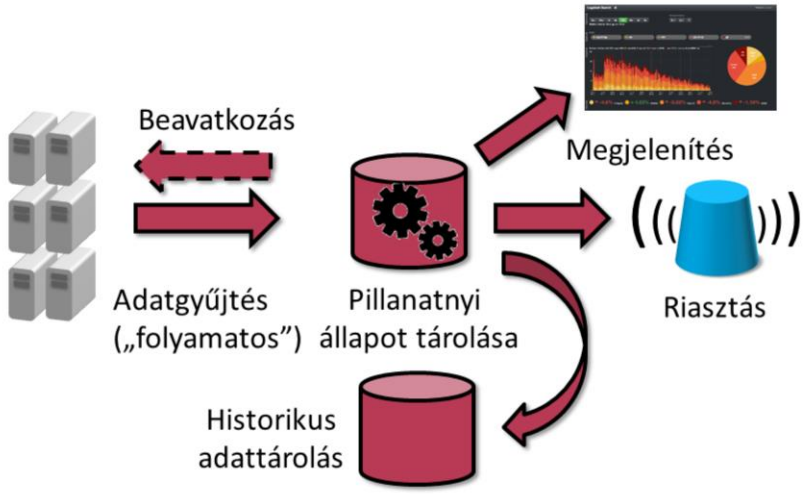
Monitorozás fogalma, részfeladatai, céljai

Alapfogalmak (ITIL)

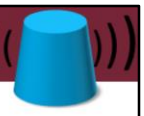
*„**Monitoring** refers to the activity of observing a situation to detect changes that happen over time.”*

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

Rendszermonitorozás részei



Monitorozás jellemzői (ITIL)



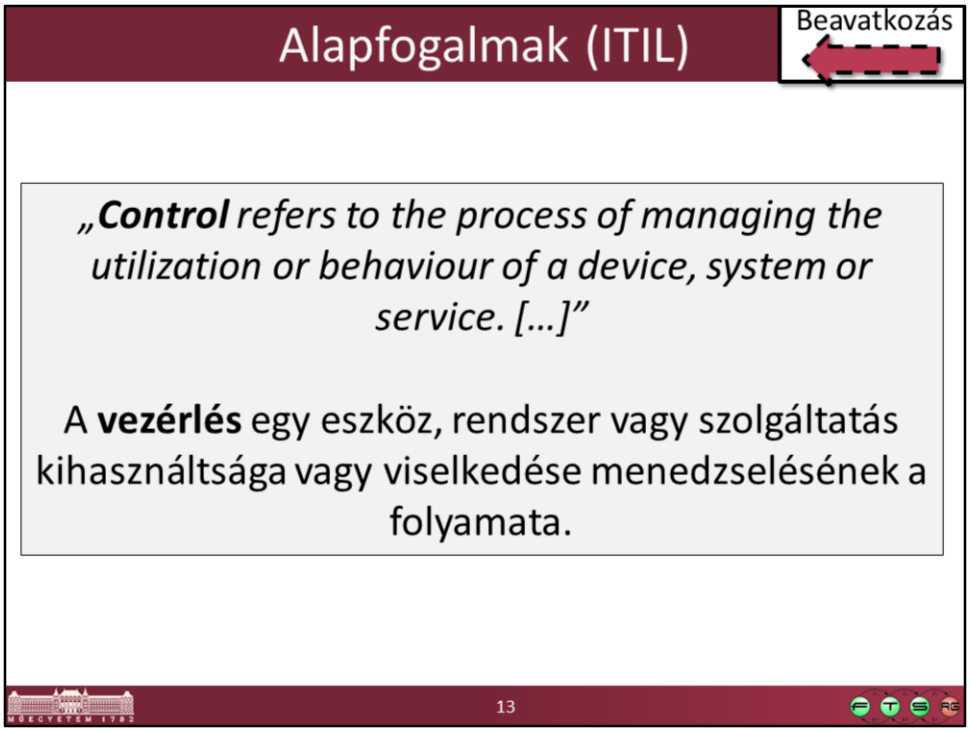
- Fontos CI-k és tevékenységek megfigyelése
- Meghatározott feltételek teljesülése → riasztás

- Megfelelőség ellenőrzése:
  - Rsz.-komponensek teljesítménye/kihasználtsága
  - Normálistól eltérő tevékenységek/tevékenységi szintek
  - Nem engedélyezett változtatások
  - Eljárásrendek
  - „Szolgáltatások” minősége
  - KPI-k

Alapfogalmak (ITIL)

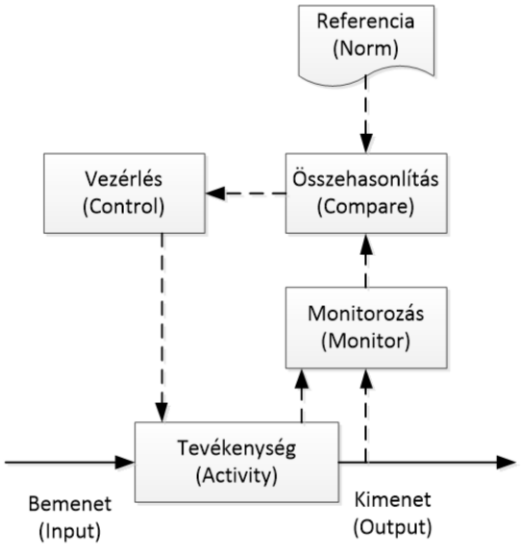
*„**Reporting** refers to the analysis, production and distribution of the output of the monitoring activity.”*

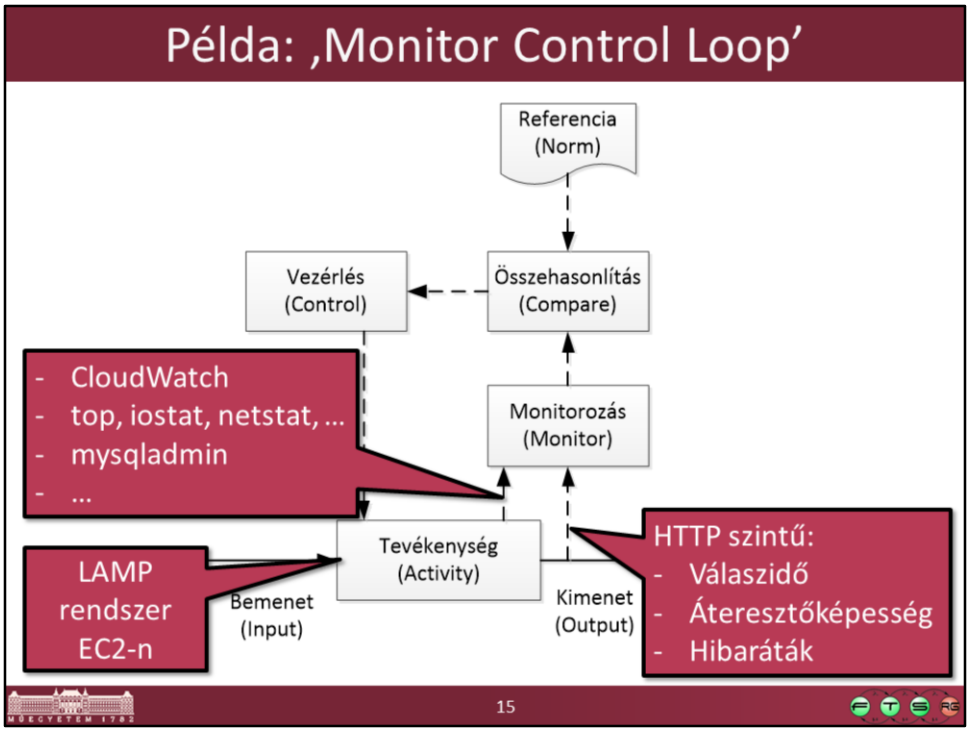
A **jelentéskészítés** a monitorozás kimenetének analízisét, „eredményének” előállítását és az eredmények megfelelő terítését fedi.



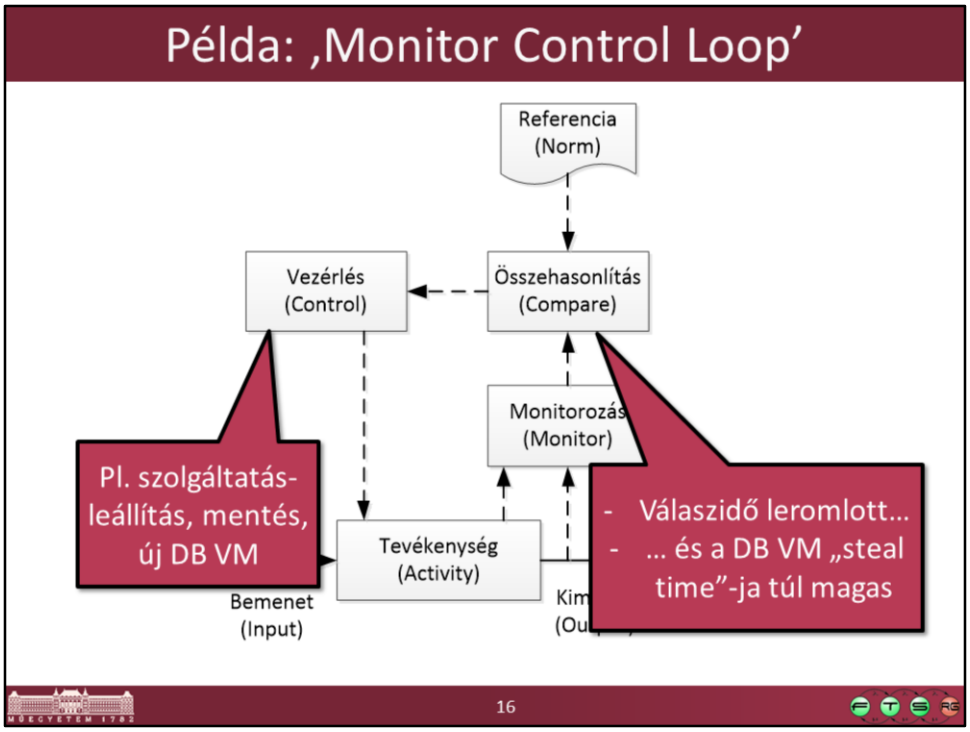
Ugyanaz a kör, mint szabályozástechnikában: a tanult megfigyelhető és vezérelhető jelekről tanultakat itt analóg módon lehet alkalmazni.

‚Monitor Control Loop' (ITIL)





„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."



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)

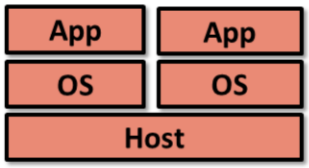
Rendszermonitorozás: állapotkép fenntartása

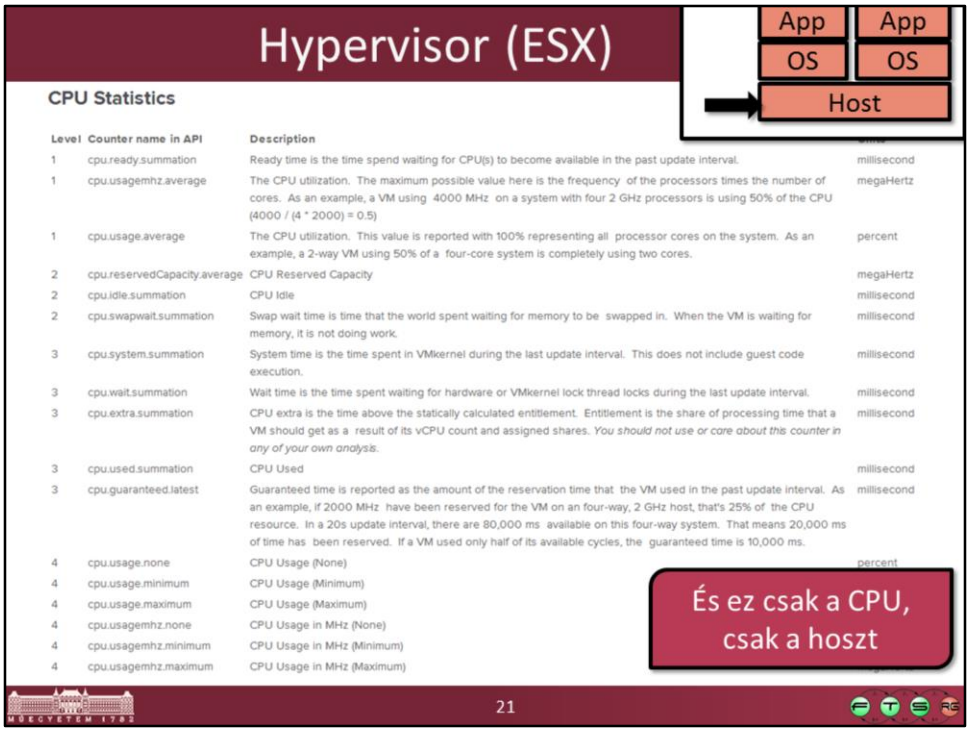
- Infrastrukturális komponensek és szolgáltatások működőképességéről
- Terhelésről, erőforrások kihasználtságáról
- Topológiáról, konfigurációról
  - Kapcsolat a konfiguráció-menedzsmenttel!
- (Elosztott) feladat-végrehajtás állapotáról
- (Adat)biztonságról

Mit?

Tipikus metrikák

Felügyelt metrikák: példák



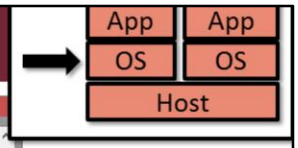


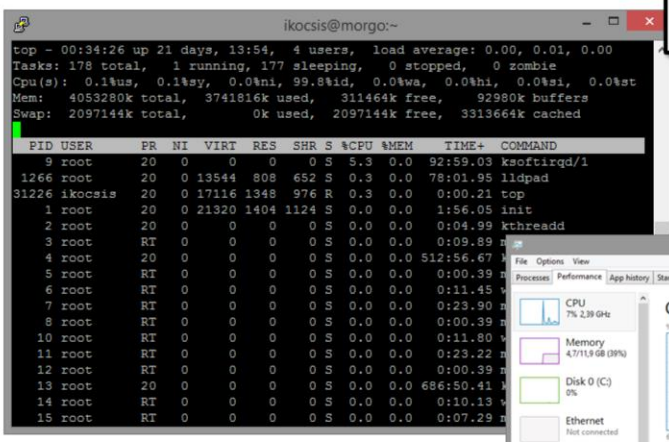
https://communities.vmware.com/docs/DOC-5600

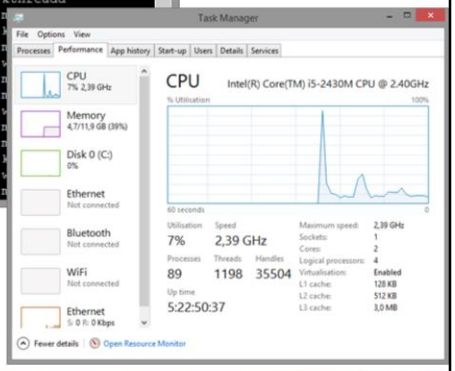
Hypervisor CPU (ESX)

- Mintavételezési ablak!
  - Élő adatok, alapértelemezett: 20s
  - Aggregáló függvények!
- Néhány példa
  - usage/usagemhz: AVG, %/MHz
  - idle: SUM, ms
  - swapwait: SUM, ms
  - system (VMkernel!): SUM, ms
  - ready: SUM, ms

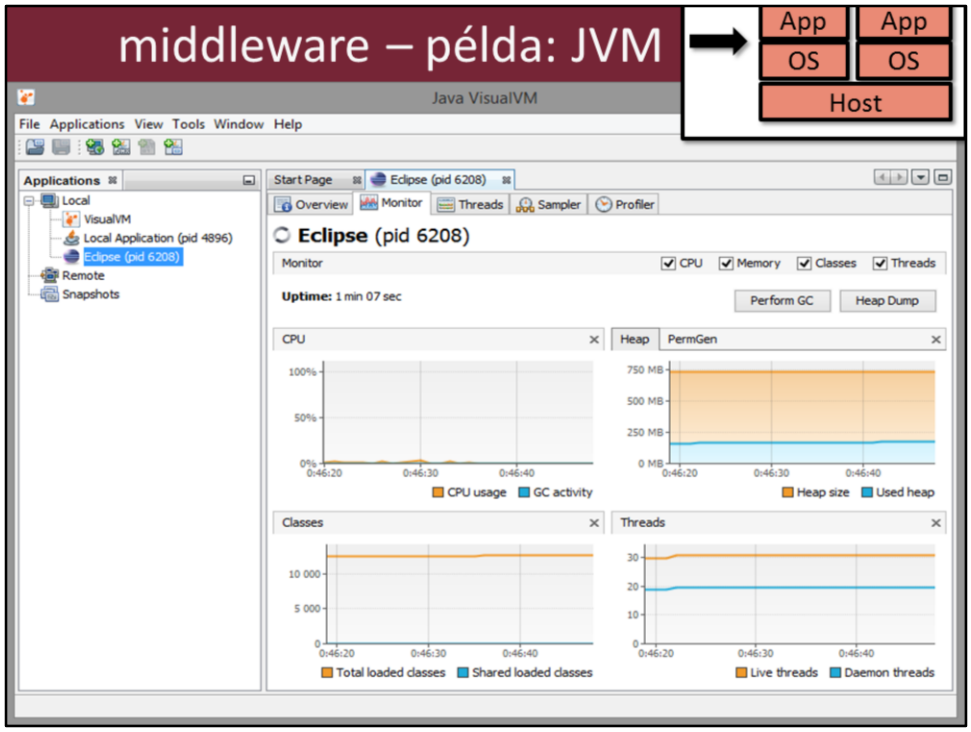
Operációs rendszer



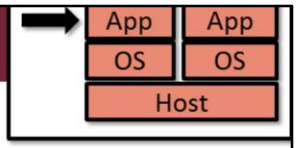




middleware – példa: JVM

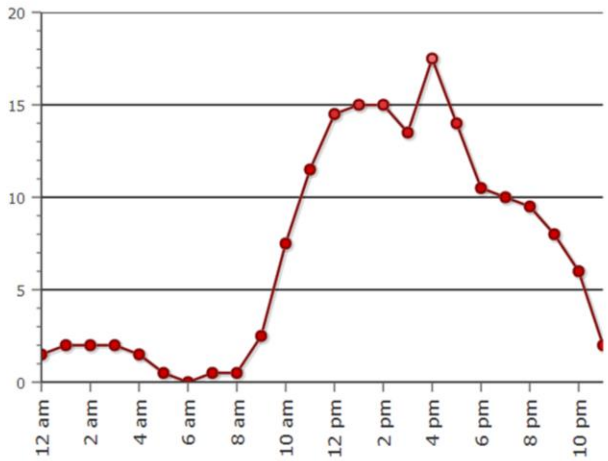


Alkalmazási szint – példa: VCL



Reservations by Hour

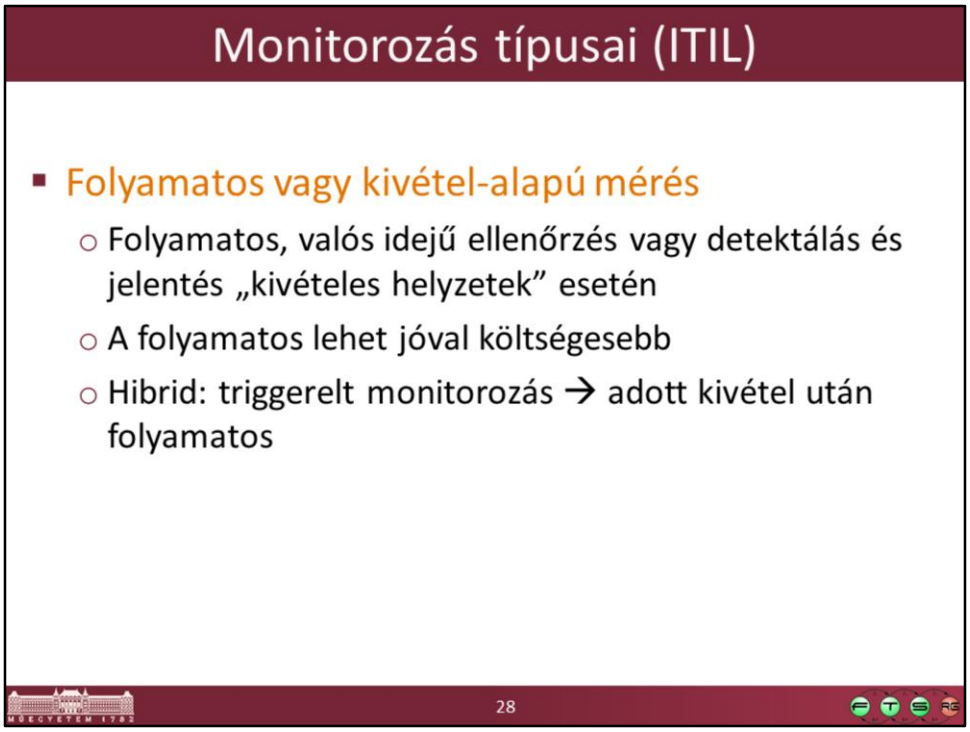
(Active reservations during given hour averaged over selected dates)



Hogyan?

Monitorozó rendszerek fajtái

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



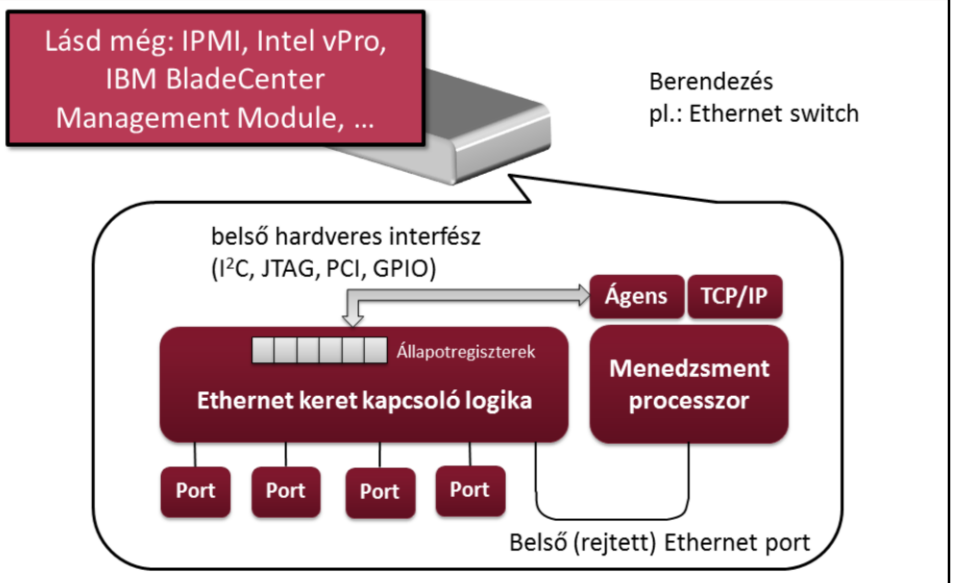
„Continuous vs. Exception-Based Measurement”

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



Adatgyűjtés megvalósítása szoftverben I.

Jellemző alapesetek:

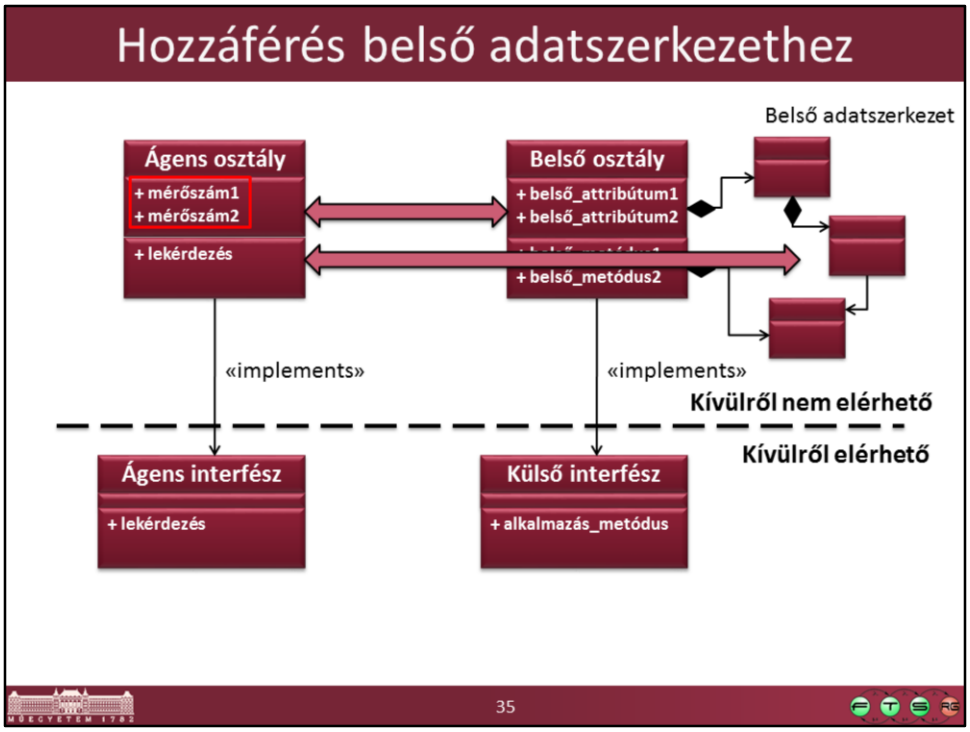
- **Olyan szoftver komponenst akarunk megfigyelni, ami nincs erre felkészítve**
  - Az ágens külön folyamat az operációs rendszeren
  - Olyan hívásokat végezhet el, ami csak egy gépen futó folyamatok között lehetséges (de a belső adatszerkezetekhez többnyire nem férünk hozzá)
  - Az operációs rendszer 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:

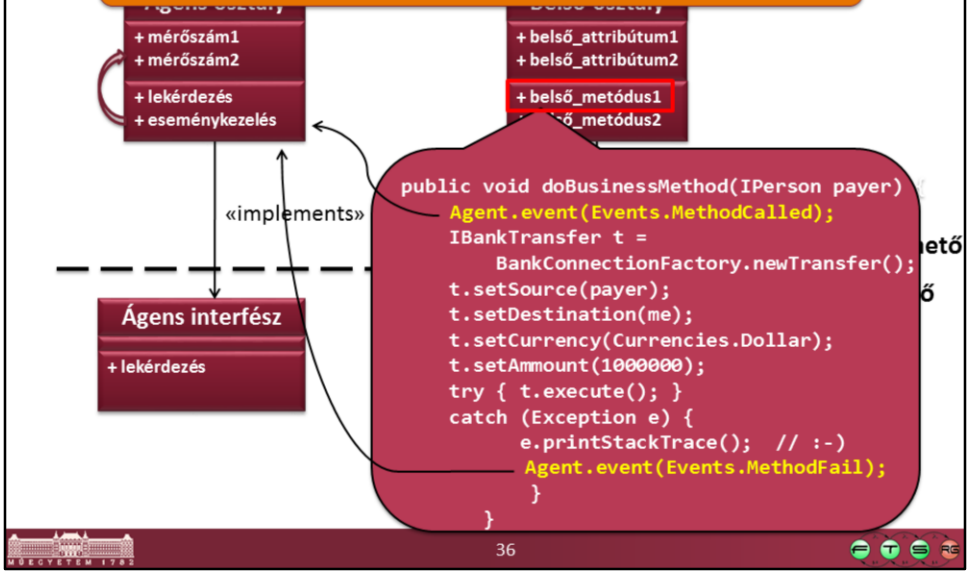
- Olyan szoftver komponenst akarunk megfigyelni, ami nincs erre felkészítve
- **Az ágens integrált része a szoftvernek**
  - Hozzáférünk a belső adatszerkezetekhez
  - Közvetlenül végezhetünk függvényhívásokat
  - Forráskód *instrumentálás* (mérő, adatgyűjtő hívások elhelyezése a forráskódban) lehetséges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



(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

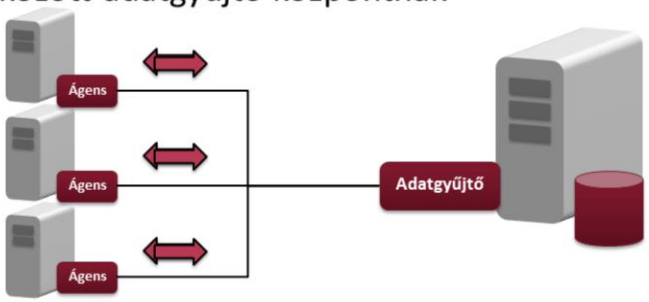
Konfigurációmenedzsment: hasonlóság!

Jellegzetes alapfunkciók

- Pillanatnyi értékek
  - Skalár mennyiség: CPU kihasználtság, RAM, tárhely telítettség, …
  - Diszkrét értékkészlet: Kiszolgáló-folyamat UP/DOWN/ERROR, …
- Összegyűjtött mérési adatok
  - Skalár 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%)

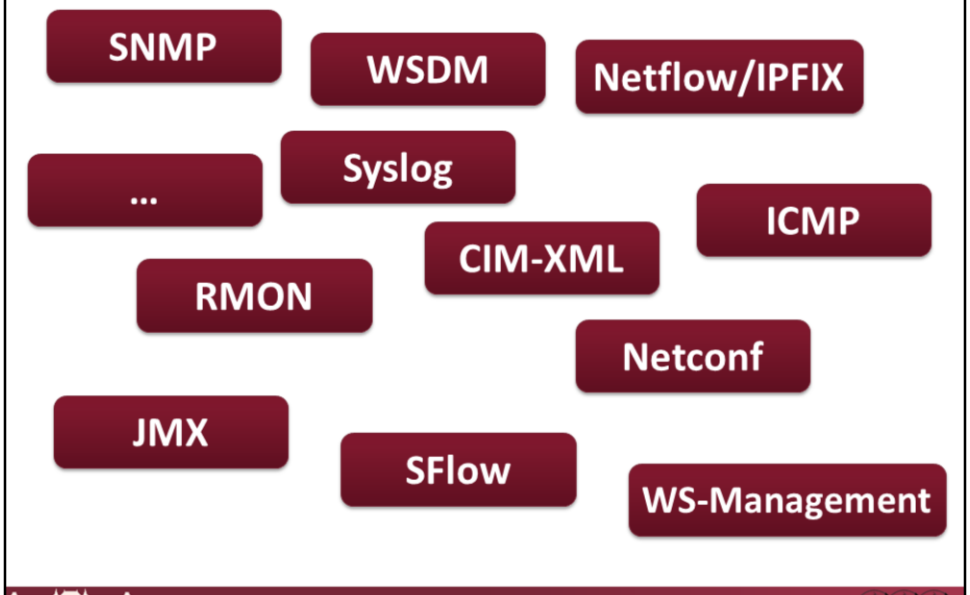
Ágens lekérdezési interfész

- Ágens interfészek működési elv szerint
  - **Pull** – a központi adatgyűjtő kezdeményezi az ágensek 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)

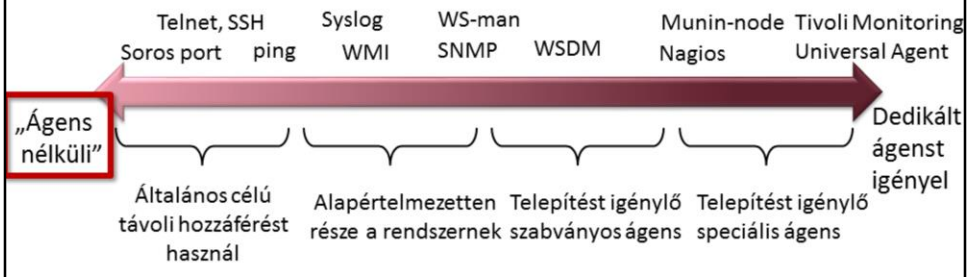
Szabványos protokollok



„Ágens alapú” és „ágens nélküli” technológiák

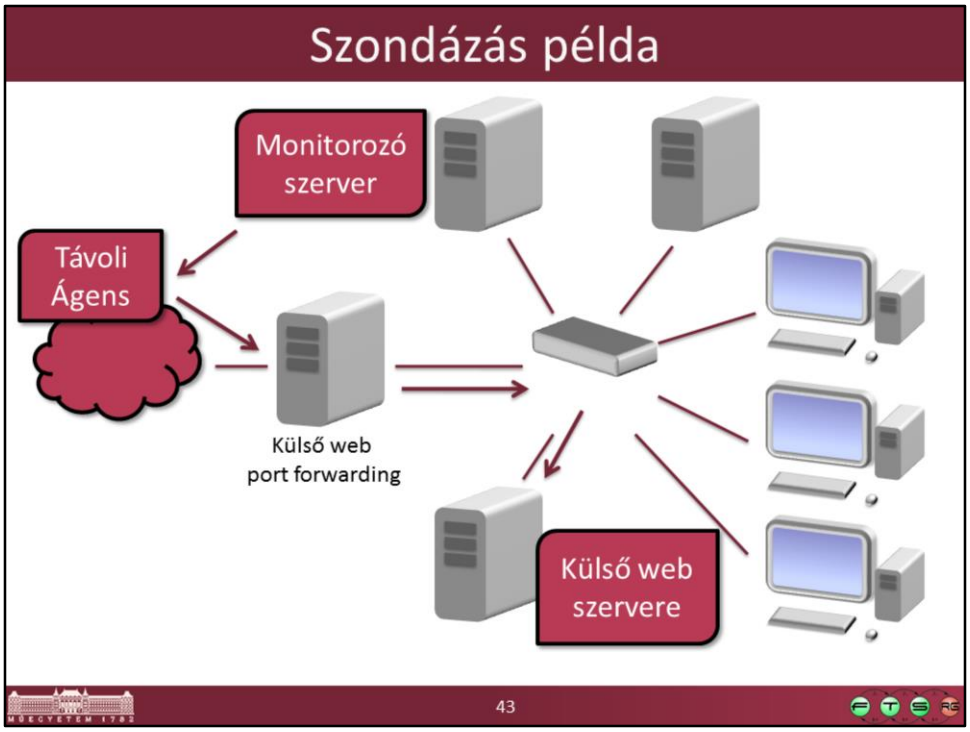
Igazából nincs olyan, hogy ágens nélküli

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



Szondázás

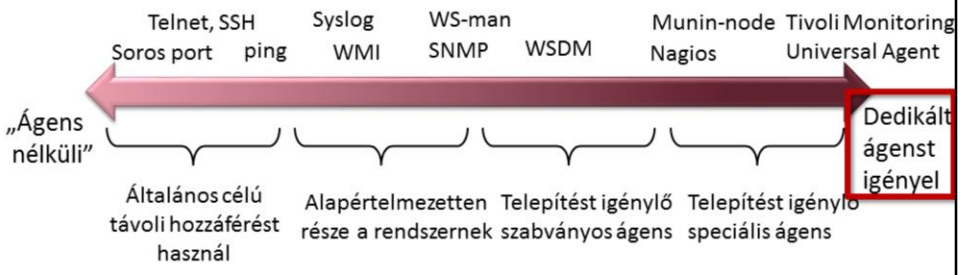
- Szondázás (probing)
  - Tipikusan „ágens nélküli”: nem „belenézni” akarunk a célrendszerbe, hanem a távolról elérhető szolgáltatását kipróbálni
  - A monitorozó rendszer hálózati kliens szerepben
  - Ilyenkor is kellhet ágens
    - **Szolgáltatás elérési pontról** (Service Access Point) nézve akarunk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



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

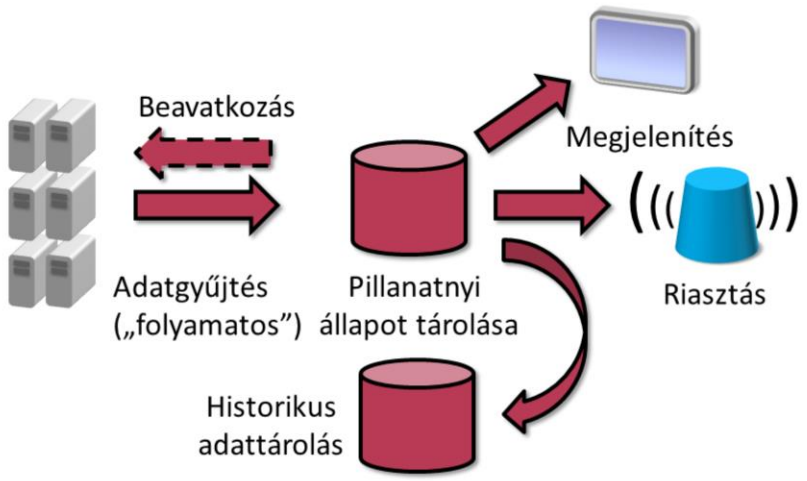
- Igazából nincs olyan, hogy ágens nélküli
  - Parancssoros belépés és értéklekérdezés: távoli hozzáférés kiszolgáló az „ágens”
  - Inkább: specializáltság alapján



Példa monitorozó rendszerre

Nagios

Rendszermonitorozás részei



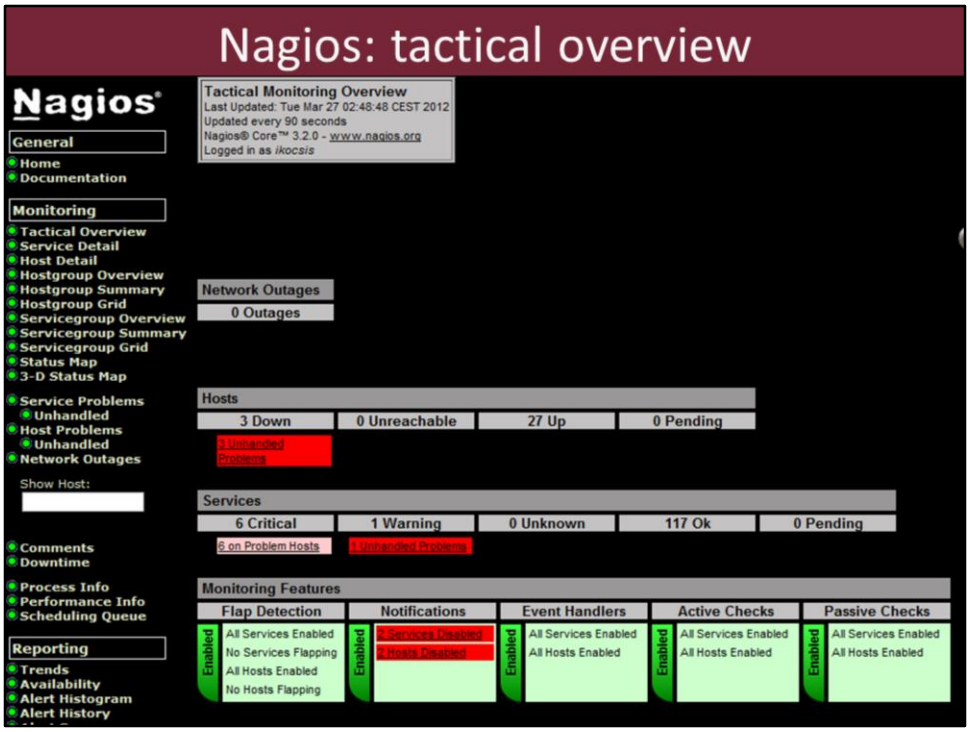
Monitorozó rendszer példa

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

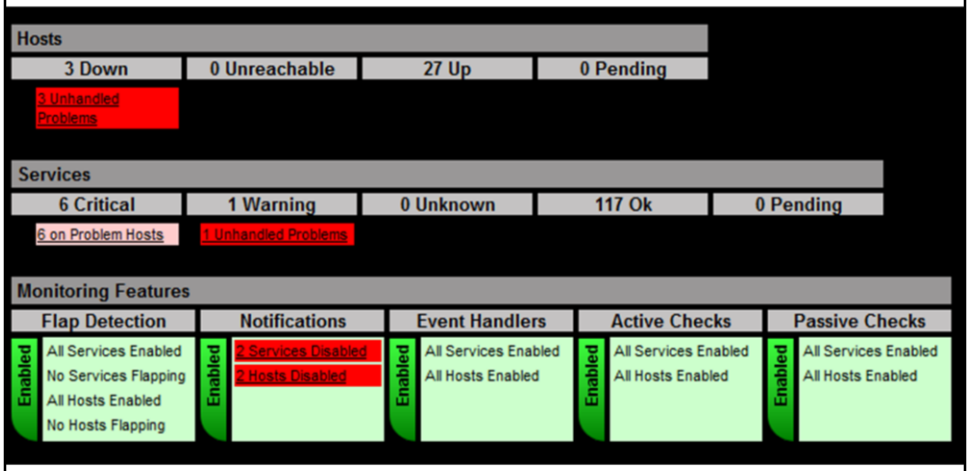
Monitorozó rendszer példa: Nagios

- 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 overview

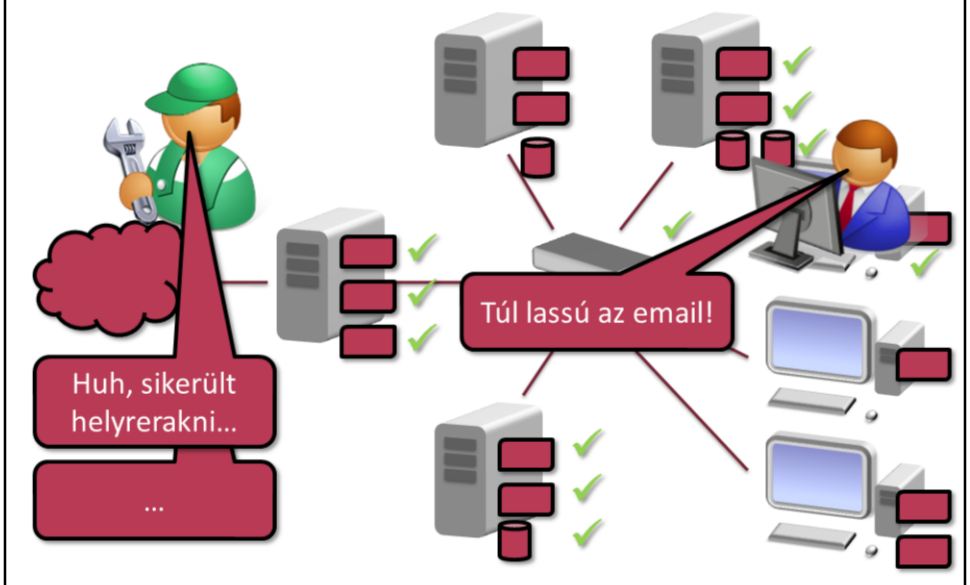


Nagios: service detail

	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



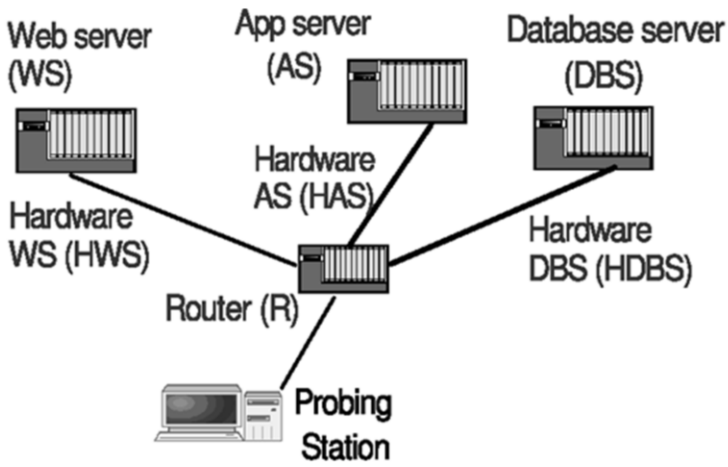
Diagnosztika

- Nem megy a webkiszolgáló. De miért nem?
  - Megfelelő megfigyelések kellenek
- Adott hibahatás okának felderítéséhez mit figyeljünk?
  - Pl. *egy* ESX hoszt több száz valósidejű 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


Web server
(WS)
App server
(AS)
Database server
(DBS)
Hardware
AS (HAS)
Hardware
WS (HWS)
Hardware
DBS (HDBS)
Router (R)
Probing
Station

(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, **pi**
server, **pingDBS** - ping Datab

!!! Implicit tudás:
- topológia-modell
- Szolgáltatás-függőségi modell
- (Egyszerű) hiba(terjedési) modell

Detektálás/lokalizálás

- Minimális hiba*detektáló* 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

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

Összefoglalás: rendszermonitorozás