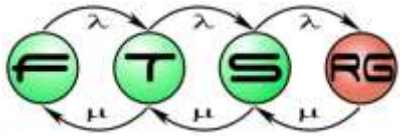
Felhő számítástechnika és Autonóm számítástechnika előadások

A kiberfizikai rendszerek (VIMIMA02) tárgyban

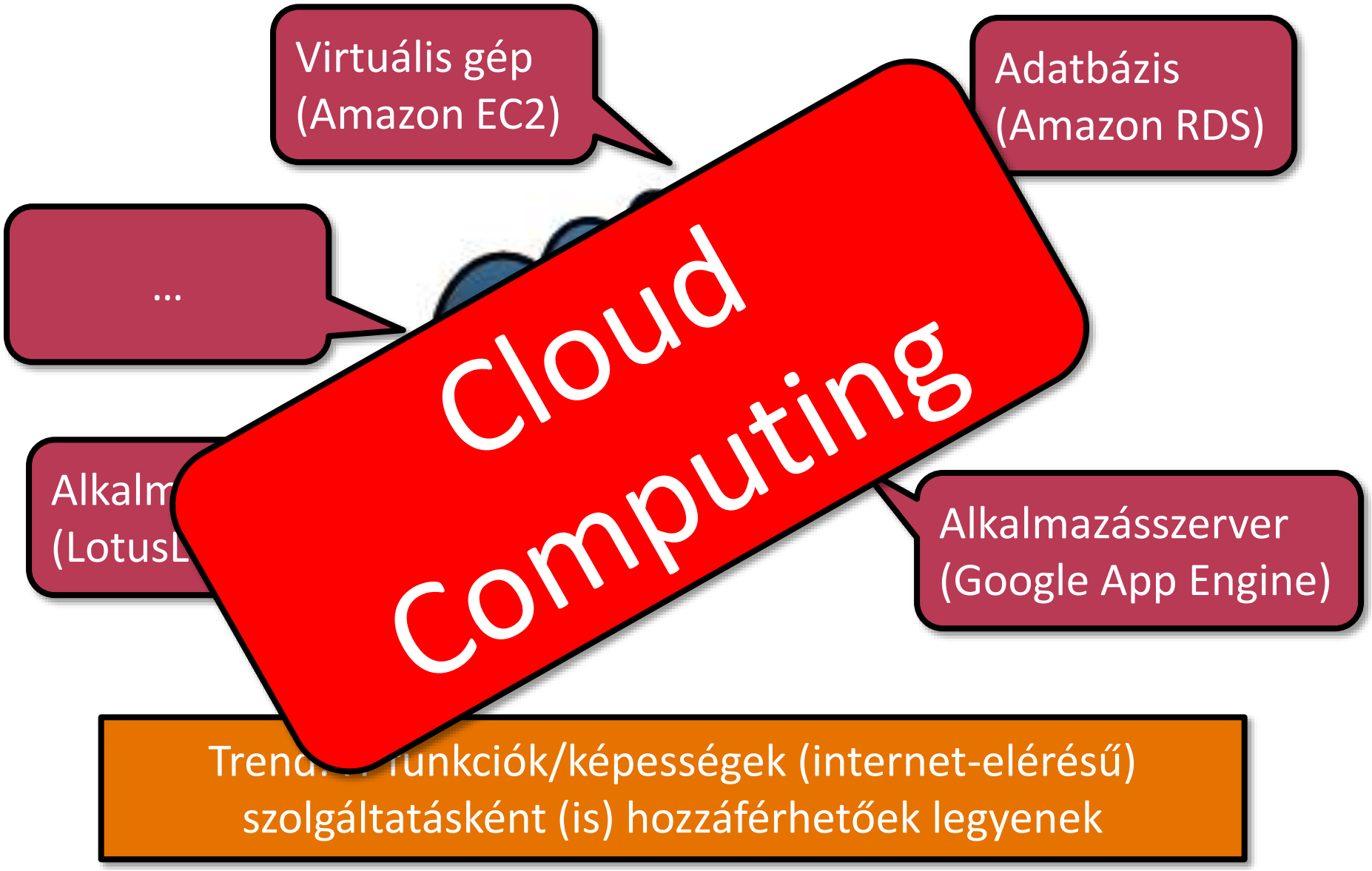
Kocsis Imre, ikocsis@mit.bme.hu

2015. ősz



Felhő számítástechnika

Mi van ma a „felhőben”?



Definíció…?

A „számítási felhők” egy modell, amely lehetővé teszi a hálózaton keresztül való, kényelmes és széles körű hozzáférést konfigurálható számítási erőforrások egy megosztott halmazához.

- NIST 800-145 alapján
- Tulajdonságok, szolgáltatási és telepítési modellek

Eredetiben

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

- NIST 800-145

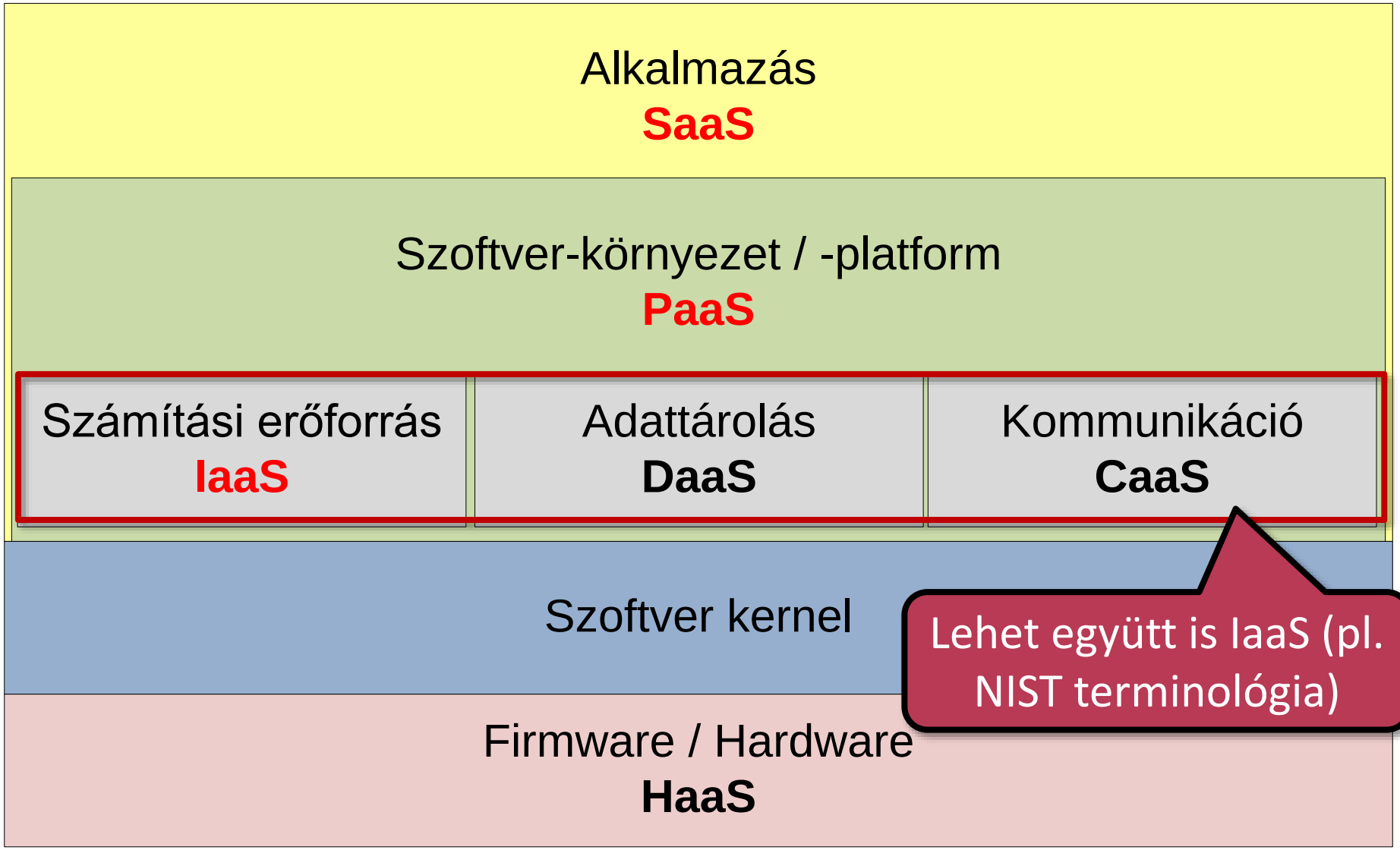
Alapvető tulajdonságok

- Széles körű hálózati hozzáférés
  - Nem csak az Internet
- Igény szerinti önkiszolgálás
- „Resource pooling”
  - „Multi-tenant model”: több bérlő egyszerre
  - Dinamikus ügyfelekhez rendelés
  - Bérlői kontroll: legfeljebb magasabb absztrakciós szinten

Alapvető tulajdonságok

- Rugalmas fel- és leskálázás
  - Látszólag végtelen,
  - akármikor előfizethető erőforrások
- Mért szolgáltatások
  - Szolgáltatás/erőforrás „használata”
  - Sokszor: használat alapú számlázás

Szolgáltatás-terminológia



SaaS

- Képesség: szolgáltató ***alkalmazásainak*** használata
  - Hozzáférés: jellemzően vékony kliens
  - Nem új koncepció
- Példák
  - Google Apps
  - Salesforce CRM
  - LotusLive
  - Microsoft Business Productivity Online Suite (BPOS)
- Néhány igen sikeres terület: kollaboráció, könyvelés, CRM, ERP, HRM, CM, PM, …

PaaS

- Képesség: saját/beszerzett alkalmazás telepítése bérelt ***futtatókörnyezet***be
  - Adott környezeti szolgáltatások
  - Adott használható API-k, nyelvek
  - Konfigurálható környezet
  - Korlátozhatja az alkalmazás-modellt

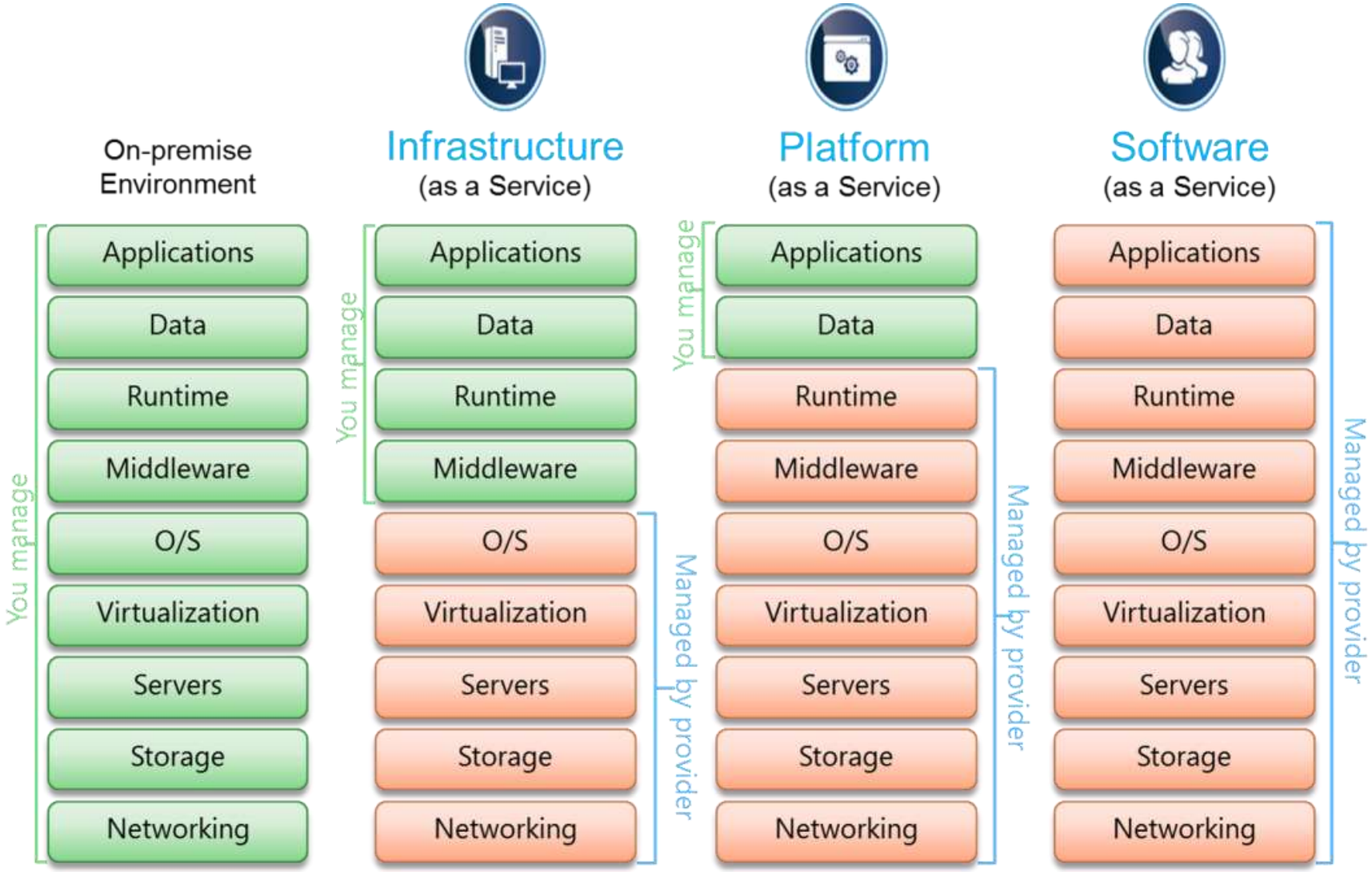
- Google AppEngine
- Microsoft Windows Azure Platform
- Amazon Beanstalk

IaaS

- Képesség: alapvető számítási **erőforrások foglalása**
  - A felhasználó „tetszőleges” szoftvert futtat
  - Jellemzően logikai/virtuális erőforrások
  - Kontroll: OS, tárolás, alkalmazások, hálózati aspektusok *egy része*

- Amazon Elastic Compute Cloud (EC2)
  - Xen alapú virtualizáció
  - Egyre teljesebb ökoszisztéma
  - Az alapszolgáltatás: „tömegtermék”
  - Érdekesség: gépidőre licitálás („bidding”)

Szolgáltatásmodellek összehasonlítása

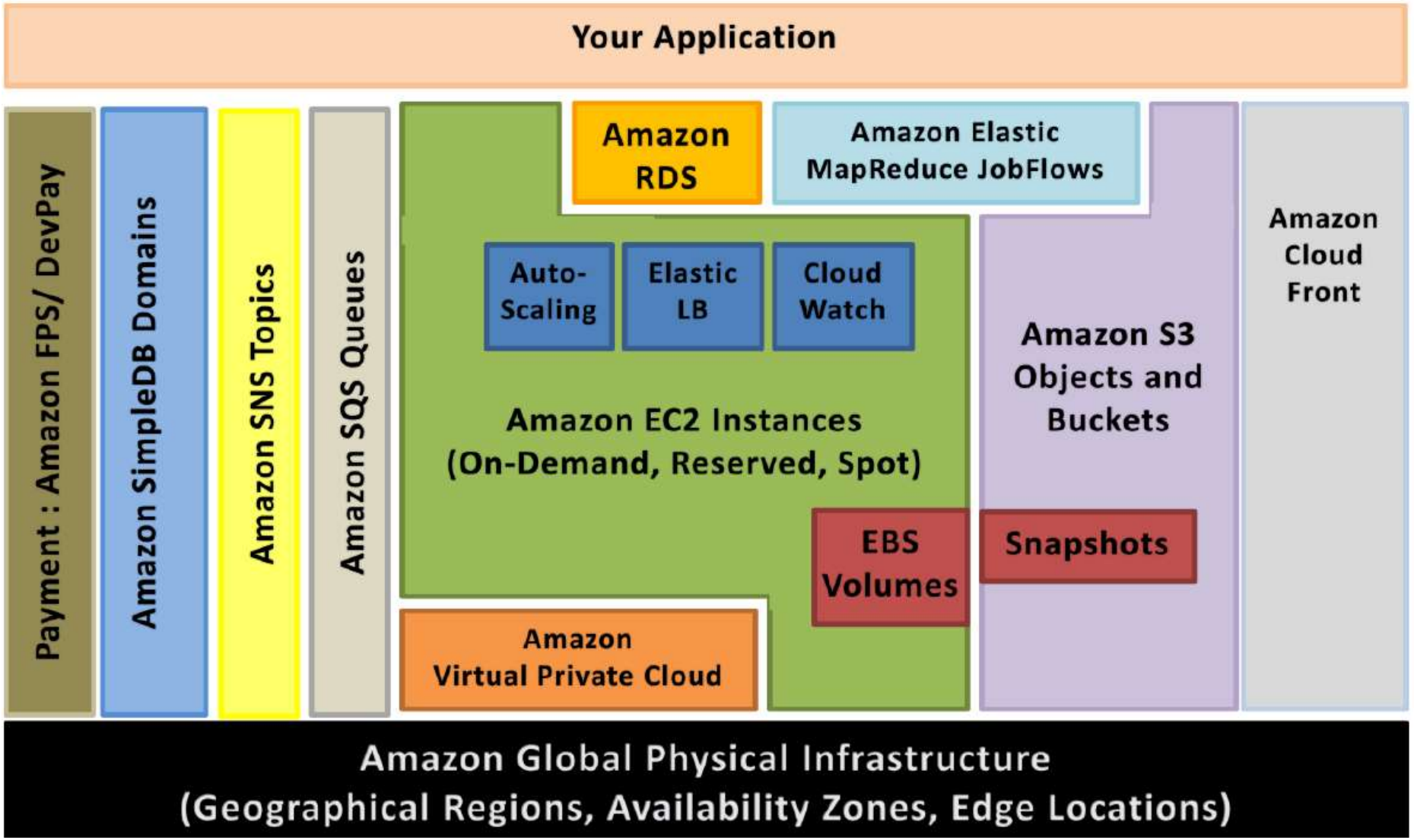


Forrás: http://cloud.dzone.com/articles/introduction-cloud-computing

Amazon EC2

- Infrastructure as a Service
  - EC2: sokáig „A” Cloud Computing (IaaS-re)
- Nem csak csupasz OS lehet
  - DB2, WebSphere, InfoSphere, Lotus Forms, Windows Server 2003/2008, MS SQL, …
- Szoros integráció a többi Amazon Web Service-szel

Amazon Web Services



Forrás: http://www.zdnet.com/blog/

Amazon Web Services

Compute & Networking


Direct Connect
Dedicated Network Connection to AWS


EC2
Virtual Servers in the Cloud


Route 53
Scalable Domain Name System


VPC
Isolated Cloud Resources

Storage & Content Delivery


CloudFront
Global Content Delivery Network


Glacier
Archive Storage in the Cloud


S3
Scalable Storage in the Cloud


Storage Gateway
Integrates On-Premises IT Environments w

Analytics


Data Pipeline
Orchestration for Data-Driven Workflows


Elastic MapReduce
Managed Hadoop Framework


Kinesis
Real-time Processing of Streaming Big Data

App Services


CloudSearch
Managed Search Service


Elastic Transcoder
Easy-to-use Scalable Media Transcoding


SES
Email Sending Service


SNS
Push Notification Service


SQS
Message Queue Service


SWF
Workflow Service for Coordinating Application Compon

Database


DynamoDB
Predictable and Scalable NoSQL Data Store


ElastiCache
In-Memory Cache


RDS
Managed Relational Database Service


Redshift
Managed Petabyte-Scale Data Warehouse Service

Deployment & Management


CloudFormation
Templated AWS Resource Creation


CloudTrail
User Activity and Change Tracking

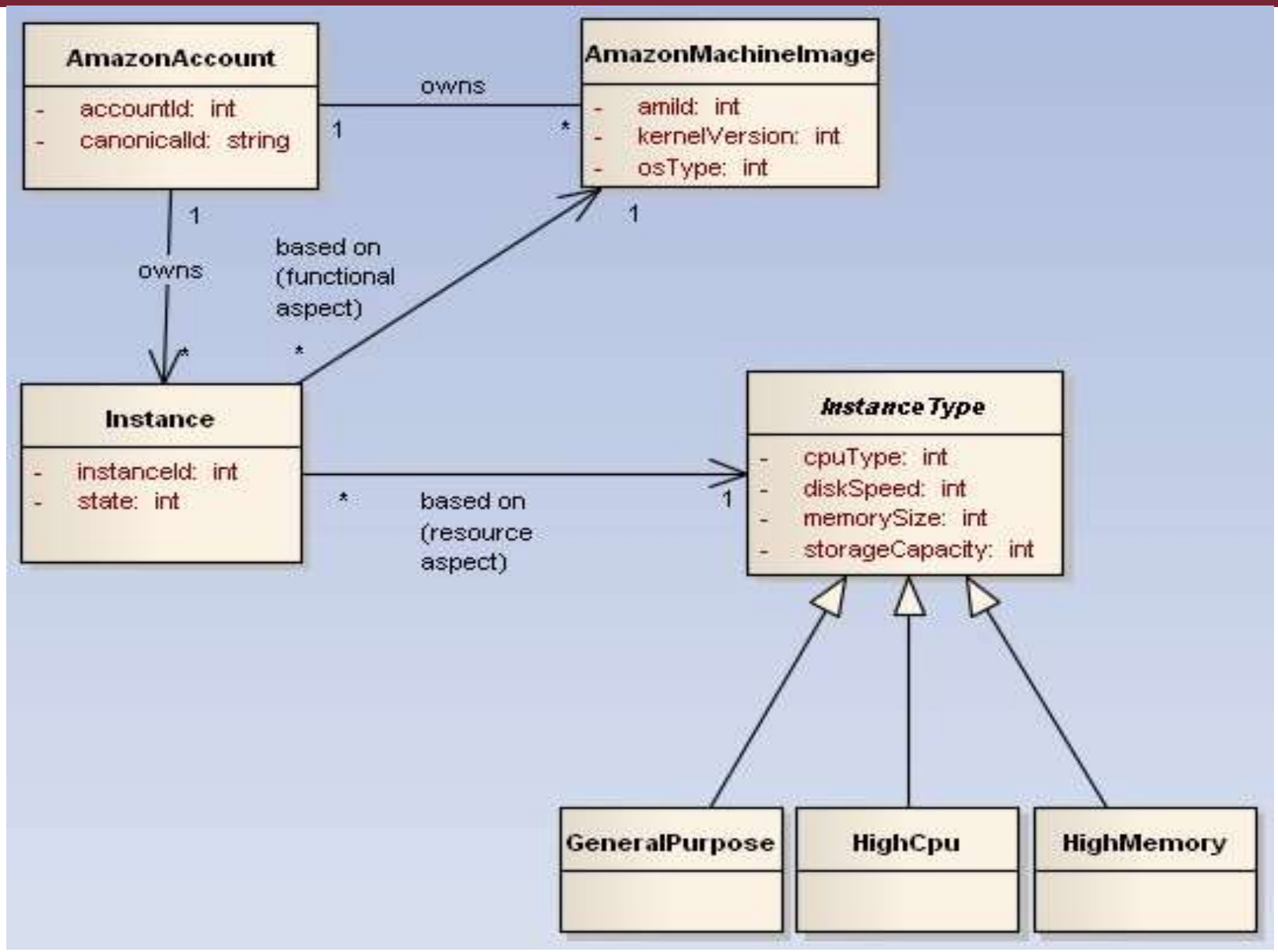

CloudWatch
Resource and Application Monitoring


Elastic Beanstalk
AWS Application Container


IAM
Secure AWS Access Control


OpsWorks
DevOps Application Management Service

Amazon EC2 - alapfogalmak



DEMO Amazon EC2

- Alapvető műveletek
- Példányok létrehozása

- Nagy(ob)ban építkezés: pl. AWS CloudFormation
  - Példa: https://s3-us-west-2.amazonaws.com/cloudformation-templates-us-west-2/WordPress_Multi_AZ.template (jsonviewer.stack.hu)

Az infrastruktúra a szolgáltatás mögött

North America

US East (Northern Virginia) Region
EC2 Availability Zones: 5*
Launched 2006

US West (Oregon) Region
EC2 Availability Zones: 3
Launched 2011

US West (Northern California) Region
EC2 Availability Zones: 3*
Launched 2009

AWS GovCloud (US) Region
EC2 Availability Zones: 2
Launched 2011

Asia Pacific

Asia Pacific (Singapore) Region
EC2 Availability Zones: 2
Launched 2010

Asia Pacific (Sydney) Region
EC2 Availability Zones: 2
Launched 2012

Asia Pacific (Tokyo) Region
EC2 Availability Zones: 3
Launched 2011

China (Beijing) Region
EC2 Availability Zones: 1 Coming Soon
Learn more at www.amazonaws.cn

Az infrastruktúra a szolgáltatás mögött

Europe / Middle East / Africa

EU (Ireland) Region
EC2 Availability Zones: 3
Launched 2007

EU (Frankfurt) Region
EC2 Availability Zones: 2
Launched 2014

AWS Edge Locations: Amsterdam, The Netherlands (2), Dublin, Ireland, Frankfurt, Germany (3), London, England (3), Madrid, Spain, Marseille, France, Milan, Italy, Paris, France (2), Stockholm, Sweden, and Warsaw, Poland

South America

São Paulo Region
EC2 Availability Zones: 2
Launched 2011

AWS Edge Locations: Rio de Janeiro, Brazil, São Paulo, Brazil

Az infrastruktúra a szolgáltatás mögött

- Titkos, de valószínűsíthetően óriási méretek
- Min. 1,4 millió, de inkább 2,8-5,6 kiszolgáló*
- Konténerizáció, saját link adatközpontok között, integrált saját CDN, …
- Más szolgáltatók: kisebbek
  - De gondoljunk bele, hogy -1 nagyságrend mi…

*Forrás: http://www.enterprisetech.com/2014/11/14/rare-peek-massive-scale-aws/

Gartner IaaS MQ 2014

Jelentősége

- Ha nem építitek/üzemeltetitek, akkor használni fogjátok
  - Adatbiztonsági problémák ellenére rég nem „shadow IT”
  - Privát és hibrid cloudok!
  - Üzleti megfontolások: következő cloud ea.

- A (virtualizált) infrastruktúrát érteni kell
  - Skálázás-hibatűrés-(~)biztonság az alkalmazásban
  - dinamikus huzalozás

- XaaS: MONaaS/MaaS, ***NFVIaaS***, UCaaS, NaaS, D(esktop)aaS, …
- IoT: „the swarm on the edge of the cloud”

Alapvető problémák

- Mikor ~~kell~~ / érdemes / nem érdemes / tilos felhőt alkalmazni?
- Hogyan teszünk különbséget?
- Teljesítmény?

Néhány jellemző használati eset

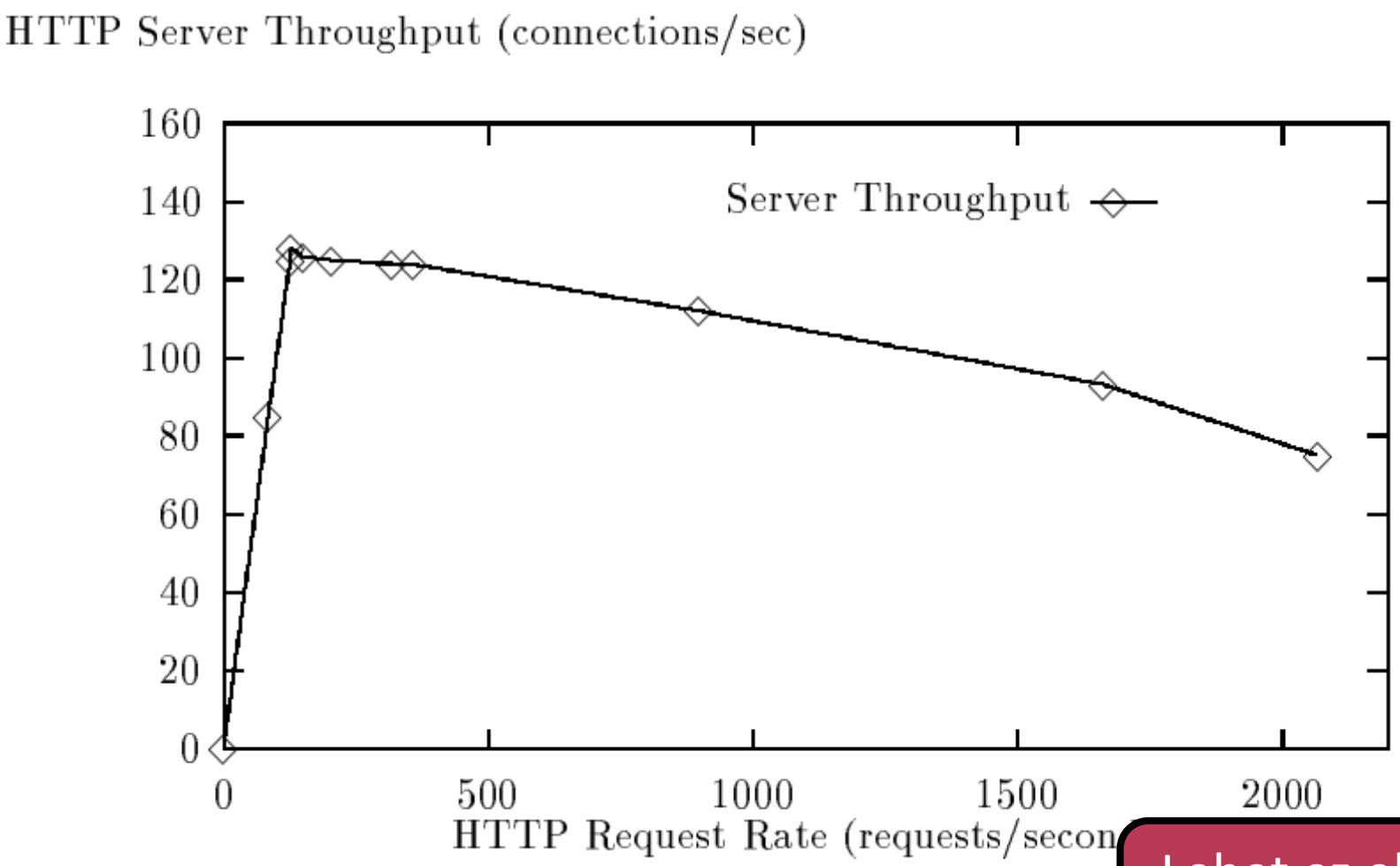
- Kiegészítő képességek és kompetenciák
- „Core and context”
- „Cash Flow”
- (Dinamikus igények miatt nem megfelelő) kapacitás
- Üzletmenet-folytonosság és katasztr. helyreállás
- „Sebesség”

Jellemzően mikor *ne*?

- Konstans terhelés
- Örökölt (*legacy*) rendszerek
- Adatbiztonság, törvényi és szabályozói megfelelés

Igények és kapacitás (*demand and capacity*)

Véges kapacitású erőforrások



Lehet ez ellen védekezni?

Ábra forrása: [2], p 10

Erőforrások skálázása

- „Scale up”

- „Scale out”
  - Kiszolgálás párhuzamosíthatósága?
  - „webscale” technológiák
  - → Fürtözés és replikáció
  - (Mongo DB Is Web Scale)

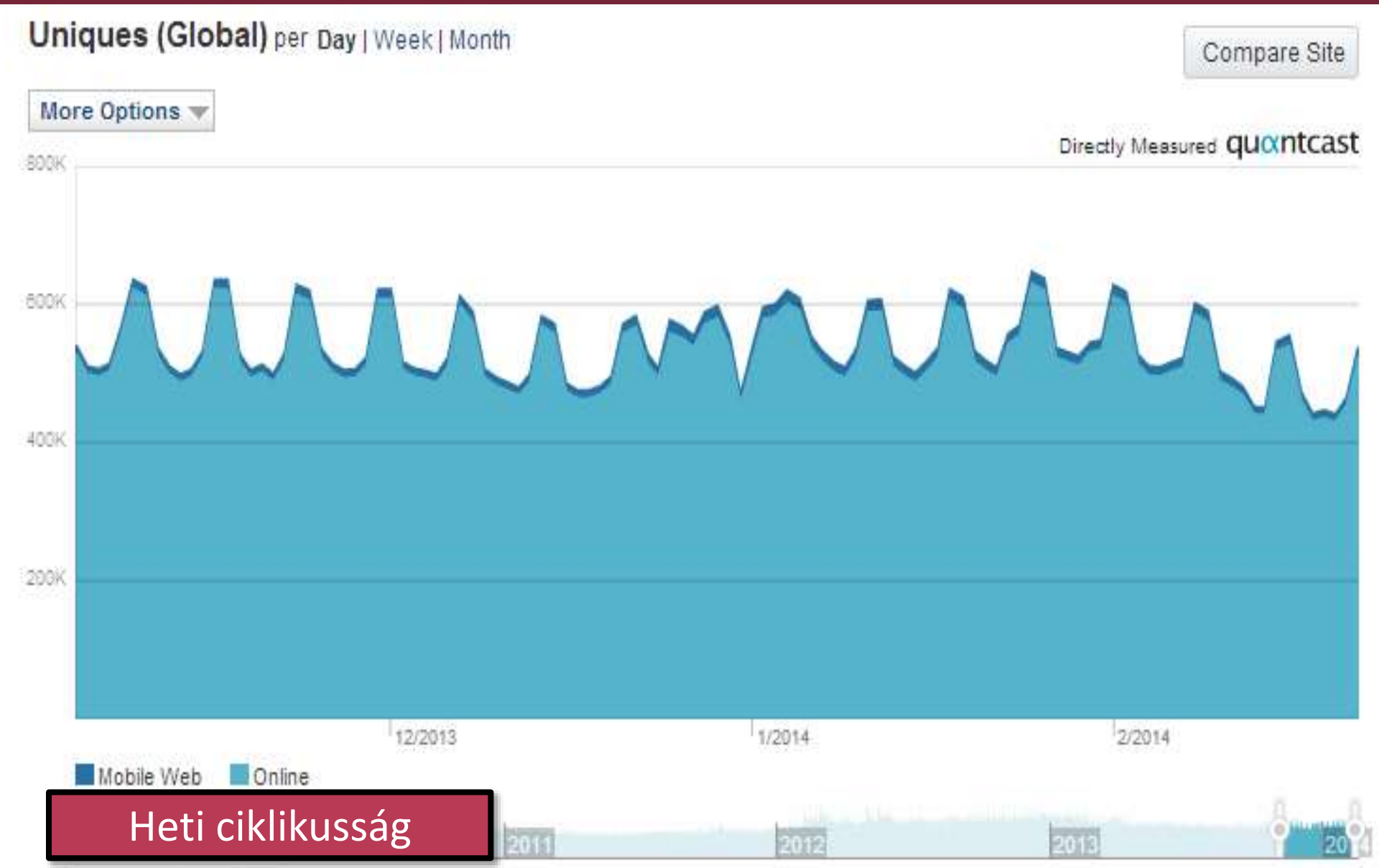
Kapacitástervezés

- Túl alacsony kapacitás: nem kiszolgált igény
  - Pl. átlagos terhelésre tervezés
- Túl nagy kapacitás: pazarlás(?)
  - Pl. csúcsterhelésre tervezés

- Helyi („*in-house*”) fizikai kapacitás ált. lassan és drágán növelhető…
  - … és csökkenthető!
- (Egyszeri) befektetés és *elköteleződés*

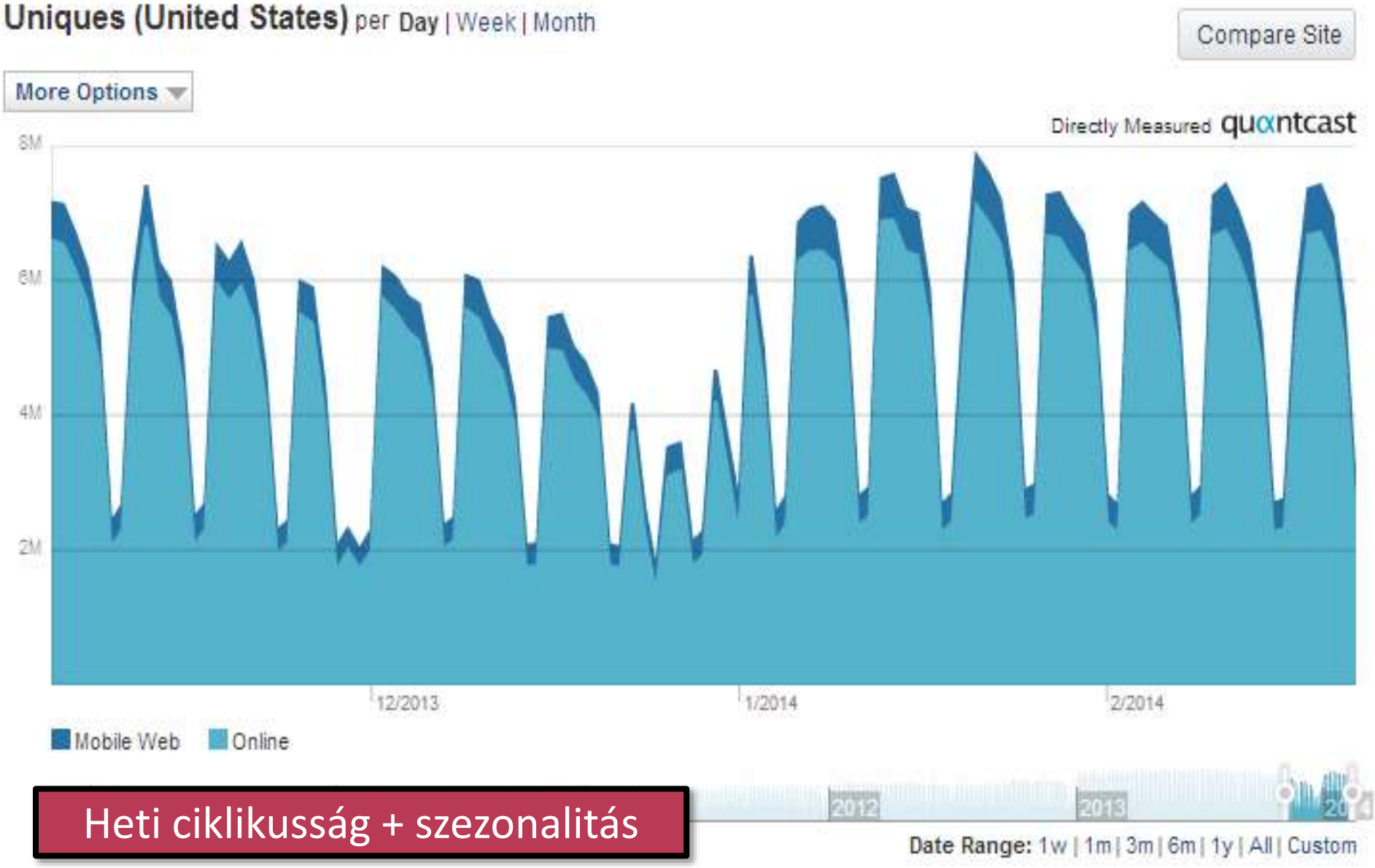
- Igények?

www.quantcast.com/utorrent.com



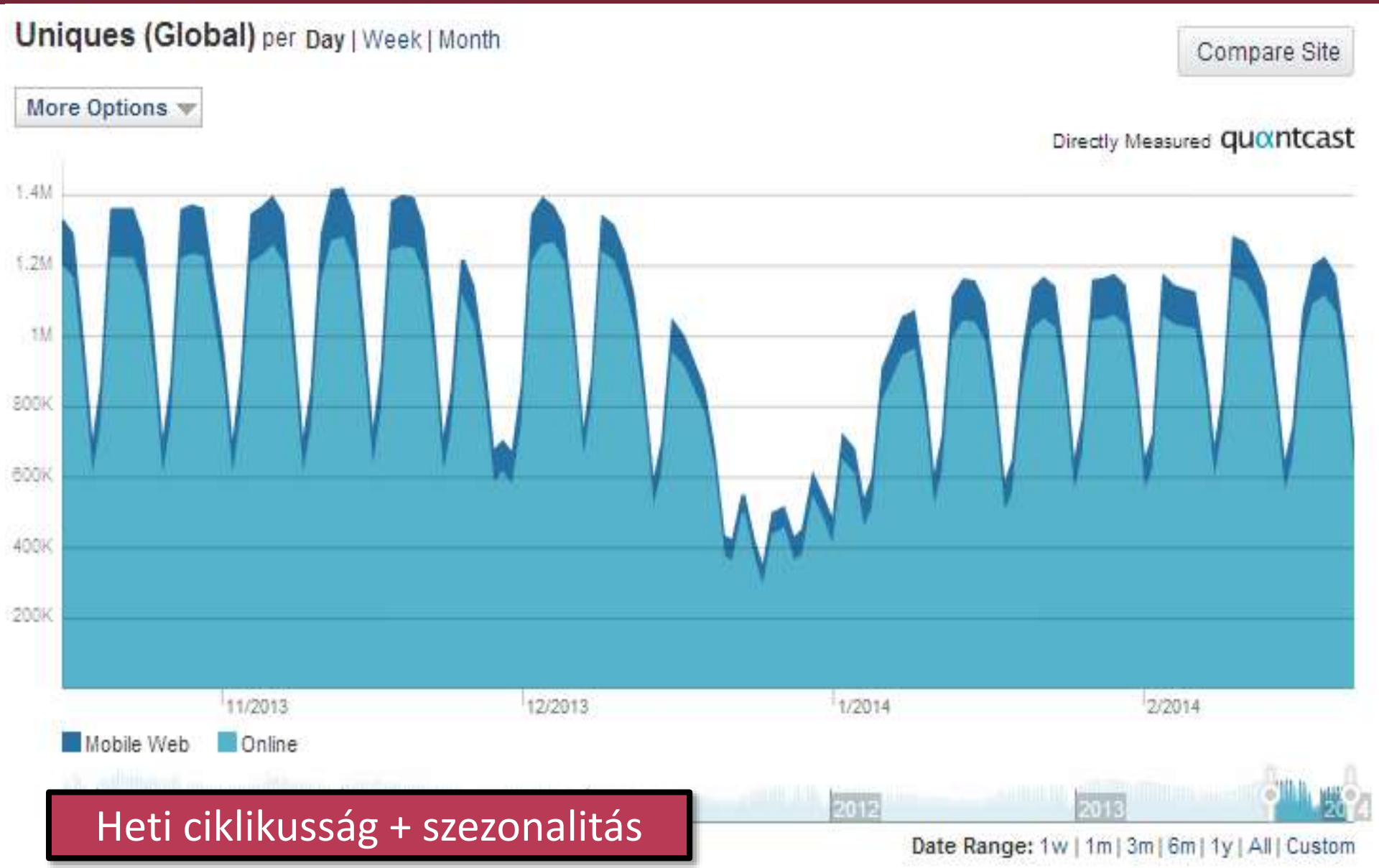
Heti ciklikusság

www.quantcast.com/linkedin.com



Heti ciklikusság + szezonalitás

www.quantcast.com/merriam-webster.com



Heti ciklikusság + szezonalitás

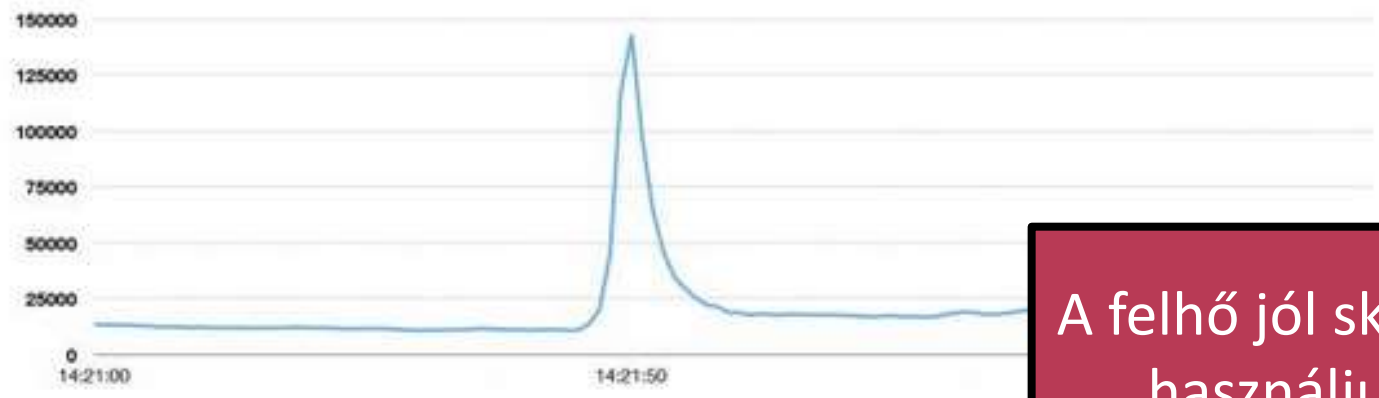
www.quantcast.com/linkedin.com

Heti ciklikusság + szezonalitás + trend + …

…előrejelezhetetlen változások

Recently, something remarkable happened on Twitter: On Saturday, August 3 in Japan, people watched an airing of Castle in the Sky, and at one moment they took to Twitter so much that we hit a one-second peak of 143,199 Tweets per second. (August 2 at 7:21:50 PDT; August 3 at 11:21:50 JST)

To give you some context of how that compares to typical numbers, we normally take in more than 500 million Tweets a day which means about 5,700 Tweets a second, on average. This particular spike was around 25 times greater than our steady state.



A felhő jól skálázható; használjuk azt!

Ára?

Linux | RHEL | SLES | Windows | Windows

Region: EU (Frankfurt)

	vCPU	ECU	Memory (G		nux/UNIX Usage
General Purpose - Current Generation					
t2.micro	1	Variable	1		$0.015 per Hour
t2.small	1	Variable	2		$0.030 per Hour
t2.medium	2	Variable	4	EBS Only	$0.060 per Hour
m3.medium	1	3	3.75	1 x 4 SSD	$0.083 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	$0.166 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	$0.332 per Hour
				2 x 80 SSD	$0.665 per Hour

~22 HUF / óra
~528 HUF / nap
~16 kHUF / hónap
~193 kHUF / év

+ egyéb költségek (EBS, S3, kimenő forgalom,…)
- „Reserved instance”

http://aws.amazon.com/economics/

Gartner, 2013 [4]

*„For **larger businesses** with **existing** internal **data centers**, well-managed virtualized infrastructure and efficient IT operations teams, IaaS for **steady-state workloads** is often no less expensive, and **may be more expensive**, than an internal private cloud.”*

Cloud „telepítési” (*deployment*) modellek [5]

- Privát: egy szervezet számára, több fogyasztó (pl. üzleti egységek). *Lehet* saját tulajdonú és helyben üzemeltetett.
- Közösségi (*community*): egy szervezeteken átívelő, igényeken osztozó közösség kizárólagos használatára.
  - Kormányzat, egészségügy, pénzügy, oktatás, …
- Publikus: nyílt (persze nem szabad) hozzáférésű. Fizikailag a szolgáltatónál.
- Hibrid: kettő vagy több különálló felhő kompozíciója adat- és alkalmazás-hordozhatósággal.

Költségoptimalizálás?

- Ára van…
  - A ki nem szolgált igényeknek
  - És a felesleges kapacitásnak

- A felhő „egységára” (*unit price*) *lehet* magasabb a saját befektetésnél, de…
- ... ugyanez igaz az autóbérlésre és a hotelszobákra

- ***Hibrid felhők:*** privát: alapterhelés, publikus: változó
  - Komolyabb optimalizációs probléma és még mindig terhelést kell becsülni…

Szolgáltatói oldalon…

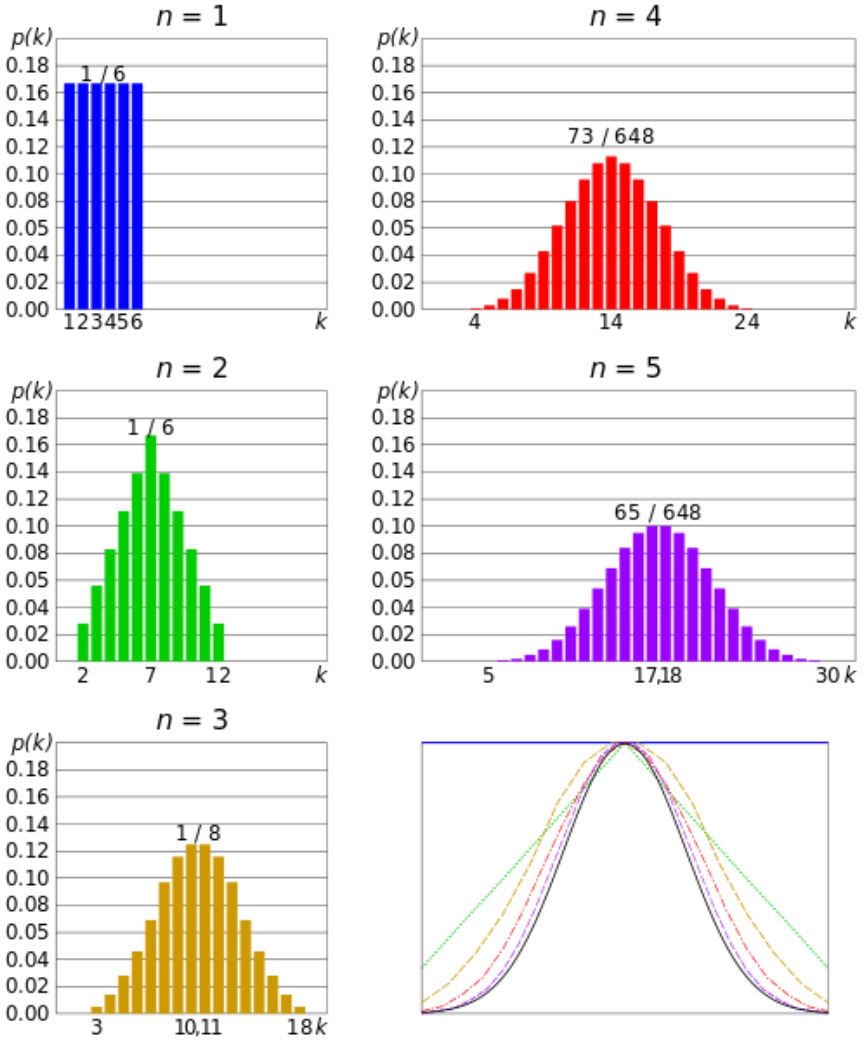
Miért éri meg a szolgáltatónak? [6]

- (Centrális határeloszlás-tétel nélkül)
- X_i azonos várhatóértékű (μ) és varianciájú (σ^2), független val. változók
- Variációs koefficiens (*coefficient of variation*): $\frac{\sigma}{\mu}$
- Összeg várhatóértéke: várh. összege
- Összeg varianciája: varianciák összege

$$\mathrm{CV}(X_{sum}) = \frac{\sqrt{n\sigma^2}}{n\mu} = \frac{1}{\sqrt{n}}\frac{\sigma}{\mu} = \frac{1}{\sqrt{n}}CV(X_i)$$

A „statisztikai multiplexálás” hatása

- **Az átlaghoz képest vett szórás csökken**
- $\frac{1}{\sqrt{n}}$: gyorsan; kisebb privát cloud-ok!
- A valóságban persze nem független minden terhelés



Ábra forrása: [7]

„Ingyen idő”/ „ingyen gyorsítás”

Párhuzamosítható terhelések

- Egyre több „zavarbaejtően” párhuzamos (*embarrassingly parallel*), „scale-out” alkalmazáskategóriánk van

- NYT TimesMachine [12]: public domain archív
  - Konverzió web-barát formátumra [13]: Hadoop, pár száz VM, 36 óra

- **A használat alapú számlázás miatt ~ugyanannyiba kerül, mint egy VM-mel**

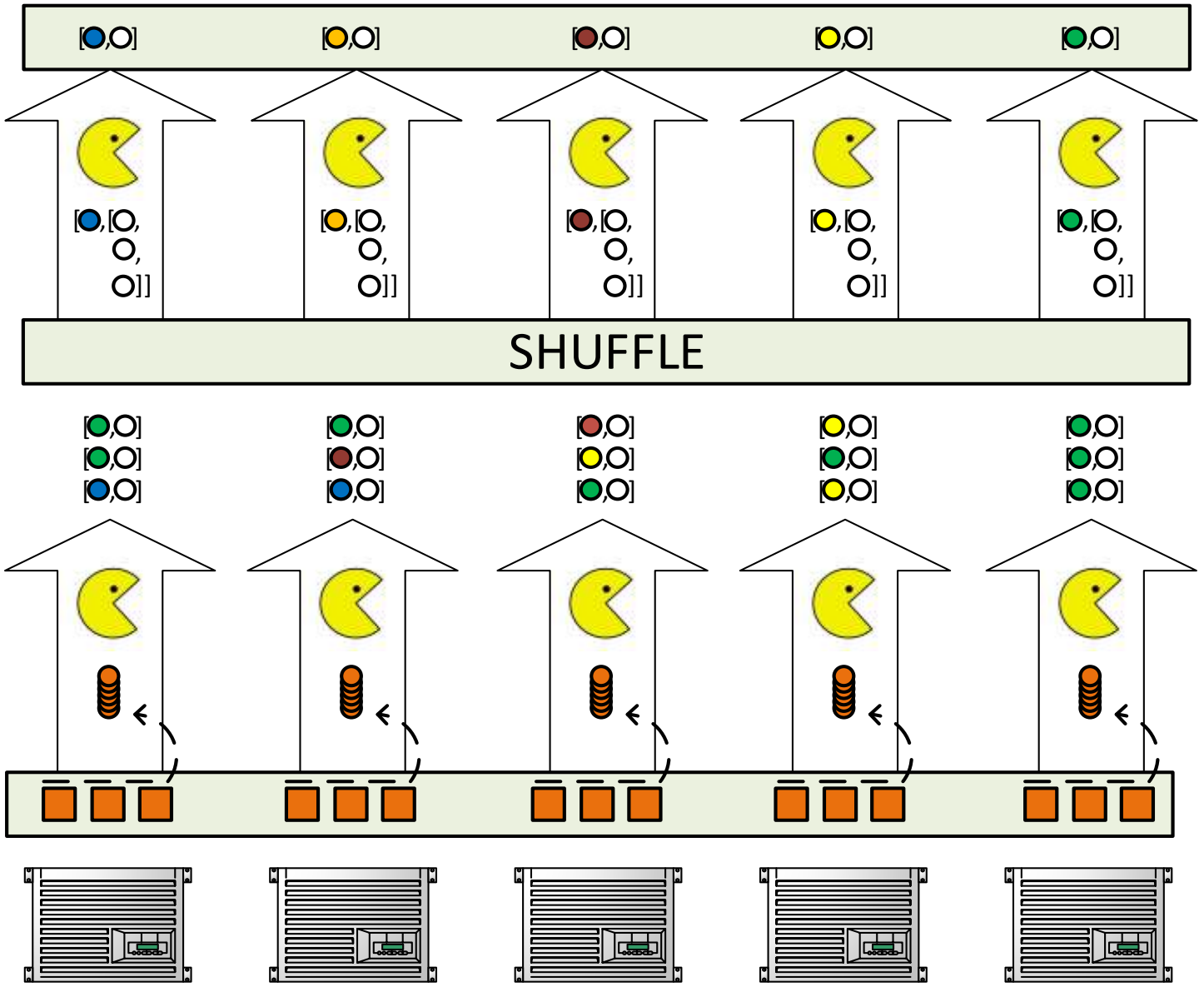
- Praktikusan: „ingyen idő”

MapReduce (Hadoop)

„Reduce”

„Map”

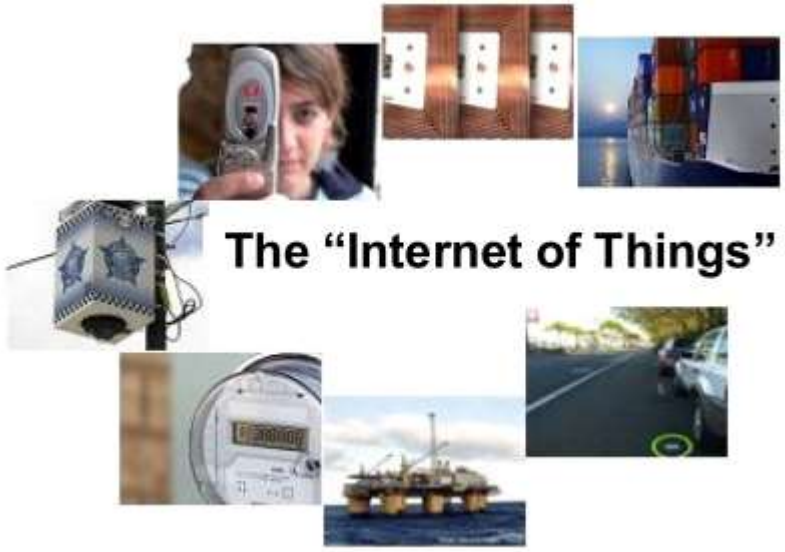
Distributed File System



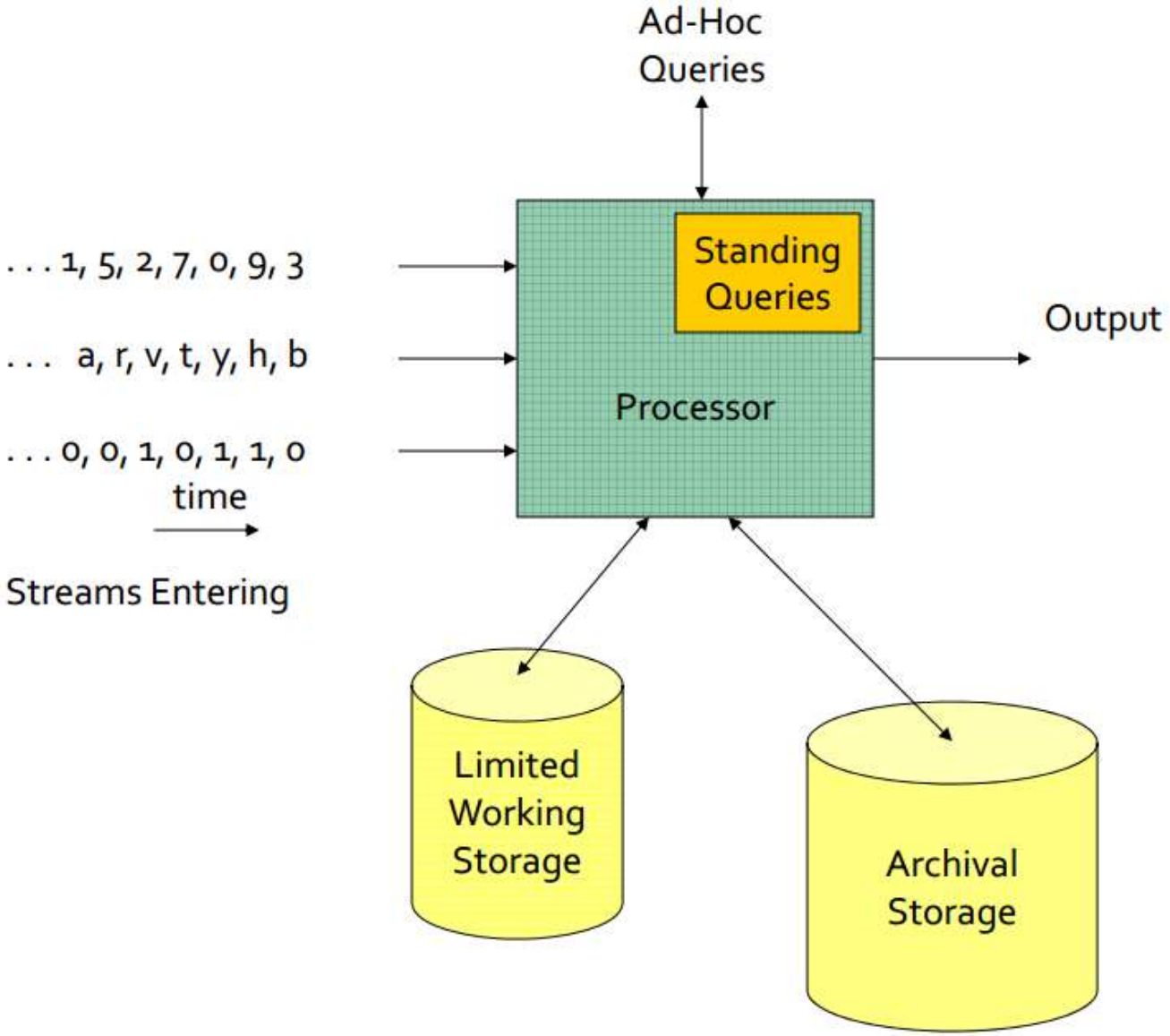
Stream processing

Adatfolyam-források

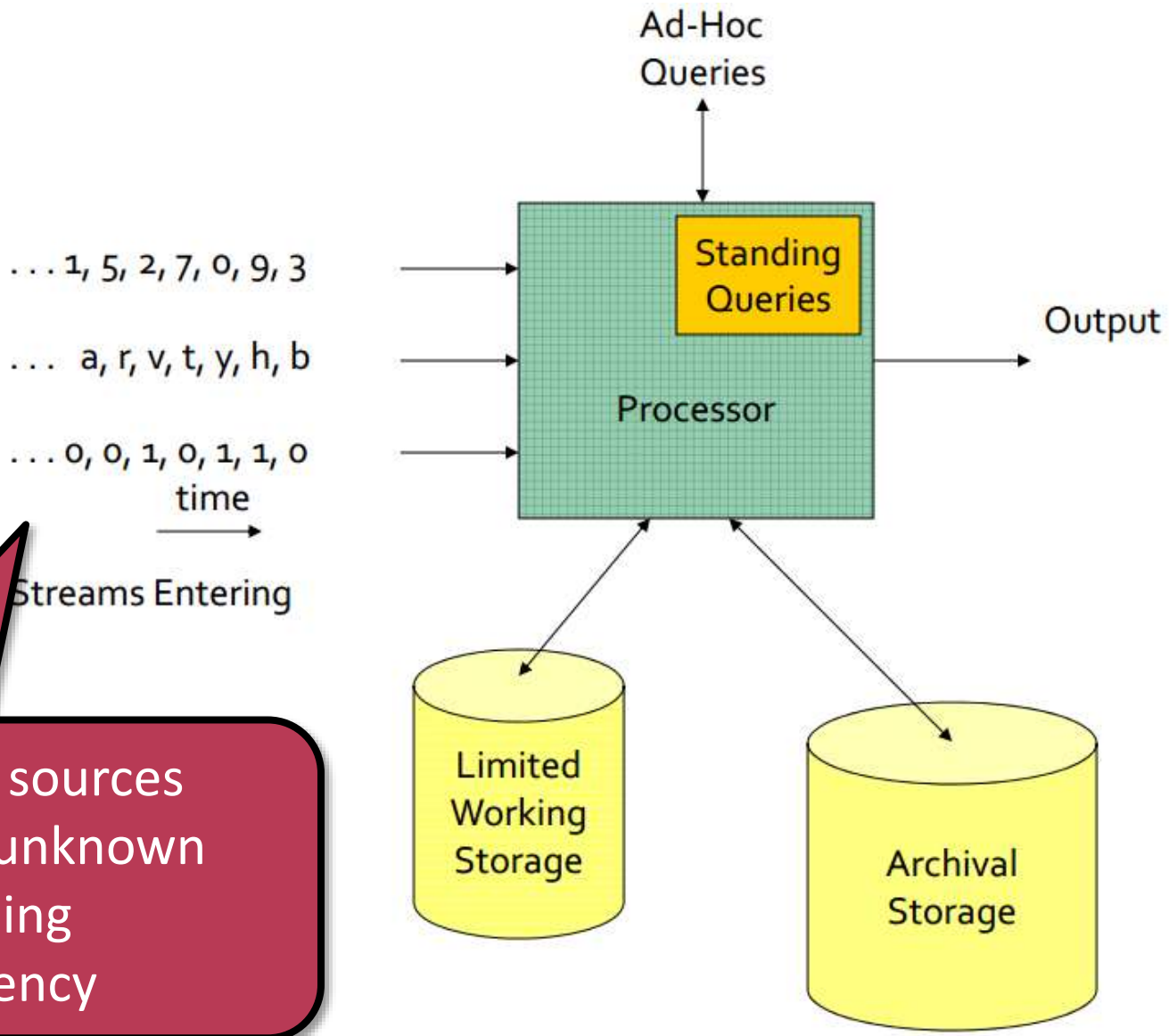
- Szenzor-adatok
  - 1 millió szenzor x 10/s x 4B
- Képek
  - Szatelitek: n TB/nap
- Internetes szolgáltatások
- Hálózati forgalom
- Tőzsdei adatok
- …



Stream processing (vs „at rest” Big Data)

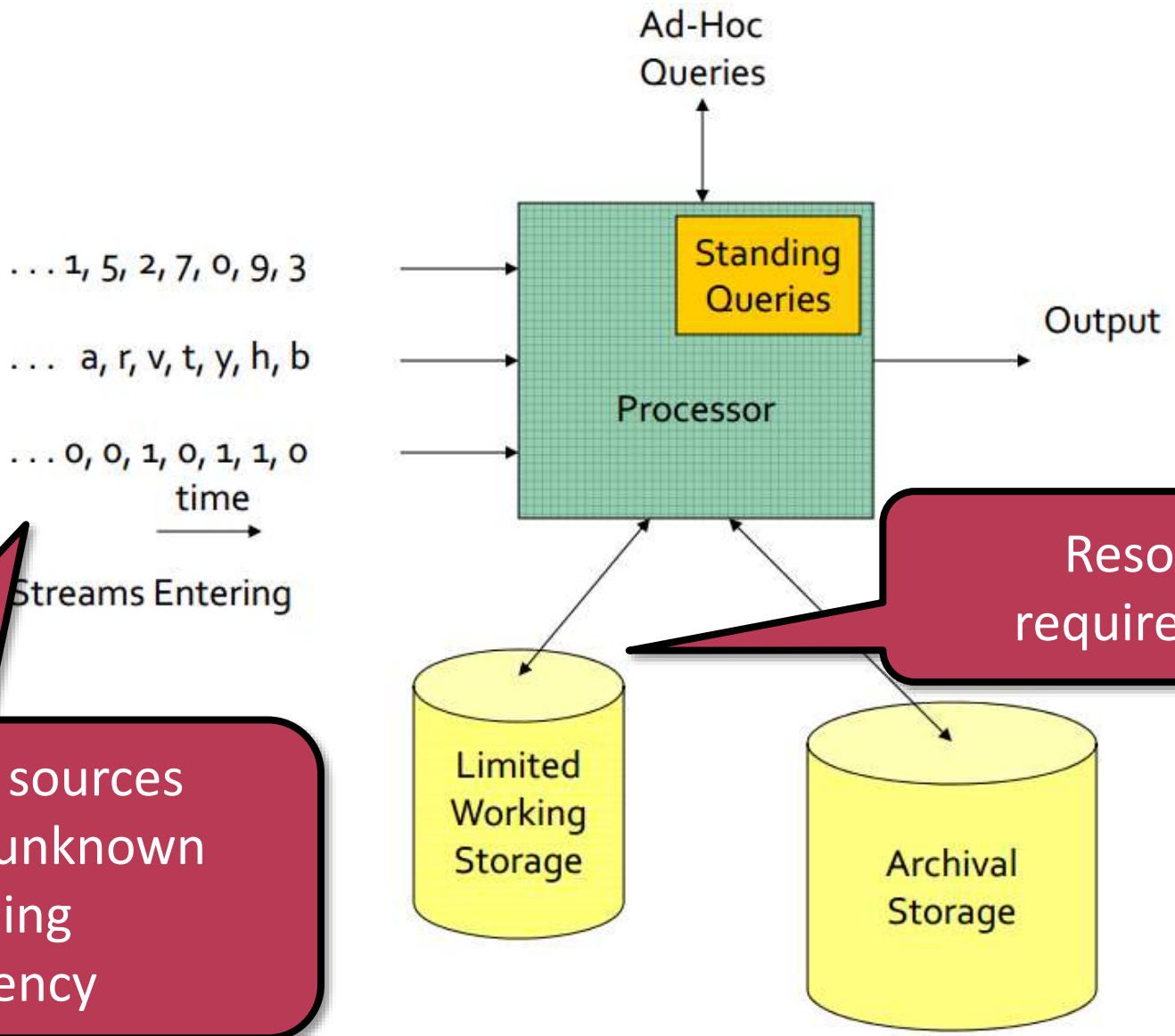


Stream processing



1. Many sources
2. With unknown sampling frequency

Stream processing

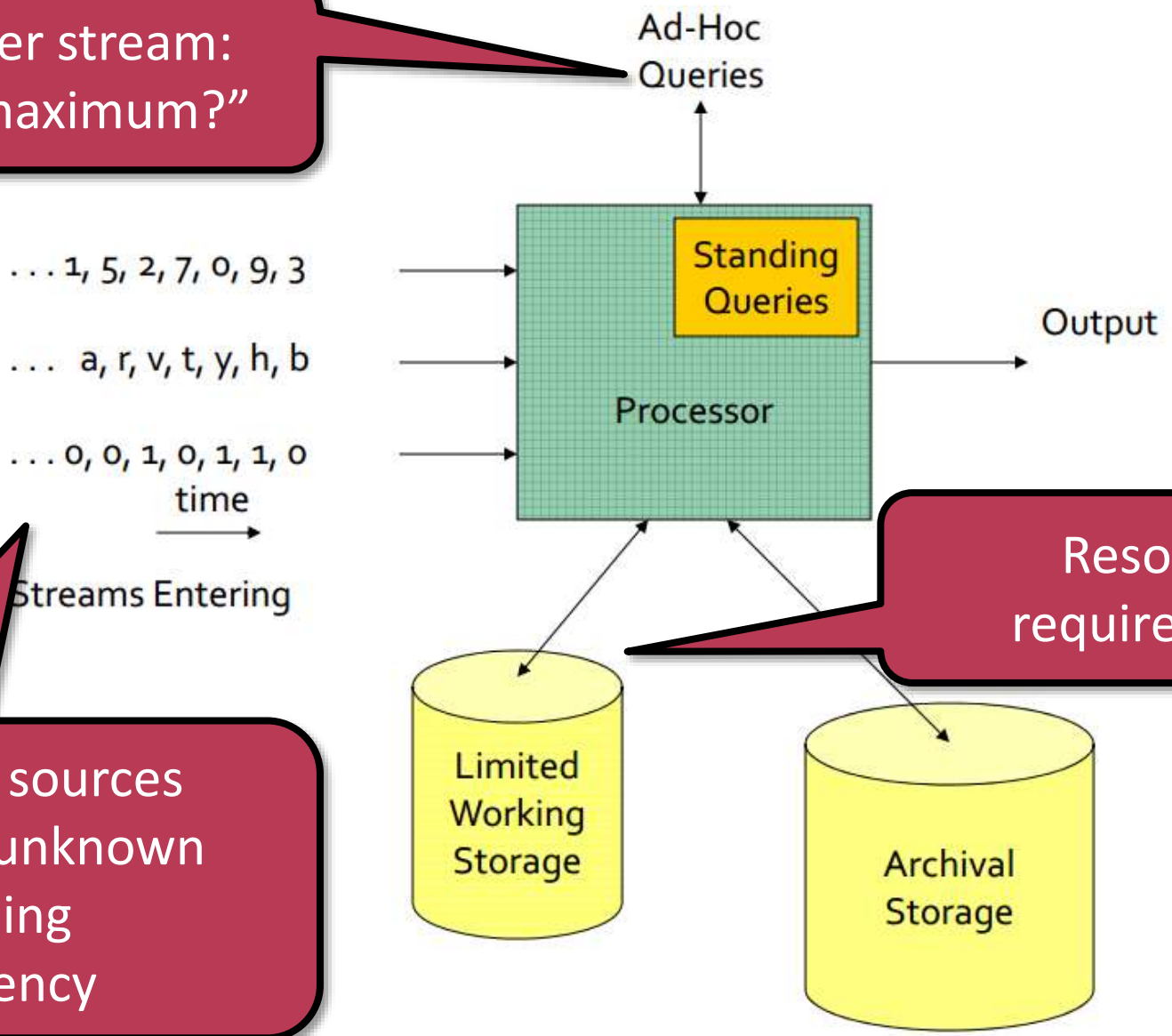


Resource requirements

1. Many sources
2. With unknown sampling frequency

Stream processing

Once per stream:
„Local maximum?”



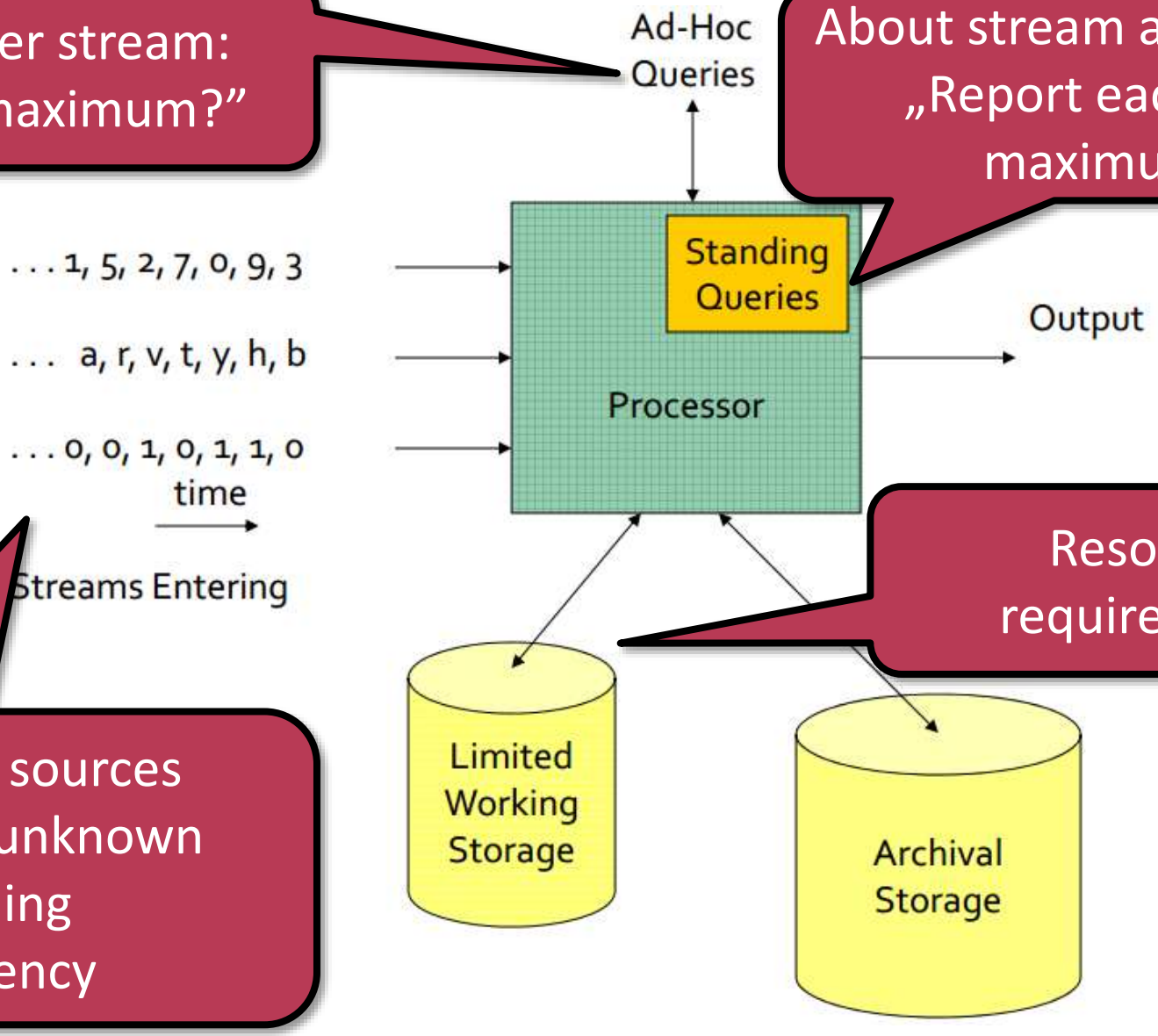
Resource requirements

1. Many sources
2. With unknown sampling frequency

Stream processing

Once per stream: „Local maximum?”

About stream at all times: „Report each new maximum”

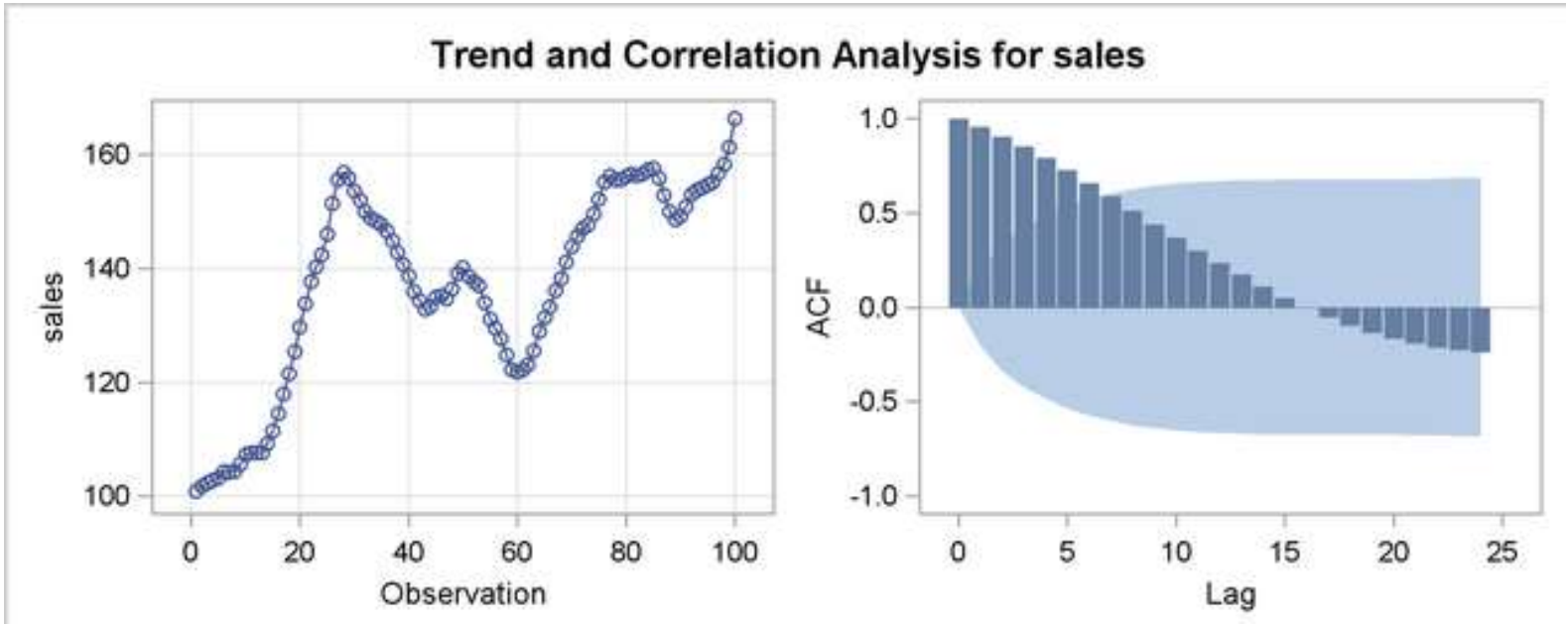


Resource requirements

1. Many sources
2. With unknown sampling frequency

Typically sliding window approches

- Autocorrelation methods
  - Where do we differ from the predicted value?
  - Where does the autocorrelation model change?

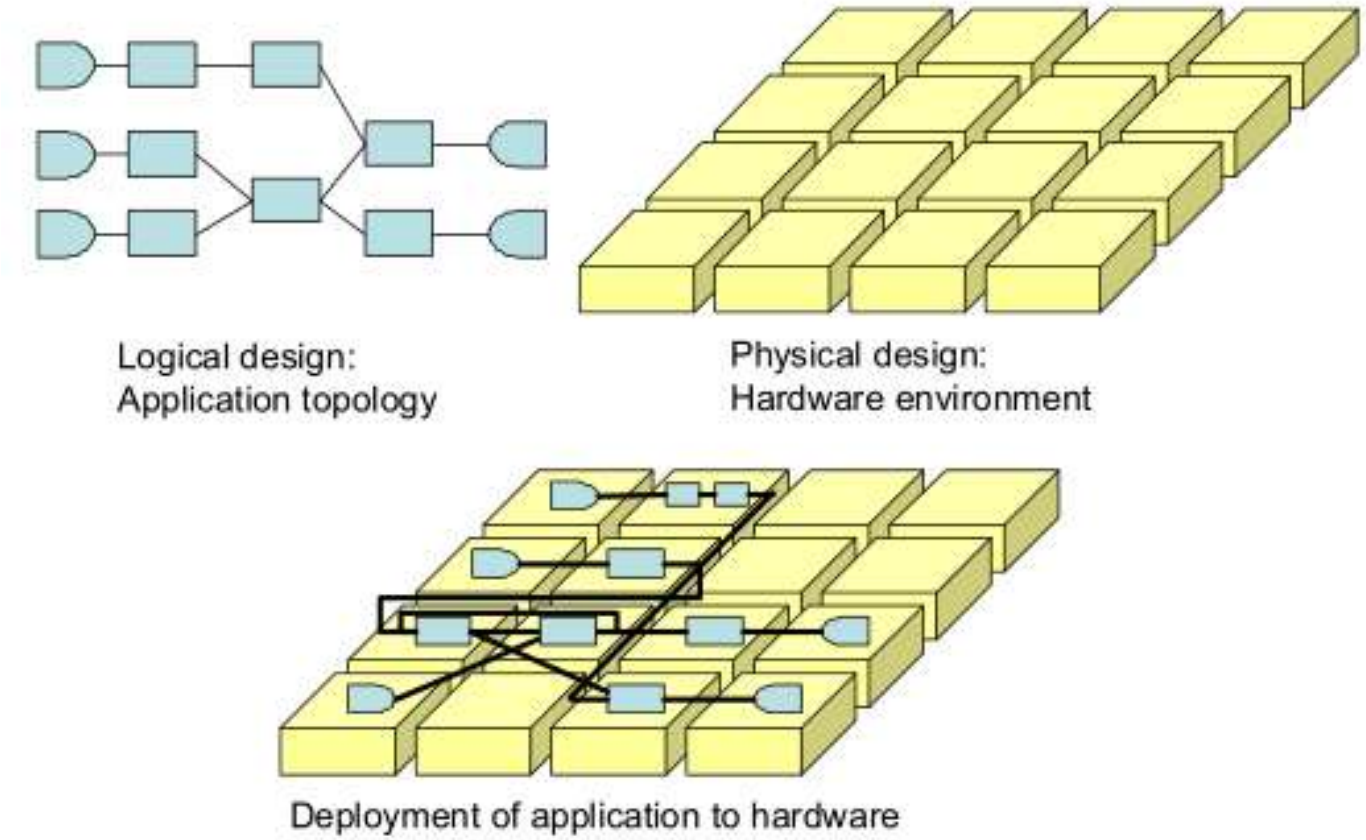


Feldolgozás: időkorlát!

- Diszk nem használható
- Megengedett memóriaigény: korlátos
- Elemenkénti számítási igény: korlátos

- Szokásos megoldások:
  - n-esenkénti (*tuple*) feldolgozási logika
  - Csúszóablakos tárolás és feldolgozás
  - Mintavételezés
  - Közelítő algoritmusok
  - WCET-menedzsment: skálázási logikán keresztül
    - Illetve lehet heurisztika/mintavétel-hangolás is, de az nehéz

IBM InfoSphere Streams



Forrás: [15], p 76

Basic cloud market forces – examples by simulation

DEMO complexmodels.com

- Value of utility resources in the cloud
- Value of resource pooling and load sharing across a grid
- Value of dispersion in latency reduction
- Value of aggregation in variability smoothing and peak reduction
- +1: CV reduction effect for uniform and binomial distribution

IaaS teljesítmény

IaaS teljesítmény

Ismeretlen / nem vezérelhető ütemezés (sched.)

„Noisy neighbors” (interferenciák)

+ menedzsment-teljesítmény?

Ismeretlen / nem vezérelhető terítés (depl.)

HW: lehet nem megismerhető, heterogén

Steal time

- Linux vendég (*guest*)
  - /proc/stat cpu
  - 2.6.11+ (+ kell hipervizor támogatás)

```
top - 10:38:23 up 37 days,  7:43,  2 users,  load average: 1.15, 0.88, 0.86
Tasks: 147 total,   3 running, 144 sleeping,   0 stopped,   0 zombie
Cpu(s):  40.0%us,  0.0%sy,  0.0%ni, 0.0%id,  0.0%wa,  0.0%hi,  0.1%si, 60.0%st
Mem:   4332028k total,  3614752k used,   717276k free,    47692k buffers
Swap:  2097148k total,    20480k used,  2076668k free,  1000060k cached
```

- ESXi: „CPU ready” metrika
  - Mérés: hipervizorban, nem VM-ben

IaaS teljesítmény

- Telepítési döntések
  - Használjam-e ezt a felhőt?
- Kapacitástervezés
  - Erőforrások típusa, mennyisége
- Telj. előrejelzés
  - Várható QoS
  - … és variabilitása

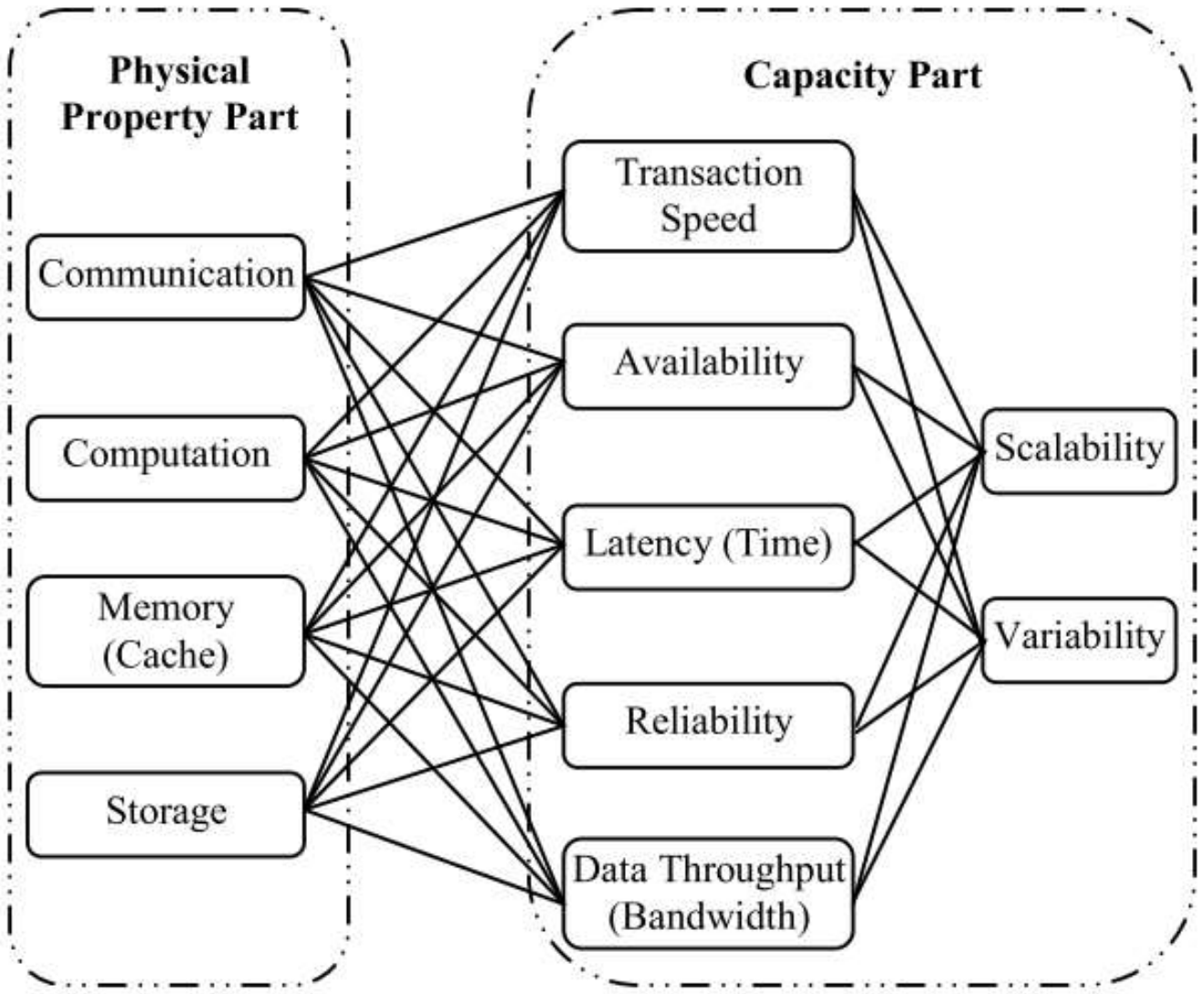
Benchmarkolás!

Benchmarkolás (pragmatikusan)

- (De-facto) **standard alkalmazások**
- jól definiált **metrikákkal**, mint kimenetekkel
- melyek adott **alrendszerekre** fókuszálhatnak
- IT rendszerek **összehasonlítása** céljából.

- Népszerű benchmarkok: pl. Phoronix Test Suite
- Benchmarking as a Service: cloudharmony.com
- Google PerfKitBenchmarker és PerfKitExplorer

VM teljesítmény-jellemzők térképe [9]



Felhő-specifikus (és fontos): variabilitás [11]

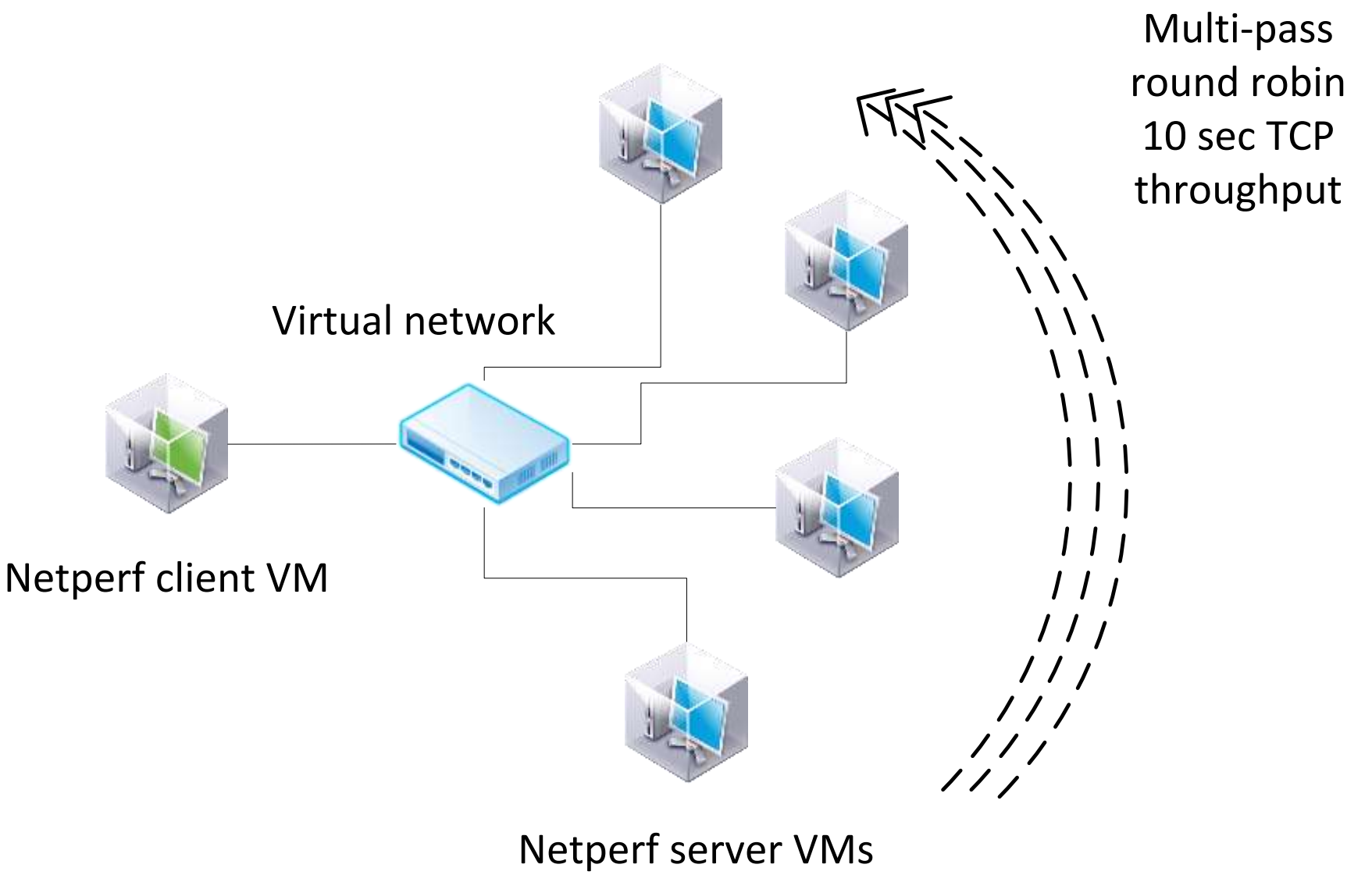
- Teljesítmény-stabilitás (**stability**): a rendelkezésre bocsátott erőforrások képessége időben állandó teljesítményt nyújtani

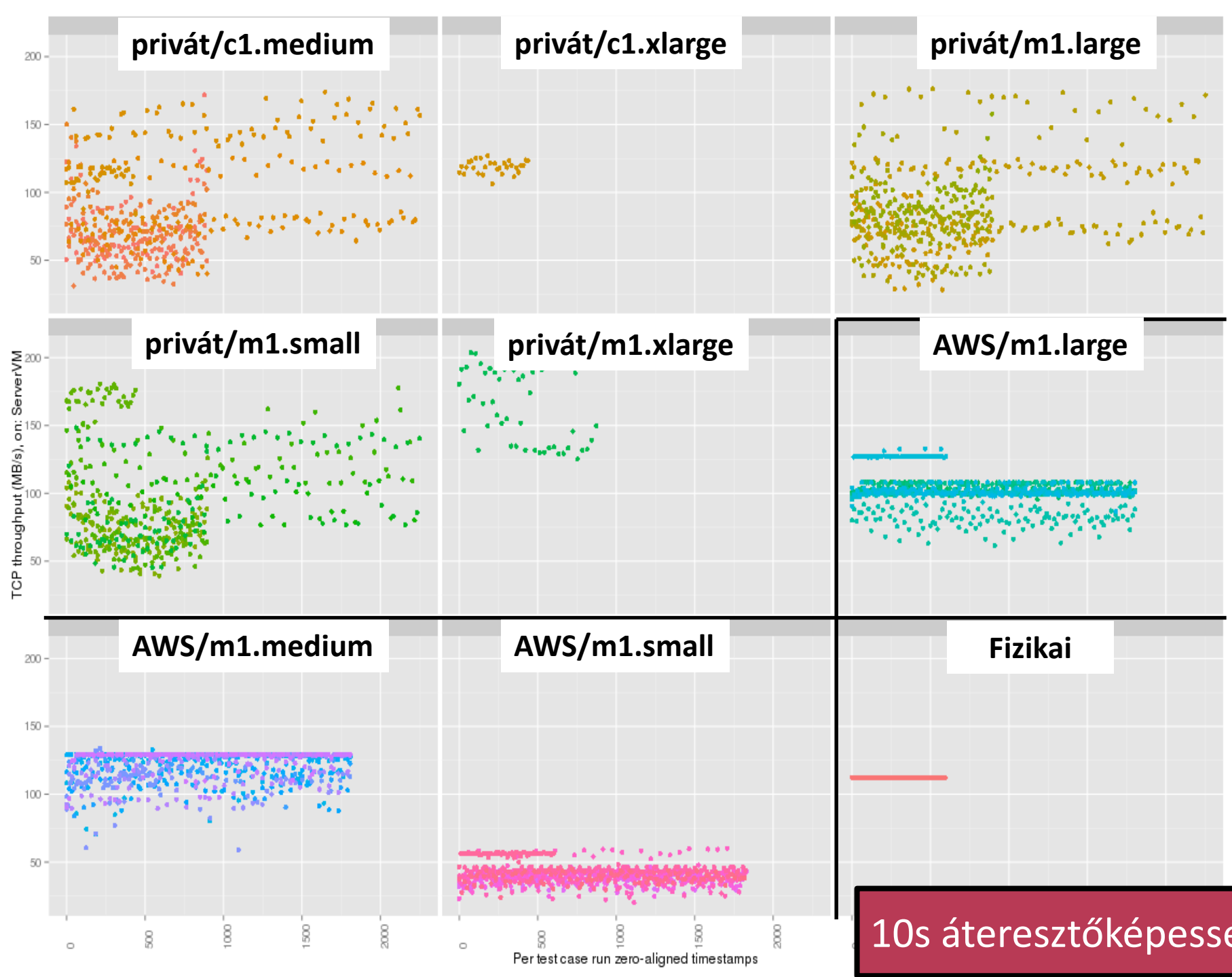
- Teljesítmény-homogenitás (**homogeneity**): bizonyosság abban, hogy az erőforrás-teljesítmény rendelkezésre álló példányok vizsgálata alapján jól előrejelezhető

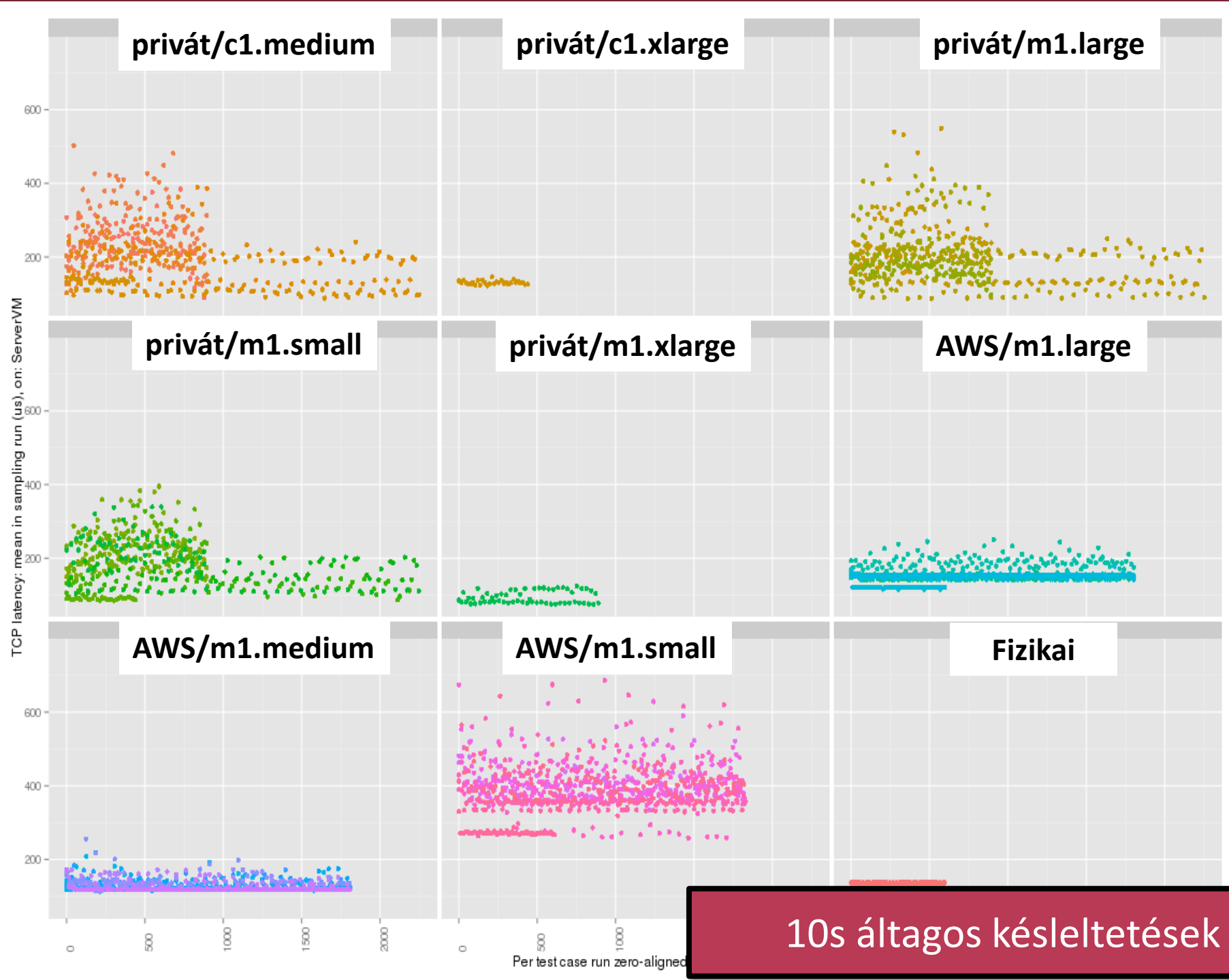
Microbenchmark támogatás: hálózat [10]

Capacity	Metrics	Benchmark
Transaction Speed	Max Number of Transfer Sessions	SPECweb 2005 [22]
Availability	Packet Loss Frequency	Badabing Tool [43]
Latency	Correlation between Total Runtime and Communication Time	Application Suite [30]
	TCP/UDP/IP Transfer Delay (s, ms)	CARE [45]
		Ping [5]
		Send 1 byte data [20]
		Latency Sensitive Website [5]
		Badabing Tool [43]
	MPI Transfer Delay (s, μs)	HPCC: b_eff [42]
		Intel MPI Bench [18]
		mpptest [8]
		OMB-3.1 with MPI [44]
Reliability	Connection Error Rate	CARE [45]
	Probe Loss Rate	Badabing Tool [43]
Data Throughput	TCP/UDP/IP Transfer bit/Byte Speed (bps, Mbps, MB/s, GB/s)	iperf [5]
		Private tools TCPTest/UDPTest [43]
		SPECweb 2005 [22]
		Upload/Download/ Send large size data[23]
	MPI Transfer bit/Byte Speed (bps, MB/s, GB/s)	HPCC: b_eff [42]
		Intel MPI Bench [18]
		mpptest [8]
		OMB-3.1 with MPI [44]

Példa: hálózat benchmarkolás netperffel




privát/c1.medium
privát/c1.xlarge
privát/m1.large
privát/m1.small
privát/m1.xlarge
AWS/m1.large
AWS/m1.medium
AWS/m1.small
Fizikai
TCP throughput (MB/s), on: ServerVM
Per test case run zero-aligned timestamps
10s áteresztőképességek

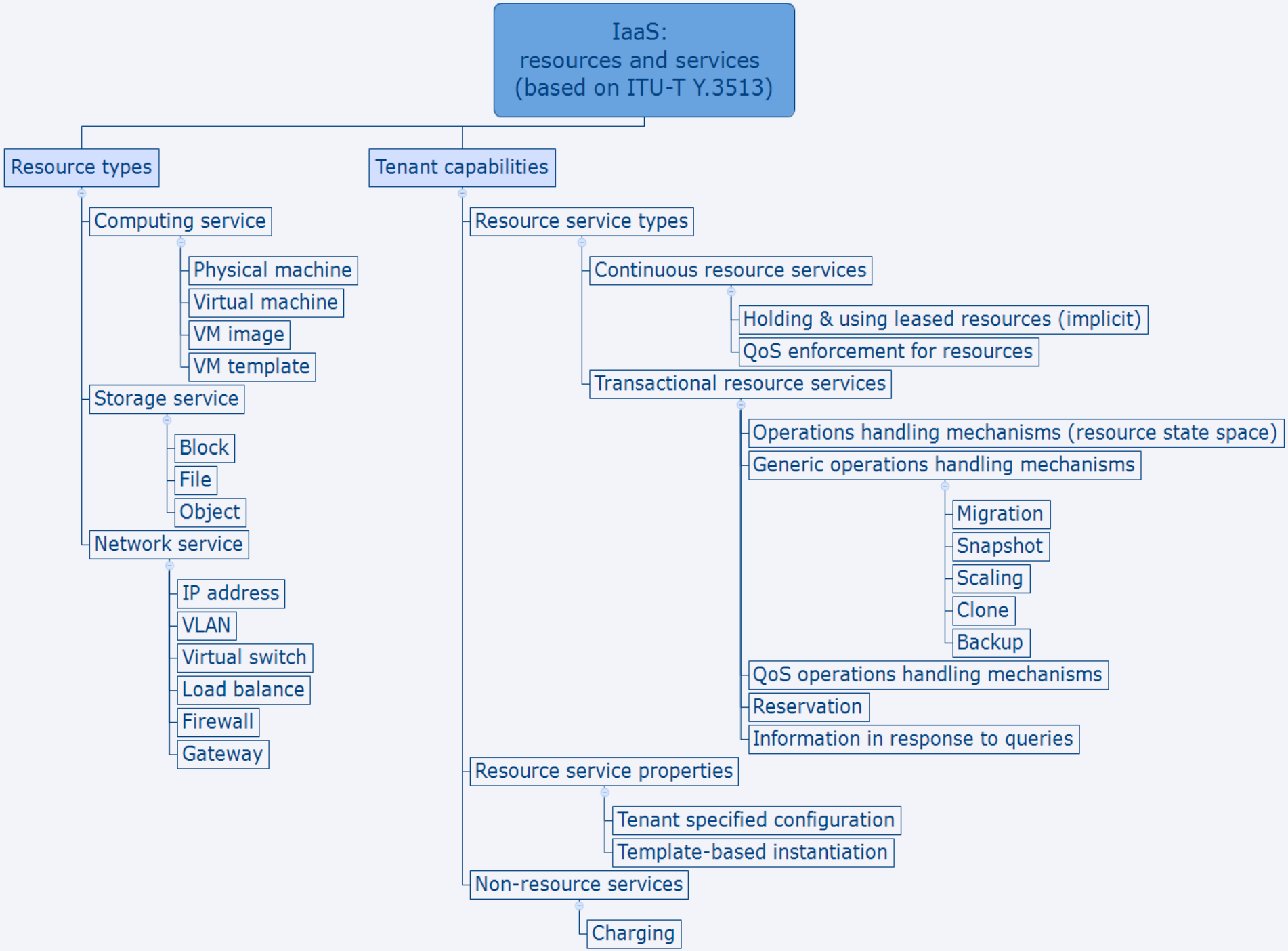


10s általagos késleltetések

Hivatkozások

- [1] Weinman, J. (2012). Cloudonomics: The Business Value of Cloud Computing. John Wiley & Sons.
- [2] Banga, G., & Druschel, P. (1997). Measuring the Capacity of a Web Server. In *USENIX Symposium on Internet Technologies and Systems*. USENIX Association. Retrieved from https://www.usenix.org/publications/library/proceedings/usits97/full_papers/banga/banga.pdf
- [3] https://blog.twitter.com/2013/new-tweets-per-second-record-and-how
- [4] Leong, L. et al. (2013.) Gartner Magic Quadrant for Cloud Infrastructure as a Service. Gartner. http://www.gartner.com/technology/reprints.do?id=1-1IMDMZ5&ct=130819&st=sb
- [5] Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing (p. 3).
- [6] Weinman, J. (2011). Smooth Operator: The Value of Demand Aggregation. Retrieved from http://www.joeweinman.com/Resources/Joe_Weinman_Smooth_Operator_Demand_Aggregation.pdf
- [7] http://en.wikipedia.org/wiki/Central_limit_theorem
- [8] CSMIC. (2011). Service Measurement Index Version 1.0. Retrieved from http://csmic.org/wp-content/uploads/2011/09/SMI-Overview-110913_v1F1.pdf
- [9] Li, Z., OBrien, L., Cai, R., & Zhang, H. (2012). Towards a Taxonomy of Performance Evaluation of Commercial Cloud Services. In 2012 IEEE Fifth International Conference on Cloud Computing (pp. 344–351). IEEE. doi:10.1109/CLOUD.2012.74
- [10] Li, Z., O'Brien, L., Zhang, H., & Cai, R. (2012). On a Catalogue of Metrics for Evaluating Commercial Cloud Services. In 2012 ACM/IEEE 13th International Conference on Grid Computing (pp. 164–173). IEEE. doi:10.1109/Grid.2012.15
- [11] J. Dejun, G. Pierre, and C. Chi, "EC2 performance analysis for resource provisioning of service-oriented applications," in in *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*, A. Dan, F. Gittler, and F. Toumani, Eds. Springer Berlin Heidelberg, 2010, pp. 197–207.
- [12] http://timesmachine.nytimes.com/browser
- [13] http://open.blogs.nytimes.com/2008/05/21/the-new-york-times-archives-amazon-web-services-timesmachine/
- [14] Rajaraman, A., & Ullman, J. D. (2011). Mining of Massive Datasets. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139058452
- [15] International Technical Support Organization. IBM InfoSphere Streams: Harnessing Data in Motion. September 2010. http://www.redbooks.ibm.com/abstracts/sg247865.html

“Mi lehet fontos”?
Platform hibák és minőség


IaaS:
resources and services
(based on ITU-T Y.3513)
Resource types
Computing service
Physical machine
Virtual machine
VM image
VM template
Storage service
Block
File
Object
Network service
IP address
VLAN
Virtual switch
Load balance
Firewall
Gateway
Tenant capabilities
Resource service types
Continuous resource services
Holding & using leased resources (implicit)
QoS enforcement for resources
Transactional resource services
Operations handling mechanisms (resource state space)
Generic operations handling mechanisms
Migration
Snapshot
Scaling
Clone
Backup
QoS operations handling mechanisms
Reservation
Information in response to queries
Resource service properties
Tenant specified configuration
Template-based instantiation
Non-resource services
Charging

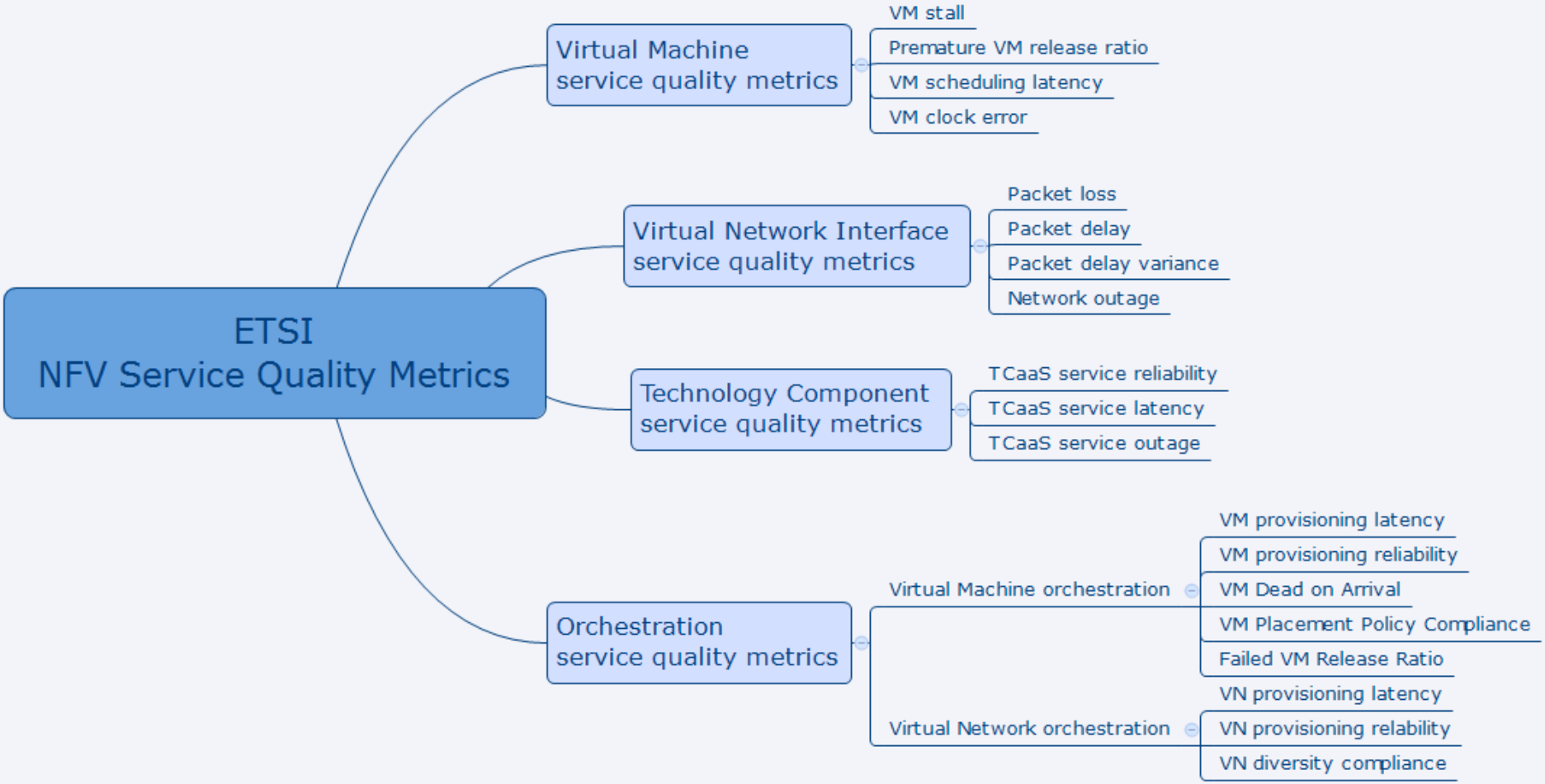
IaaS resource operations handling mechanisms

Resource type	Operations handling mechanisms
Physical machine	start, shutdown, hibernate, wakeup
Virtual machine	create, delete, start, shutdown, suspend, restore, hibernate, wakeup
VM Image	add, import, store, register, deregister, query, update, delete, export
VM template	upload, update, disable, enable, query, delete
Storage	create, attach, detach, query, delete
IP address	apply, bind, unbind, query, release

Fő veszélyforrások (*platform failure modes*)

- Latency variation of VM services
  - e.g. resource accesses, real-time notifications, system calls, instruction execution, interrupt handling
- Variability of resource service performance
- VM failure
- Nondelivery of configured VM capacity
- Delivery of degraded VM capacity
- Changes in the tail latency of IaaS constituent services
- (VM) clock even jitter
- (VM) clock drift
- Failed or slow allocation and startup of VM instance

IaaS minőségi jellemzők



Felhők összehasonlítása

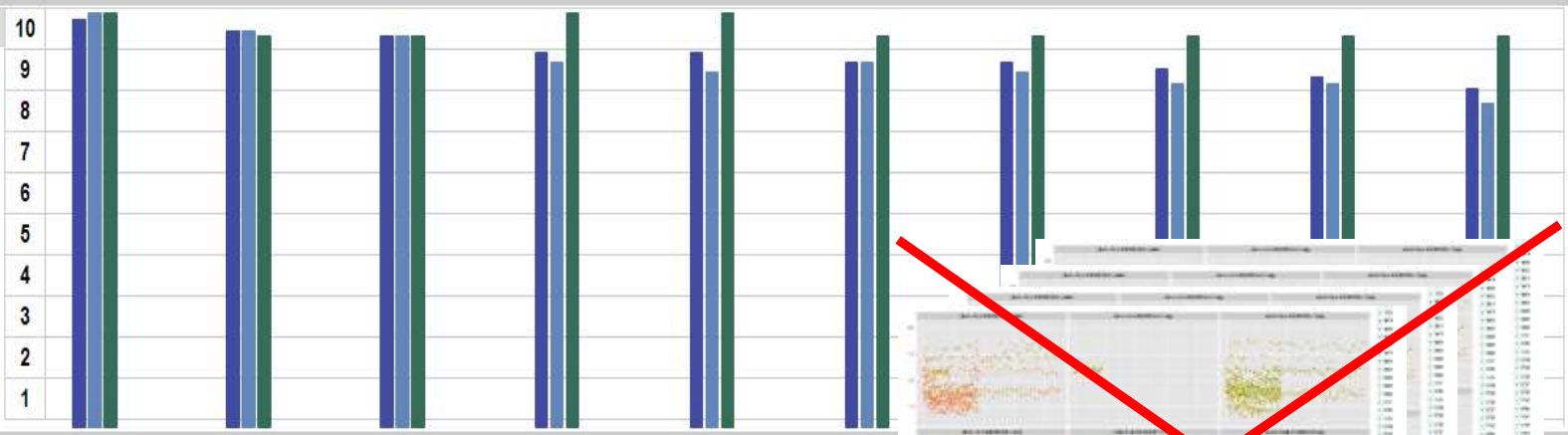
Data exploration versus ranking/KPIs

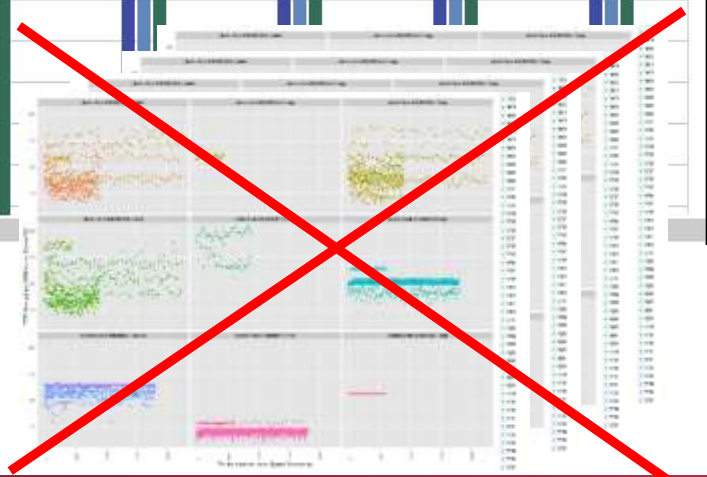
Rank	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
	Cuisinart Brew Central	Mr. Coffee SJX	Melitta 10-Cup Coffee Maker	BUNN Phase Brew	Cuisinart Automatic Brew and Serve	Mr. Coffee FTX	DeLonghi 12-Cup Drip Coffee Maker	Zojirushi Fresh Brew	Kalorik 12-Cup Coffee Maker	Bonavita 8-Cup Coffee Maker
Reviewer Comments	Read Review	Read Review	Read Review	Read Review	Read Review	Read Review	Read Review	Read Review	Read Review	Read Review
Ratings	9.90	9.48	9.38	9.05	8.95	8.85	8.75	8.58	8.48	8.18

10-9 Excellent
8-6 Good
5-4 Average
3-2 Poor
1-0 Bad

Print/Email

- Overall Rating
- Basic Features
- Convenience Features
- Help & Support





IaaS ranking/KPIs

- ~~Emergent~~ research topic
  - e.g. CloudGenius, SMICloud
  - Roots: Web Service selection

- Industrial significance
  - „Service Measurement Index” (CA & Carnegie Mellon)
  - Commercial solutions?

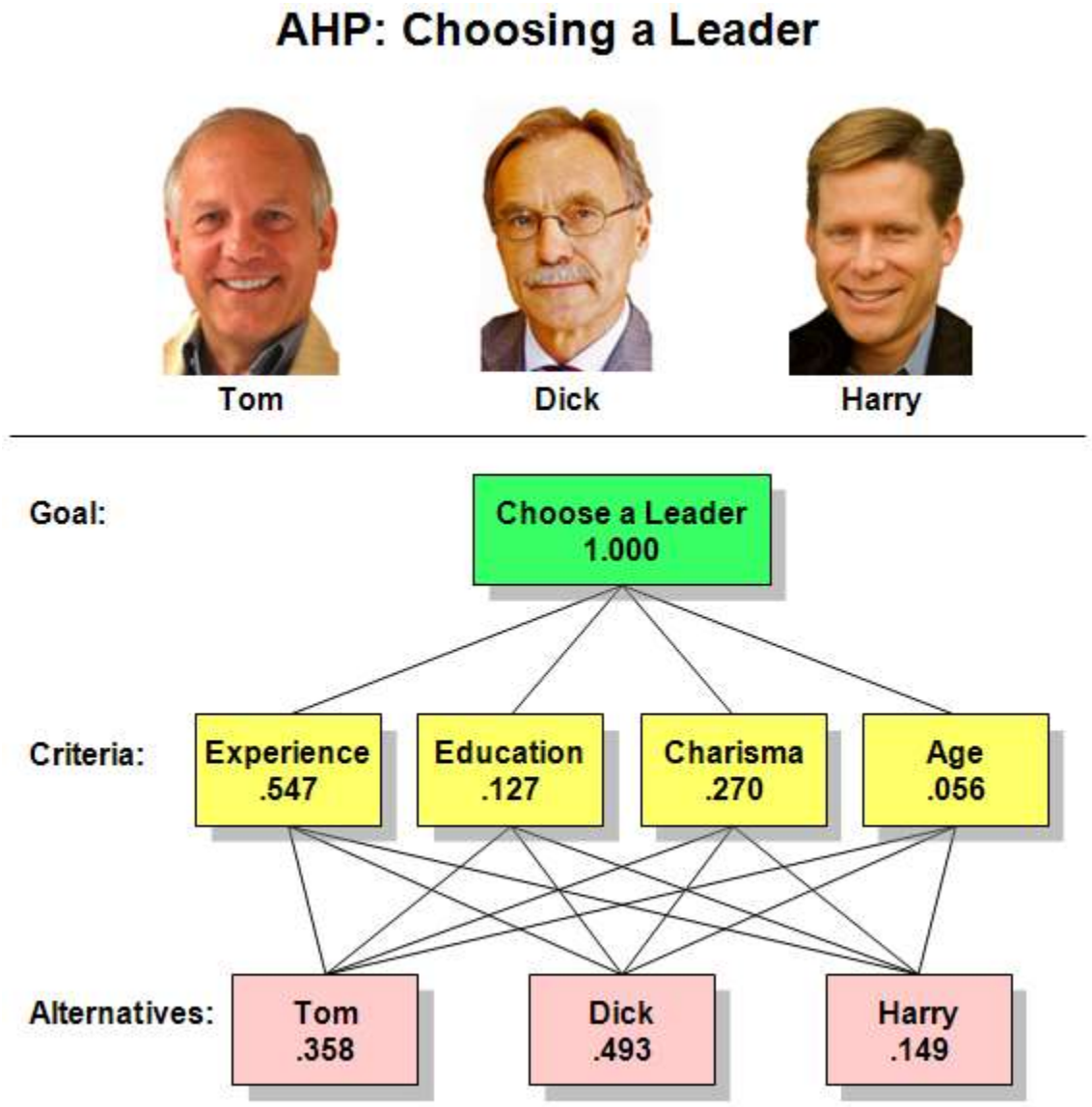
- Ranking/sel. over different properties: MCDM

The Service Measurement Index

Attribute	SMI Attribute Definition
Accuracy	The extent to which a service adheres to its requirements.
Functionality	The specific features provided by a service.
Suitability	How closely the capabilities of the proposed service match the features needed by the client.
Interoperability	The ability of a service to easily interact with other services (from the same cloud service provider and from other cloud service providers).
Service Response Time	An indicator of the time between when a service is requested and when the response is available.

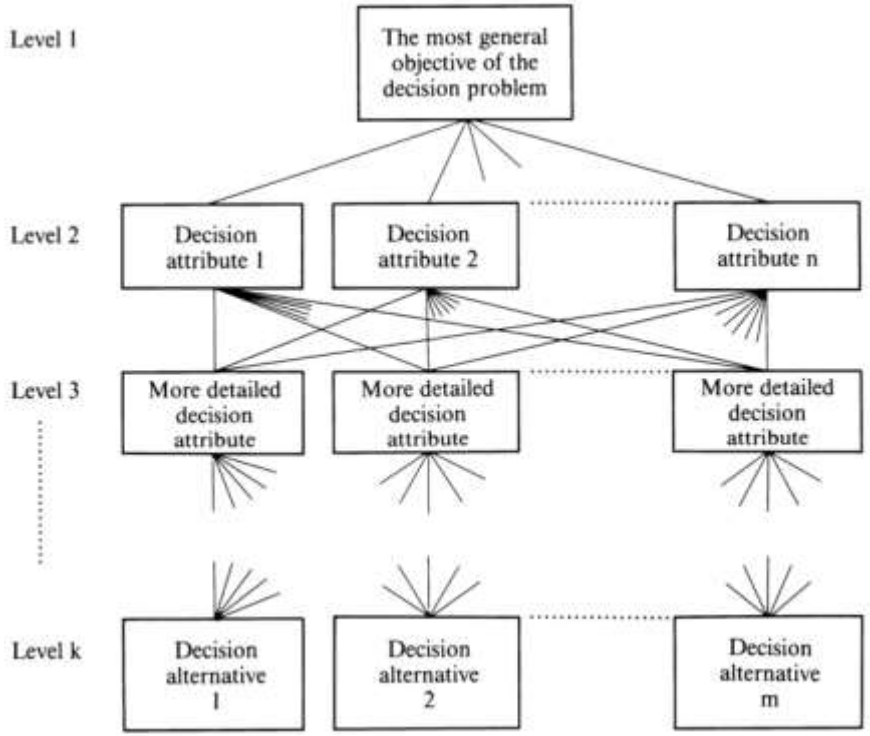
SMICloud: AHP

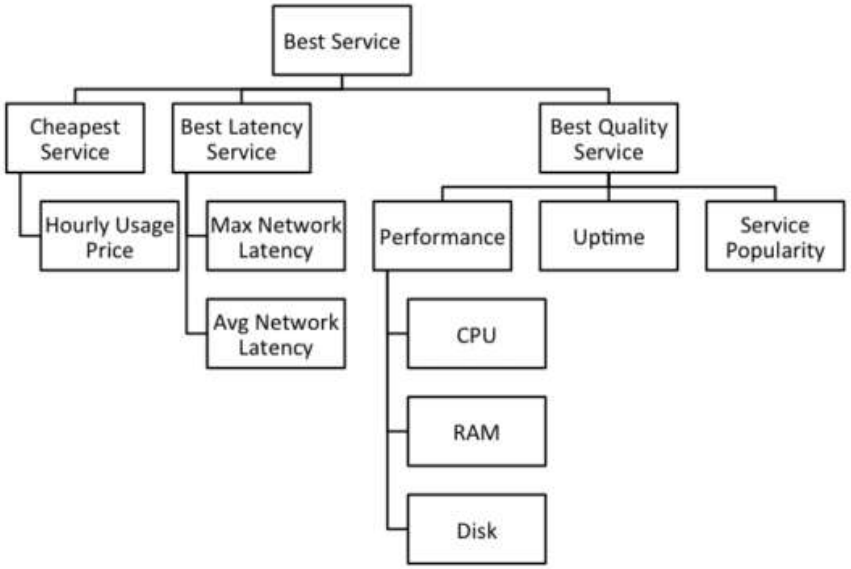
The Analytic Hierarchy Process (simplified)



- Hierarchy
- Criteria: relative weights (importance)
- Covering criteria weights
- Alternative weights by c.c.
- „Synthesize” (weighted summation style)

AHP with true hierarchy





CloudGenius

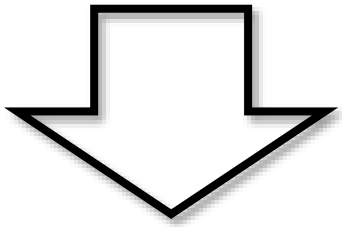
„Level” priority vectors: pairwise comparisons

Intensity of Importance	*Definition*	*Explanation*
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or slight	
3	Moderate importance	Experience and judgment slightly favor one activity over another
4	Moderate plus	
5	Strong importance	Experience and judgment strongly favor one activity over another
6	Strong plus	
7	Very strong or demonstrated importance	An activity is favored very strongly over another; its dominance demonstrated in practice
8	Very,very strong	
9	Extreme importance	The evidence favoring one activity over another is of the highest possible order of affirmation
1.1–1.9	When activities is added to 1 tc appropriate	A better alternative way to assigning the small
Reciprocals of above	If activity *i* has numbers assigr with activity *j*, value when cor	
Measurements from ratio scales		

Performance criterion 1	Performance criterion 2	Judgment	Fundamental scale: 1-to-2 ratio
Storage	Memory	We usually **strongly favor** better storage perf. over better memory perf.	5
Storage	Comp.	We usually **slightly favor** better computation perf. over better storage perf.	1/3
Storage	Comm.	We usually **slightly favor** better communication perf. over better storage perf.	1/3
Memory	Comp.	We usually **very strongly** favor better computation perf. over better memory perf.	1/7
Memory	Comm.	We usually **very strongly** favor better communication perf. over better memory perf.	1/7
Comp.	Comm.	Both are **equally important**	1

„Level” priority vectors: pairwise comparisons

$$W_{\text{goal}} = \begin{array}{c|cccc} & \text{Storage} & \text{Memory} & \text{Computation} & \text{Communication} \\ \hline \text{Storage} & 1 & 5 & \frac{1}{3} & \frac{1}{3} \\ \text{Memory} & \frac{1}{5} & 1 & \frac{1}{7} & \frac{1}{7} \\ \text{Computation} & 3 & 7 & 1 & 1 \\ \text{Communication} & 3 & 7 & 1 & 1 \end{array}$$

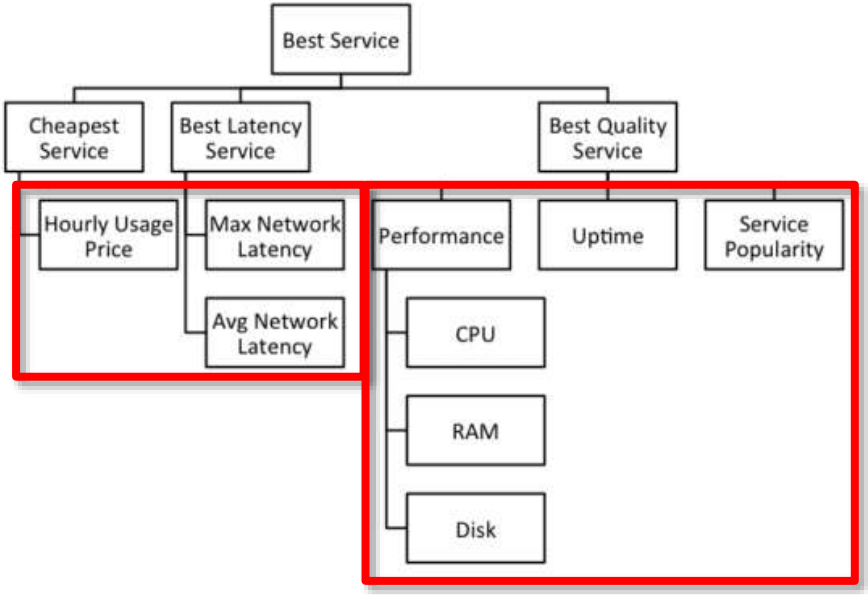


$$w_{goal} = \begin{bmatrix} \textit{Storage} & \textit{Memory} & \textit{Computation} & \textit{Communication} \\ 0.15995212 & 0.04682919 & 0.39660935 & 0.39660935 \end{bmatrix}$$

Not *mandatory* – e.g. measurement results on rate scales

Synthesis on covering criteria

- Evaluate alternatives on c.c. (weights);
- „Synthesize” using c.c. weights

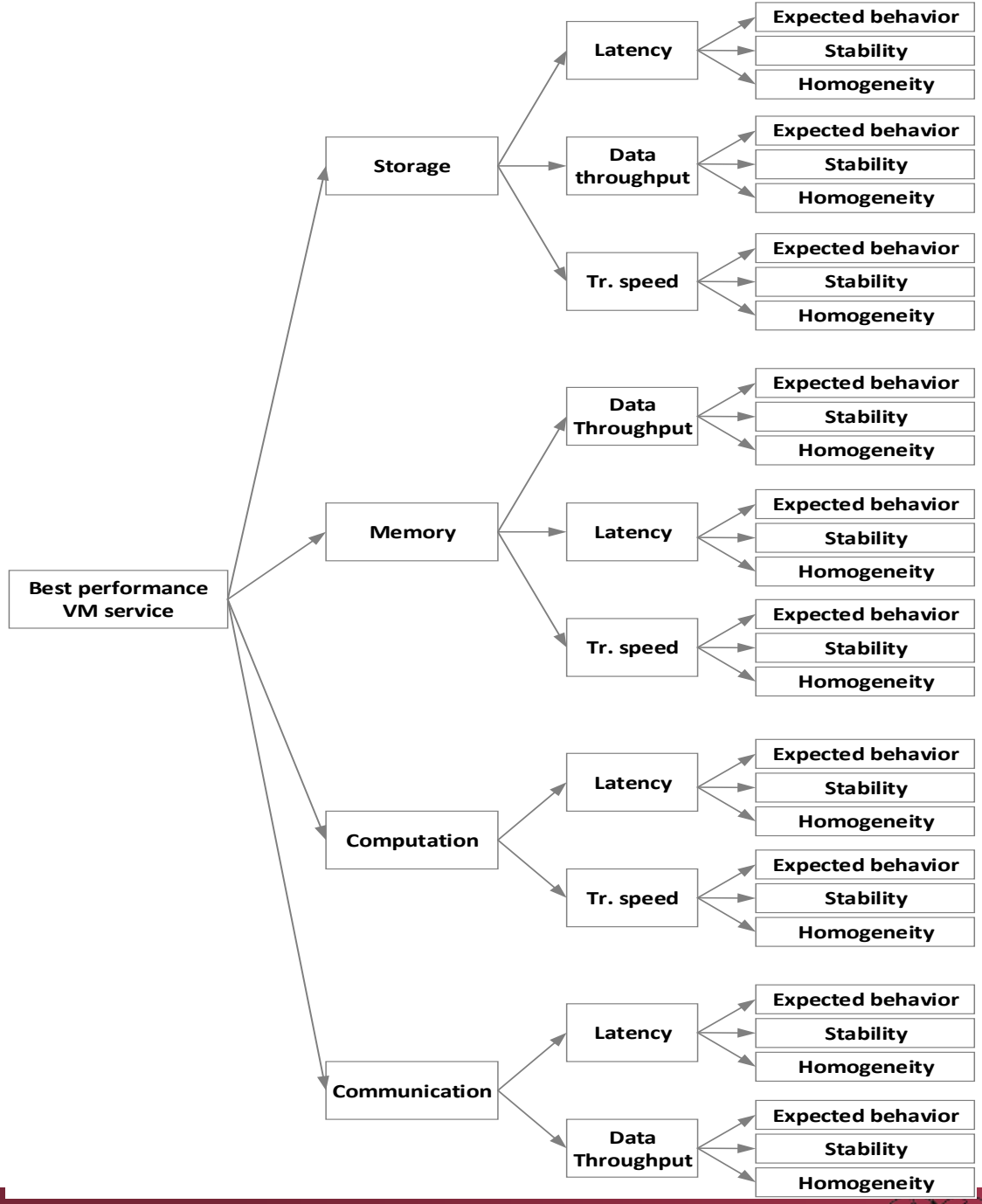


$$x_j^D = \sum_{i=1}^{m} \frac{w_i}{w} \frac{a_{ij}}{\sum_{k=1}^{n} a_{ik}} = \sum_{i=1}^{m} \left(\frac{w_i}{w} \frac{1}{\sum_{k=1}^{n} a_{ik}} \right) a_{ij}$$

$$x_j^I = \sum_{i=1}^{m} \frac{w_i}{w} \frac{a_{ij}}{\max_k a_{ik}} = \sum_{i=1}^{m} \left(\frac{w_i}{w} \frac{1}{\max_k a_{ik}} \right) a_{ij}$$

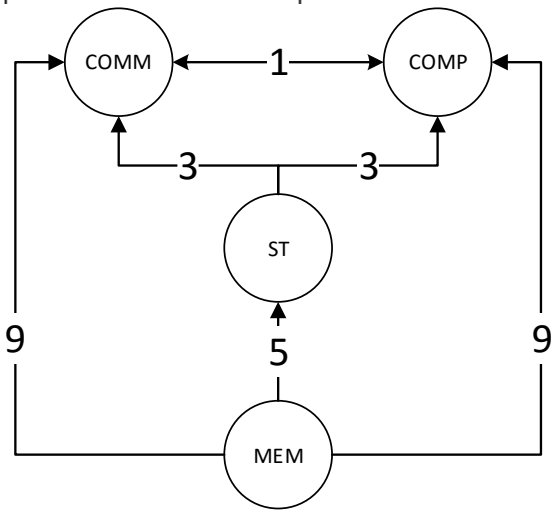
$$x_j^R = \sum_{i=1}^{m} \frac{w_i}{w} \frac{1}{a_i^*} a_{ij}$$

A hierarchy


Best performance VM service
Storage
Latency
Data throughput
Tr. speed
Memory
Data Throughput
Latency
Tr. speed
Computation
Latency
Tr. speed
Communication
Latency
Data Throughput
Expected behavior
Stability
Homogeneity

First level

Performance criterion 1	Performance criterion 2	Judgment	Fundamental scale: 1-to-2 ratio
Storage	Memory	We usually **strongly favor** better storage perf. over better memory perf.	5
Storage	Comp.	We usually **slightly favor** better computation perf. over better storage perf.	1/3
Storage	Comm.	We usually **slightly favor** better communication perf. over better storage perf.	1/3
Memory	Comp.	We usually **very strongly** favor b computation perf. over better memory p	
Memory	Comm.	We usually **very strongly** favor b communication perf. over better memory	
Comp.	Comm.	Both are **equally important**	



$$w_{goal} = [\ \underset{0.15995212}{Storage}\quad \underset{0.04682919}{Memory}\quad \underset{0.39660935}{Computation}\quad \underset{0.39660935}{Communication}\ ]$$

Second level

$$w_{comm} = \begin{bmatrix} \textit{Latency} & \textit{Data throughput} \\ 0.25 & 0.75 \end{bmatrix}$$

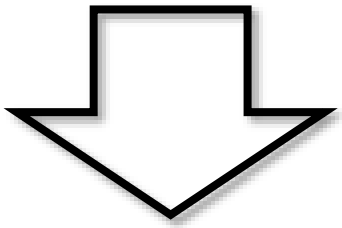
$$w_{comp} = \begin{bmatrix} \textit{Latency} & \textit{Tr. speed} \\ 0.5 & 0.5 \end{bmatrix}$$

$$w_{mem} = \begin{bmatrix} \textit{Data throughput} & \textit{Latency} & \textit{Tr. speed} \\ 0.334 & 0.333 & 0.333 \end{bmatrix}$$

$$w_{st} = \begin{bmatrix} \textit{Latency} & \textit{Data throughput} & \textit{Tr. speed} \\ 0.6 & 0.2 & 0.2 \end{bmatrix}$$

Third level

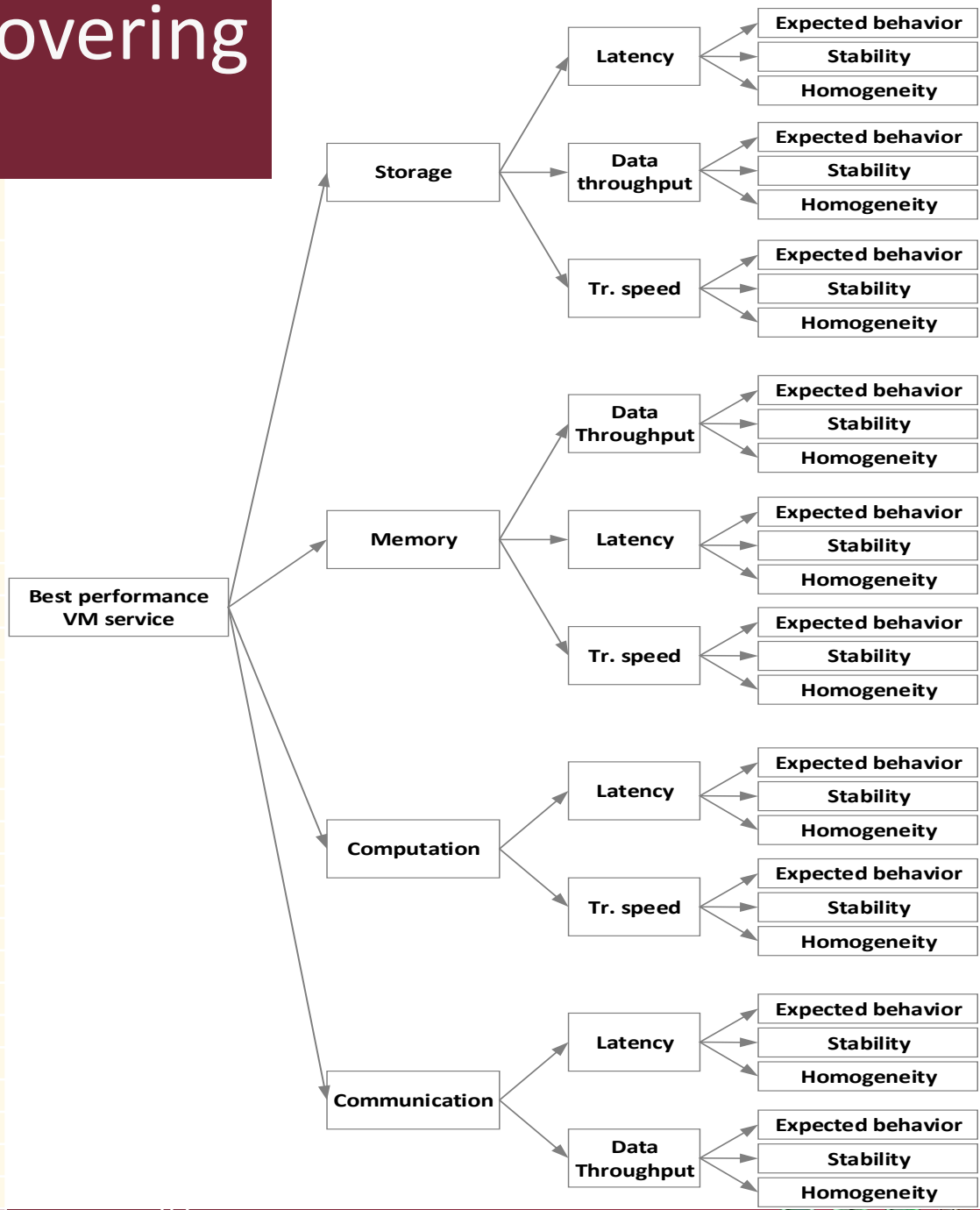
$$W_{level3} = \begin{array}{c|ccc} & \textit{Expected behavior} & \textit{Stability} & \textit{Homogeneity} \\ \hline \textit{Expected behavior} & 1 & 4 & 9 \\ \textit{Stability} & \frac{1}{4} & 1 & 5 \\ \textit{Homogeneity} & \frac{1}{9} & \frac{1}{5} & 1 \end{array}$$



$$w_{level3} = \begin{array}{c} \\ [\end{array} \begin{array}{ccc} \textit{Expected behavior} & \textit{Stability} & \textit{Homogeneity} \\ 0.70852417 & 0.23114819 & 0.06032764 \end{array} \begin{array}{c} \\ ] \end{array}$$

Global weights for covering criteria

Weight rank	Covering criterion (abbrev.)	Global weight (%)
1	COMM_dat_thr_exp	21,08
2	COMP_latency_exp	14,05
3	COMP_tr_speed_exp	14,05
4	COMM_latency_exp	7,03
5	COMM_dat_thr_stab	6,88
6	ST_latency_exp	6,8
7	COMP_latency_stab	4,58
8	COMP_tr_speed_stab	4,58
9	COMM_latency_stab	2,29
10	ST_dat_thr_exp	2,27
11	ST_tr_speed_exp	2,27
12	ST_latency_stab	2,22
13	MEM_latency_exp	1,99
14	COMM_dat_thr_hom	1,79
15	COMP_latency_hom	1,2
16	COMP_tr_speed_hom	1,2
17	MEM_dat_thr_exp	1,11
18	ST_dat_thr_stab	0,74
19	ST_tr_speed_stab	0,74
20	MEM_tr_speed_exp	0,66
21	MEM_latency_stab	0,65
22	COMM_latency_hom	0,6
23	ST_latency_hom	0,58
24	MEM_dat_thr_stab	0,36
25	MEM_tr_speed_stab	0,22
26	ST_dat_thr_hom	0,19
27	ST_tr_speed_hom	0,19
28	MEM_latency_hom	0,17
29	MEM_dat_thr_hom	0,09
30	MEM_tr_speed_hom	0,06



Reading material

- M. Zeleny, “Multiple criteria decision making: Eight concepts of optimality,” *Hum. Syst. Manag.*, vol. 17, pp. 97–107, 1998.
- T. L. Saaty, “How to make a decision: The analytic hierarchy process,” *Eur. J. Oper. Res.*, vol. 48, no. 1, pp. 9–26, Sep. 1990.
- T. L. Saaty, “Decision making with the analytic hierarchy process,” *Int. J. Serv. Sci.*, vol. 1, no. 1, p. 83, 2008.
- S. K. Garg, S. Versteeg, and R. Buyya, “SMICloud: A Framework for Comparing and Ranking Cloud Services,” in *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, 2011, no. Vm, pp. 210–218.
- M. Menzel and R. Ranjan, “CloudGenius: decision support for web server cloud migration,” in *Proceedings of the 21st International Conference on World Wide Web - WWW ’12*, 2012, pp. 979–988.
- [28] T. Rapcsák, “Többszempontú döntési problémák,” 2007.

Autonóm és hibatűrő informatikai rendszerek

Motiváció

- Ezredforduló: „rendszermenedzsment-válság”

‚Computing systems’ complexity appears to be approaching the limits of human capability, yet the march toward increased interconnectivity and integration rushes ahead unabated.’ (Kephart et al. 2003)

- Beüzemelés és karbantartás költségei!
  - És egyéb minőségi jellemzői, pl. rendelkezésreállás
- N.B.: „enterprise software” nézőpont

Motiváció

As systems become more interconnected and diverse, architects are less able to anticipate and design interactions among components, leaving such issues to be dealt with at runtime. (Kephart et al. 2003)

- Történelmi perspektíva: belső adatközpontok, Tivoli et al., utility computing', dinamikus WS-ek, ~~stone knives and bearskins~~
- Ami nem látszott: cloud computing

„Autonóm számítástechnika”

Az **autonomic computing (AC, autonóm informatika)** az autonóm idegrendszert modellező rendszertervezési paradigma. A rendszer alapvető állapotváltozóiban bekövetkező változás a teljes rendszer viselkedését megváltoztató beavatkozást vált ki, amely biztosítja, hogy a rendszer egyensúlyi állapotba kerül a környezetével.

„Autonóm számítástechnika”

- 2001: IBM „manifesto” az önmenedzsment jegyében
- Fő inspiráció: az (emberi) idegrendszer
  - Tágabb értelemben a biológiai rendszerek
- Három alapvető elv
  - Szabályozási körök (*control loop*)
  - „dinamikus tervkésztés”
  - Öntudattal rendelkező (*self-aware*), reflektív rendszerek
- Rendszerszintű megközelítés
  - Automatizálás + felügyelet minden rétegben
  - Federált, heterogén komponenensek kohezívan együttműködnek

Self-* tulajdonságok

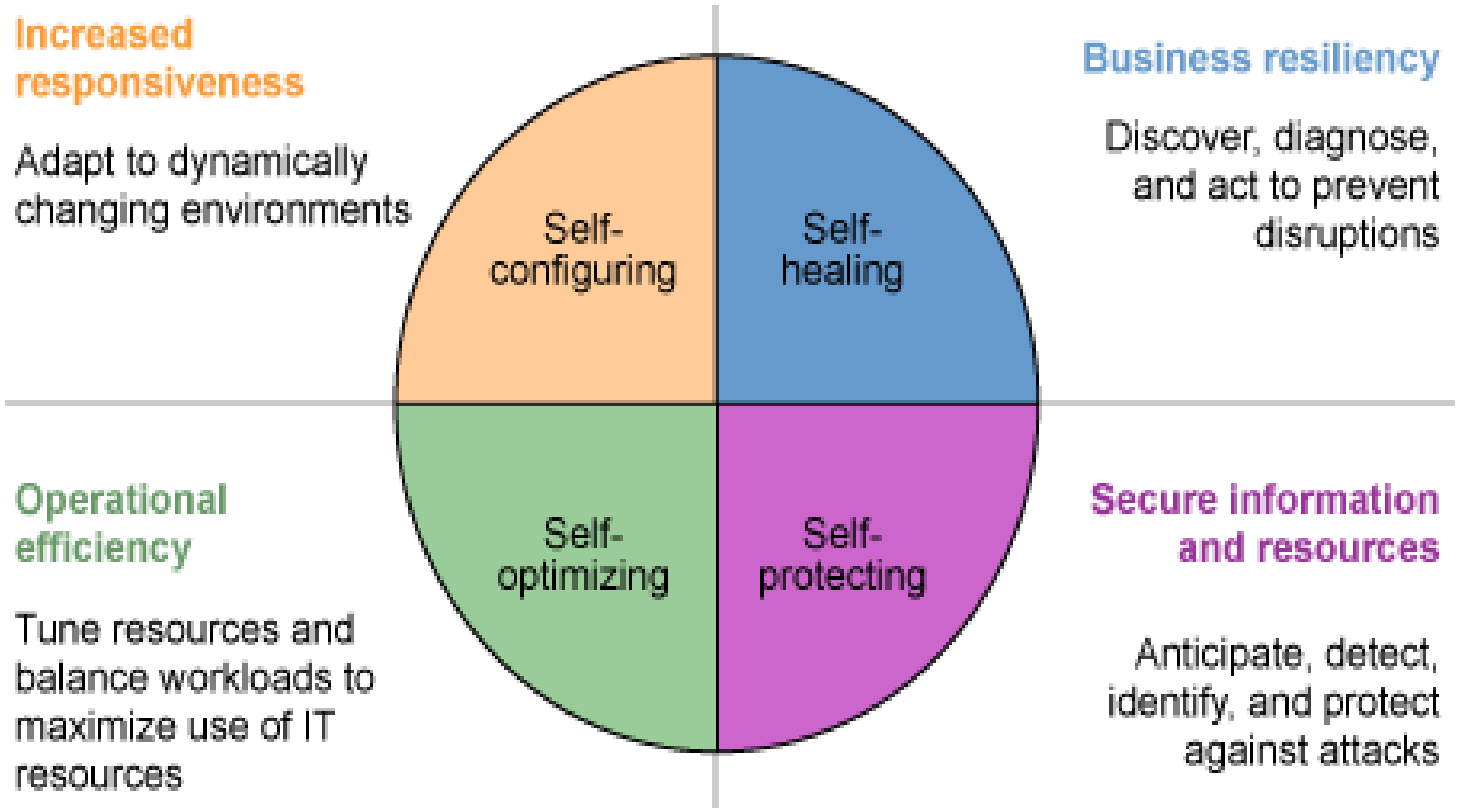
Table 1. Four aspects of self-management as they are now and would be with autonomic computing.

Concept	Current computing	Autonomic computing
Self-configuration	Corporate data centers have multiple vendors and platforms. Installing, configuring, and integrating systems is time consuming and error prone.	Automated configuration of components and systems follows high-level policies. Rest of system adjusts automatically and seamlessly.
Self-optimization	Systems have hundreds of manually set, nonlinear tuning parameters, and their number increases with each release.	Components and systems continually seek opportunities to improve their own performance and efficiency.
Self-healing	Problem determination in large, complex systems can take a team of programmers weeks.	System automatically detects, diagnoses, and repairs localized software and hardware problems.
Self-protection	Detection of and recovery from attacks and cascading failures is manual.	System automatically defends against malicious attacks or cascading failures. It uses early warning to anticipate and prevent systemwide failures.

Forrás: [1], p 43

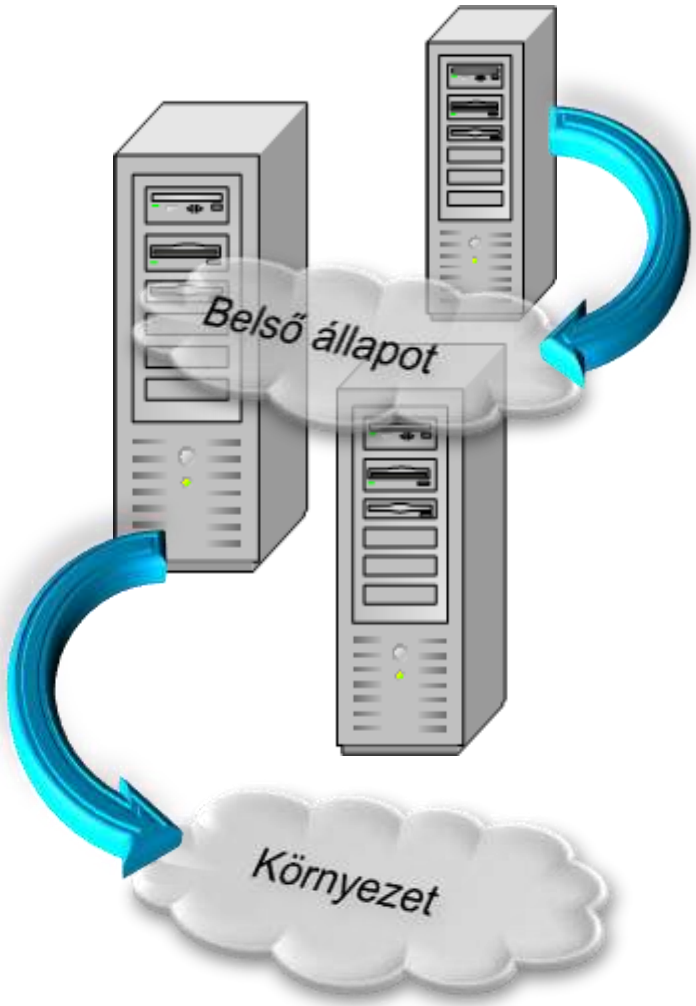
Self-* tulajdonságok

- A self-* (ön*) tulajdonságok AC rendszerek makroszkopikus tulajdonságai



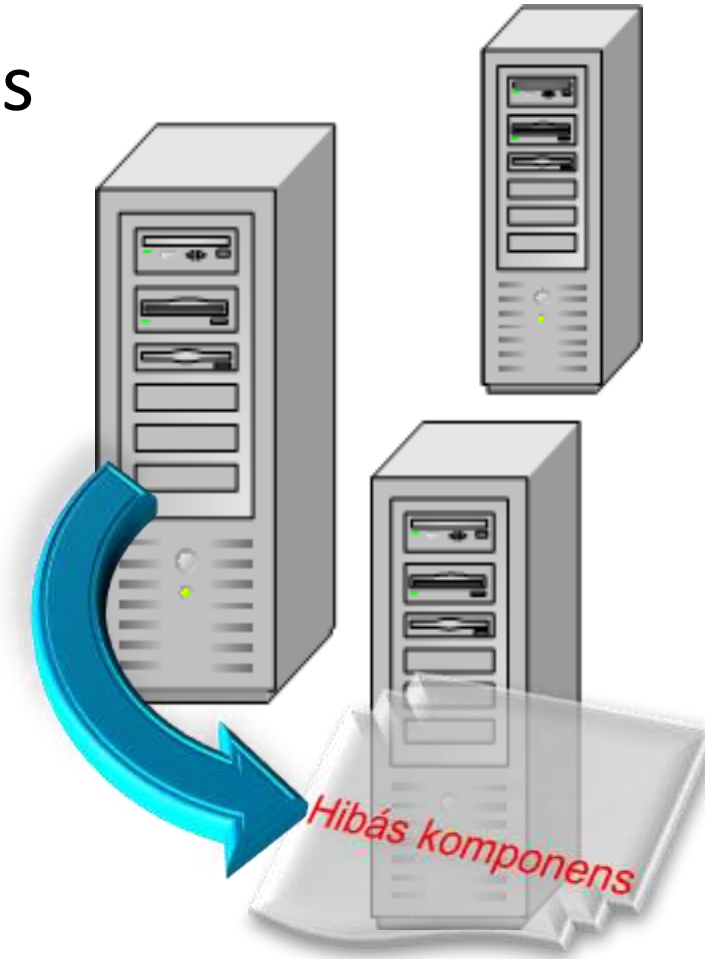
Önkonfiguráció - Self-configuration

- Automatikus adaptáció a dinamikusan változó környezethez
- Belső adaptáció
  - Komponensek hozzáadása vagy elvétele (software)
  - Futás közbeni újrakonfiguráció
- Külső adaptáció
  - A globális infrastruktúra szerint saját magát állítja be a rendszer



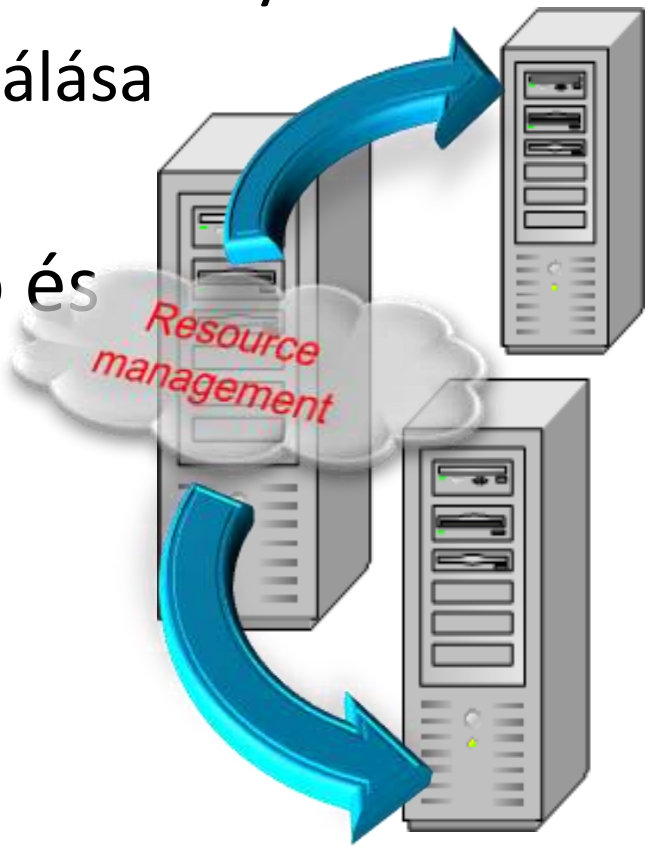
Öngyógyítás - Self-healing

- Külső zavarás felismerése, diagnosztizálása és szolgáltásmegszakítás nélküli kezelése
- Autonóm problémafelismerés és megoldás
- A hibás komponenseket
  - detektálni,
  - izolálni,
  - javítani,
  - újraintegrálni.



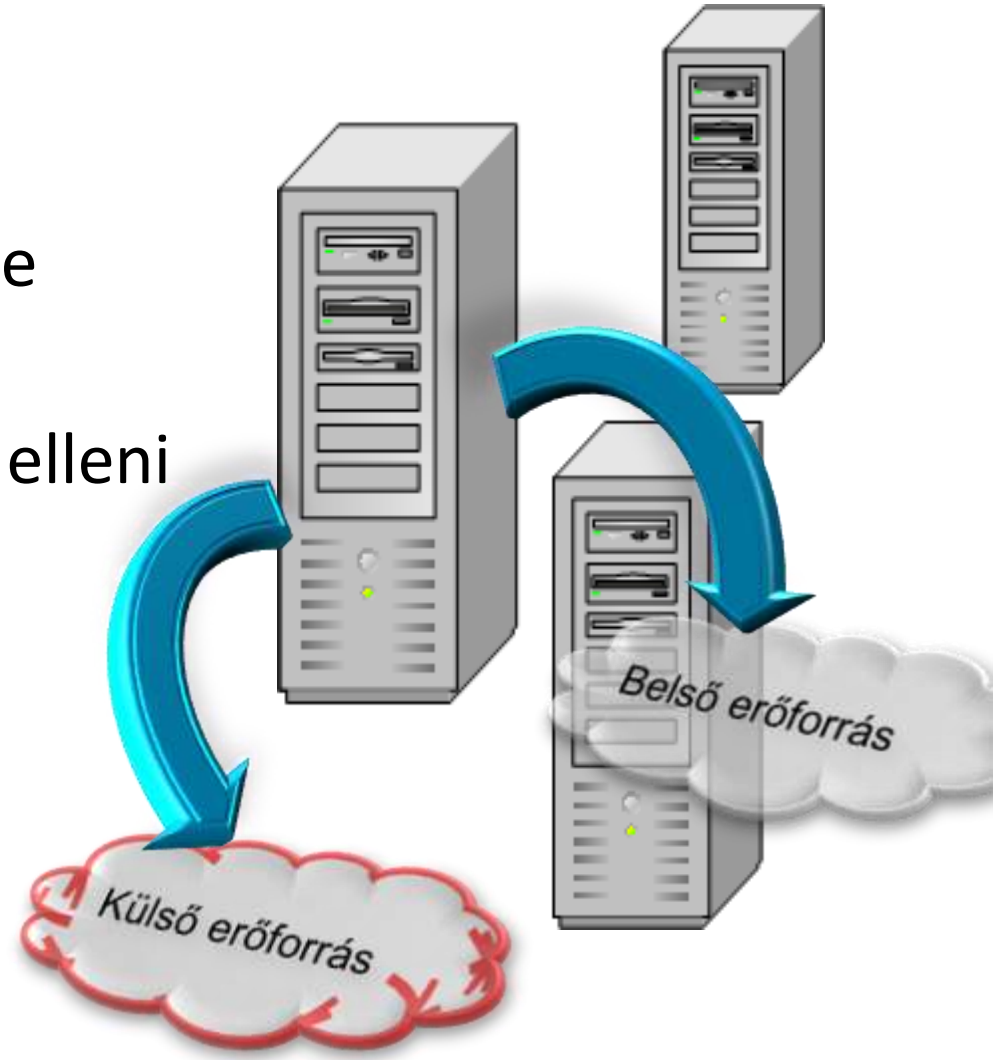
Önoptimalizáció - Self-optimization

- Erőforrások automatikus monitorozása, hangolása, felügyelete
  - Működés nem előre jelezhető körülmények között
  - Erőforrás kihasználás maximalizálása emberi beavatkozás nélkül
- Dinamikus erőforrás allokáció és terhelés-menedzsment
  - Erőforrás: tárhely, adatbázis, hálózat
  - Példa: dinamikus szerver fürtök

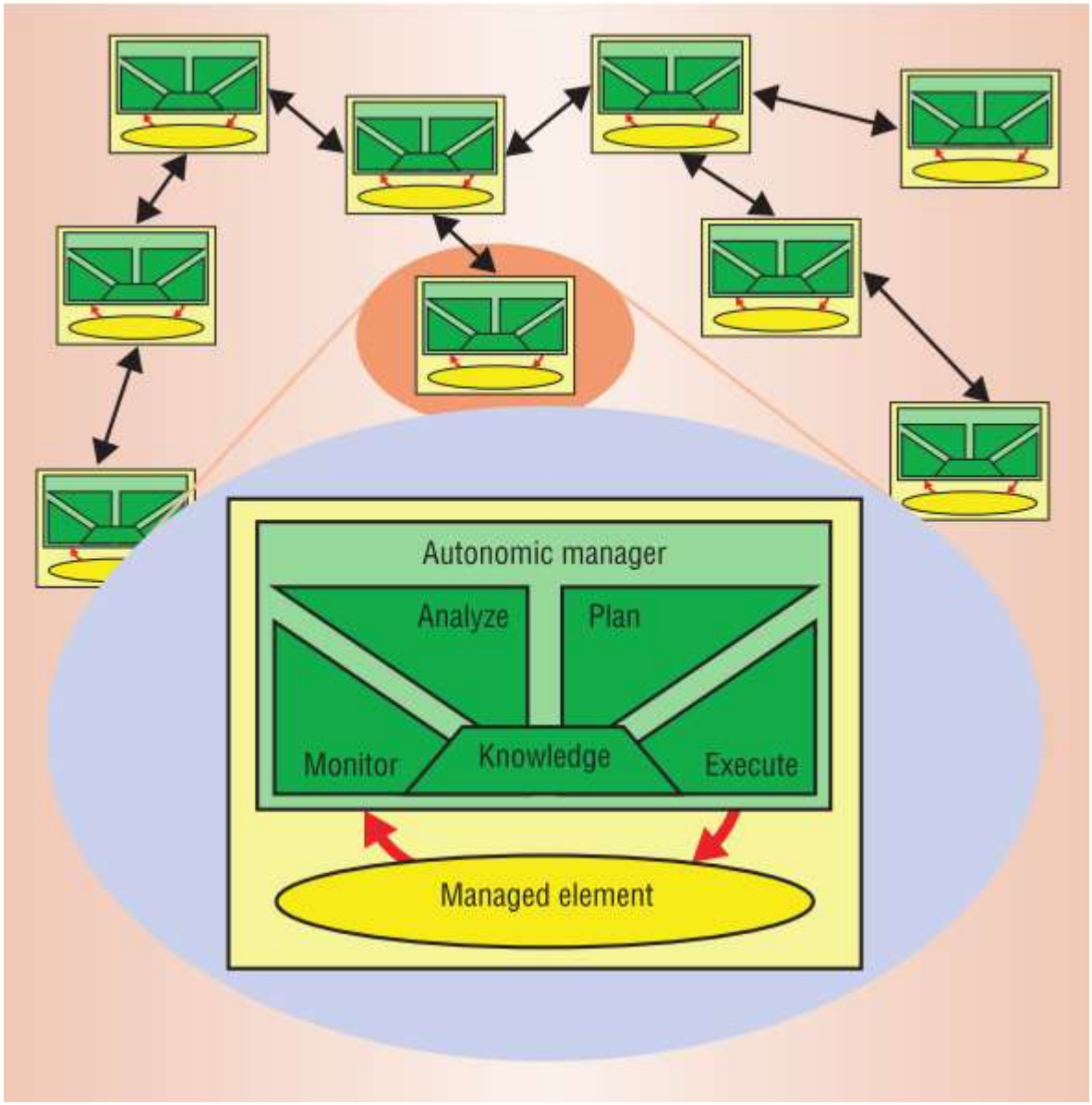


Önvédelem - Self-protection

- Támadásokra való felkészülés, detektálás, azonosítás és védelem
  - Felhasználói hozzáférés definiálása és felügyelete minden erőforrásra
  - Jogosulatlan hozzáférés elleni védelem



Megvalósítási minta: MAPE-K



Forrás: [1], p 44

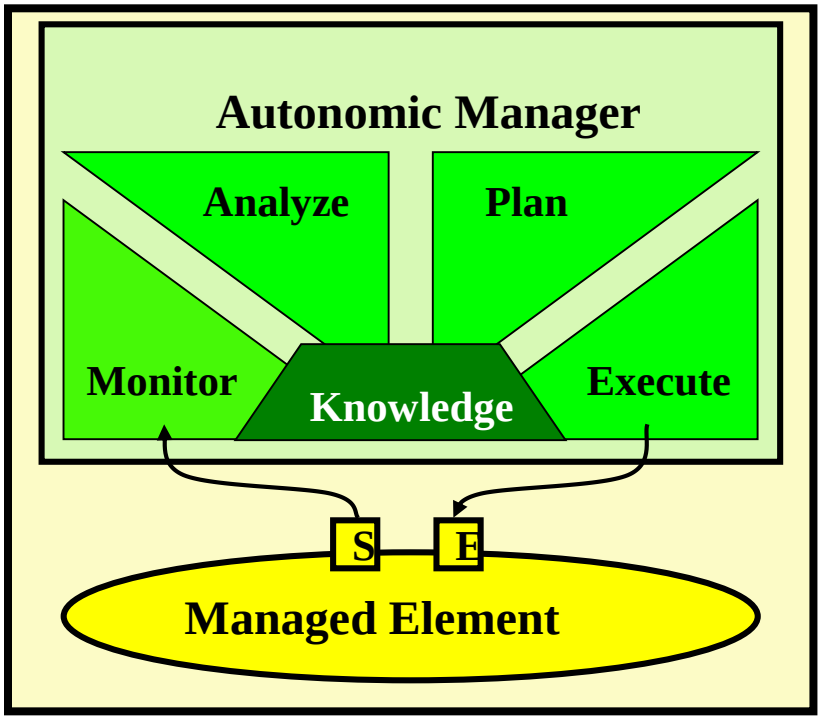
Autonomic Element - AE

- Az architektúra alapeleme a

- Felügyelt egységből
  - Adatbázis, alkalmazásszerver , stb
- És autonóm menedzserből álló

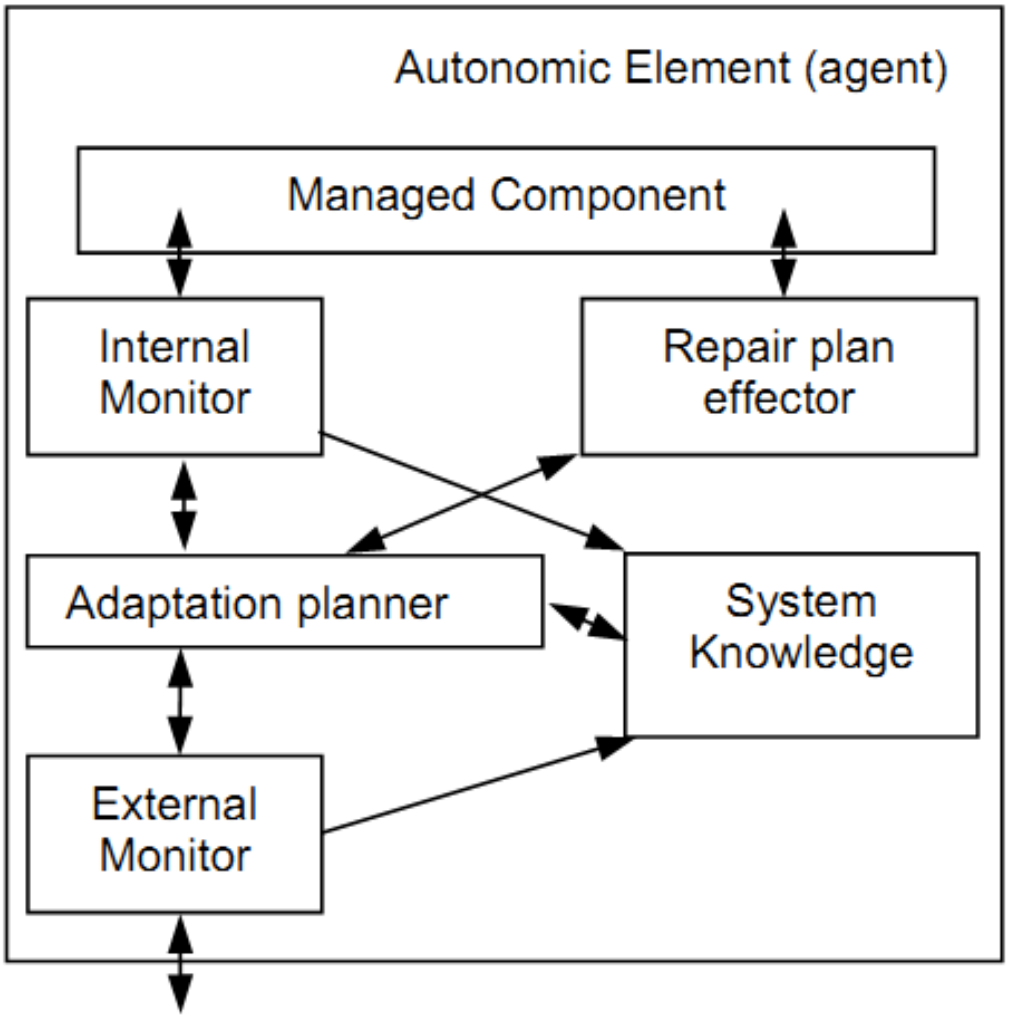
Autonóm egység

- Feladatai:
  - A funkcionalitás nyújtása
  - Saját viselkedésének felügyelete a self-* tulajdonságok alapján
  - Együttműködés más autonóm egységekkel



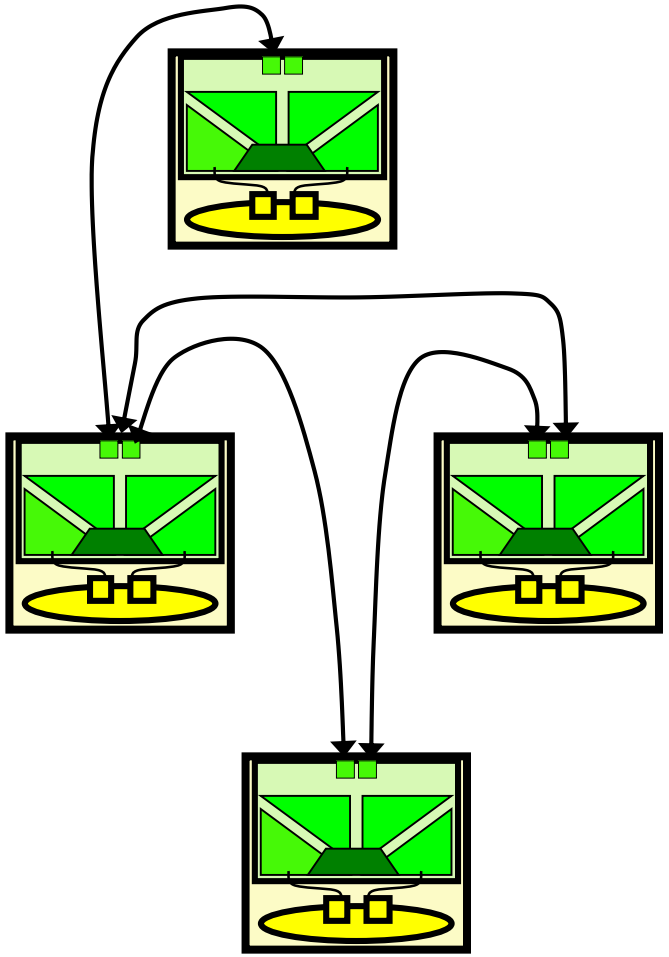
Az autonóm egység

Általánosított „ágens”



AE: Kölcsönhatások

- Kapcsolatok AE-k között:
  - Dinamikus, ideiglenes, célorientált
  - Szabályok és kényszerek definiálják
  - Egyezség által jön létre
    - Ez lehet tárgyalás eredménye
  - Teljes spektrum
    - Peer-to-peer
    - Hierarchikus
  - Házirendek (policy) szabályozhatják



Önszervezés

- Az önszervezés
  - alacsony szintű egységekben végrehajtott
  - dinamikus folyamatok összessége, amely során
  - struktúra vagy rend jelenik meg
  - globális szinten.

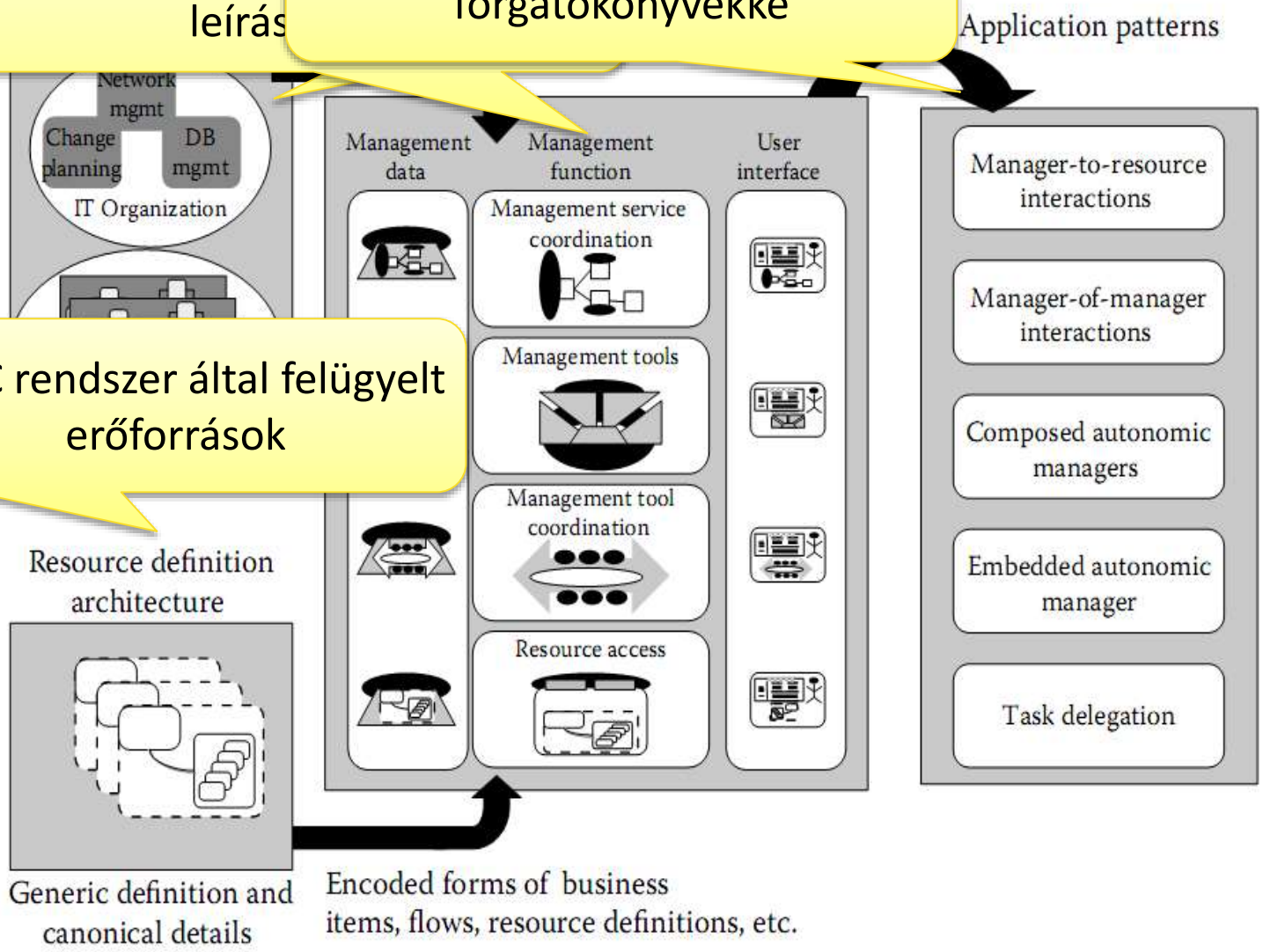
- Az önszervező viselkedést eredményező szabályokat (amelyek a kölcsönhatásokat meghatározzák) az **AE-k csupán lokális információ alapján alkalmazzák**

AC referencia architektúra

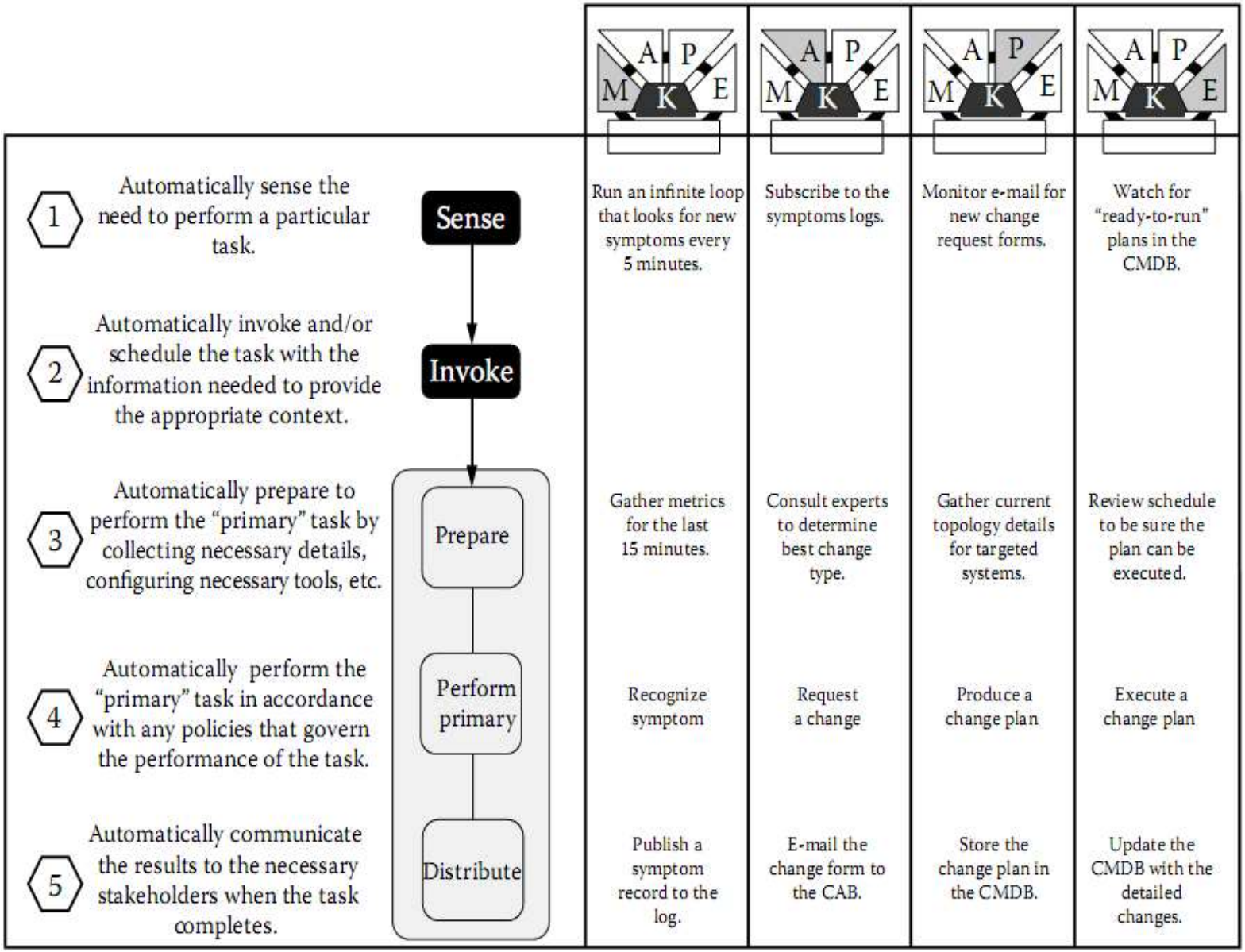
IT építőelemek, és ö
leírás

Építőelemek kombinálása tipikus forgatókönyvekké

Az AC rendszer által felügyelt erőforrások

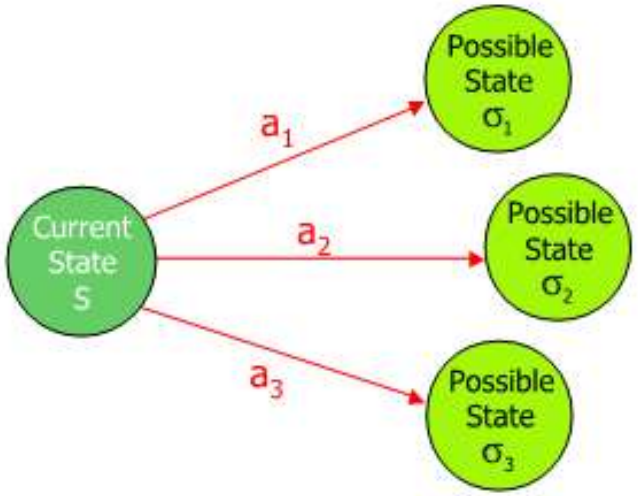


Kölcsönhatások



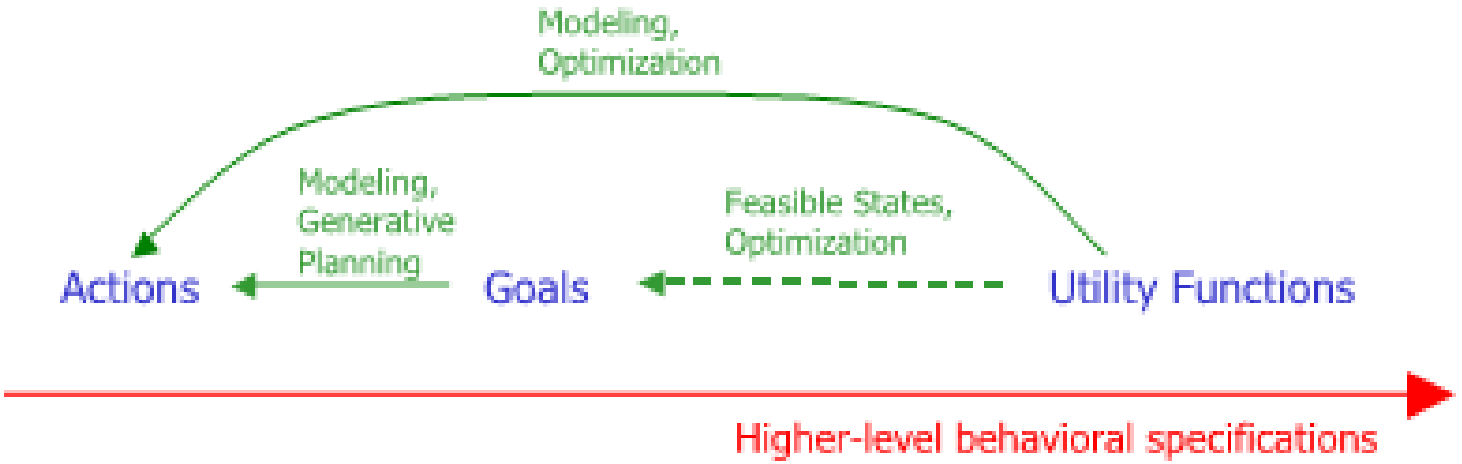
Kitekintés: AC és MI [3]

- Policy (~szabály, házirend, eljárásrend) alapú tervezés
  - Állapot alapú
- Action
  - ECA (~üzleti szabály)



- Goal
  - „Célállapot”; a rendszer dönt (pl. heurisztika)
- Utility function (hasznosság)
  - Minden állapotnak „érték”; nem bináris hasznosság
  - Rugalmasabb működés, nehezebb specifikáció

Eljárásrendek: specifikációs szintek

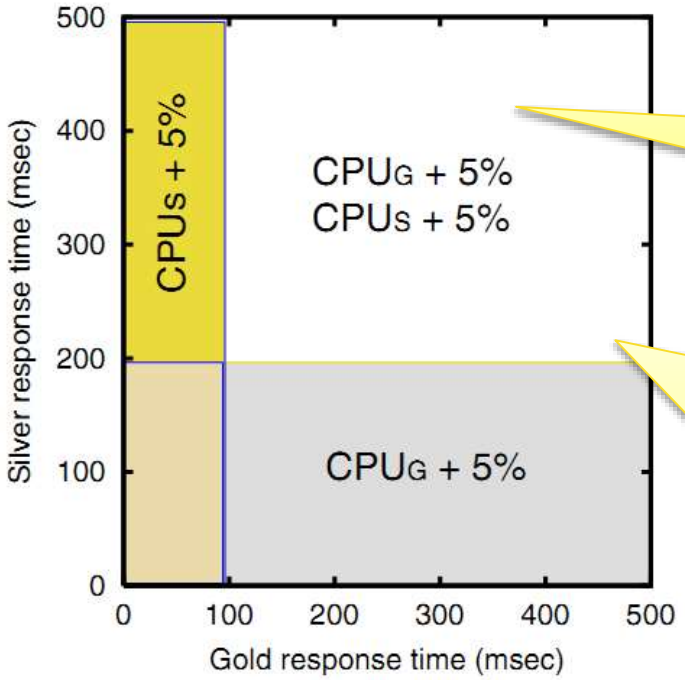


Példa: Action policy

- „Gold” és „Silver” tranzakciók egy adatközpontban
- Policy ütközés, „vergődés”

G. IF ($RT_G > 100$ msec) THEN (Increase CPU_G by 5%)

S. IF ($RT_S > 200$ msec) THEN (Increase CPU_S by 5%)



Mi lesz az osztott erőforrásokkal?

Megoldás: pl. a priori tudás bevitele
(pl. Gold fontosabb, mint Silver,
bizonyos szint fölött nem kérünk plusz CPU-t,
másik szerverre allokáljuk a terhelést, …)

Példa: Goal policy

- Ugyanaz az adat
- „Vágyott”+elérhető tartományok

Pontosabban: M/M/1-ből
és a Little-törvényből adódik

$$T_i = \frac{1}{\alpha C_i - \lambda_i}$$

T: adott tranzakcióosztály válaszideje
C: erőforrás
α: kapcsolat a CPU és a válaszidő közt
λ: érkezési ráta
(egyszerű sorbanállási modell alapján)

DEMO Let’s plot this ourselves

```
t_class <- function(x){1/(x-1)}

goldandsilver <- function(goldperc, sumserver){
  goldrpt <- t_class(sumserver*goldperc)
  silverrpt <- t_class(sumserver*(1-goldperc))
  c(goldrpt,silverrpt)
}


res <- sapply(seq(from=0.3, to=0.7, by=0.005), goldandsilver, 5)

response_times <- data.frame(t(res))
response_times$servers <- 5

res2 <- sapply(seq(from=0.15, to=0.85, by=0.005), goldandsilver, 10)
response_times2 <- data.frame(t(res2))
response_times2$servers <- 10

res3 <- sapply(seq(from=0.08, to=0.92, by=0.005), goldandsilver, 20)
response_times3 <- data.frame(t(res3))
response_times3$servers <- 20

res4 <- rbind(response_times, response_times2, response_times3)

res4$SNUM <- as.factor(res4$servers)
res4$servers <- NULL
colnames(res4) <- c("gold_rpt", "silver_rpt", "servnum")

library(ggplot2)
qplot(data = res4, x=gold_rpt, y=silver_rpt, color=servnum) + aes(size=0.5)
```

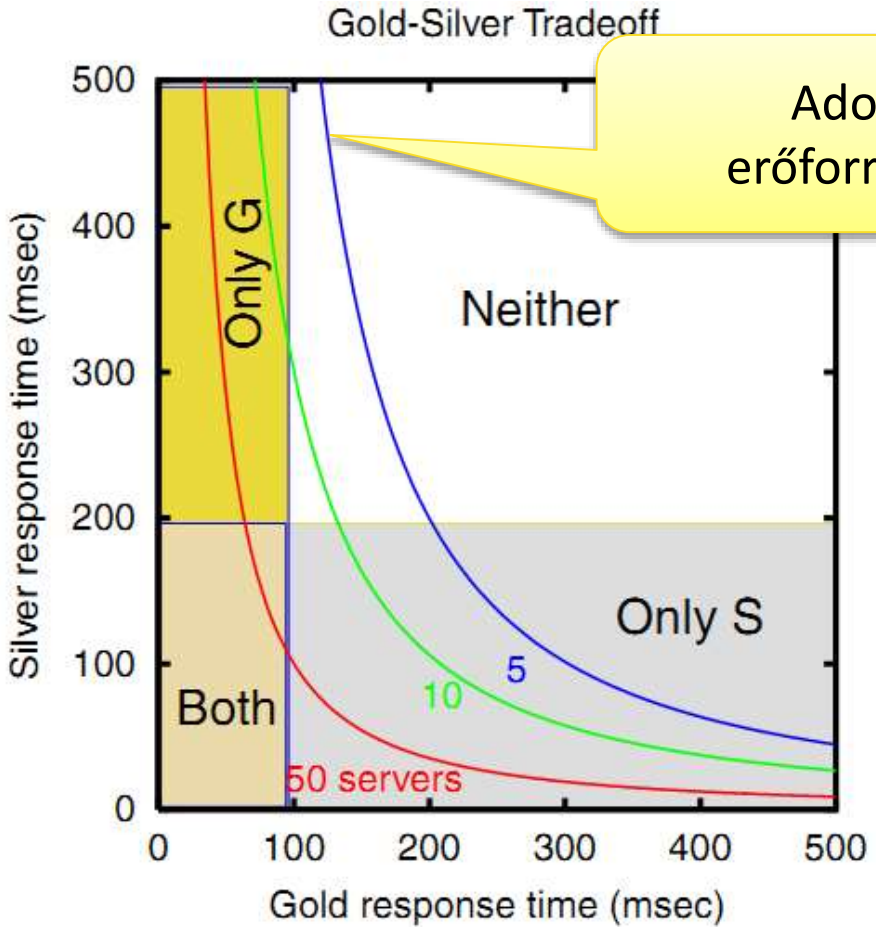
Példa: Goal policy

- Ugyanaz az adatközpont, cél:

G. $RT_G \leq 100msec$

S. $RT_S \leq 200msec$

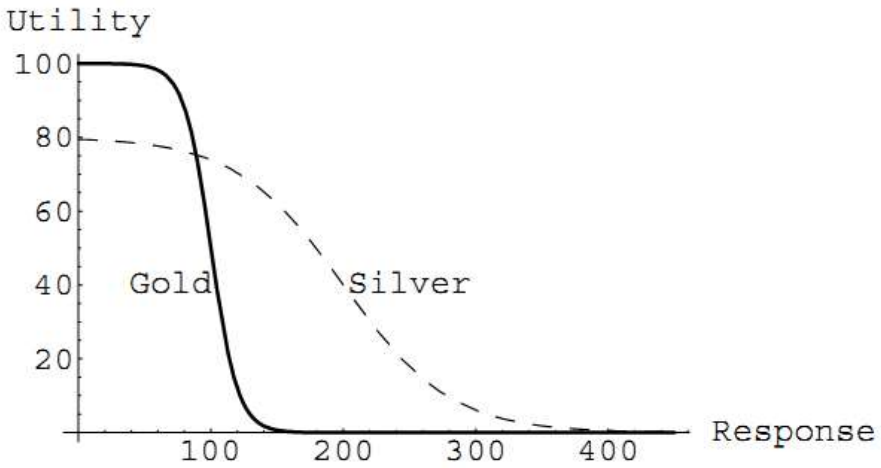
- „Vágyott”+elérhető tartományok

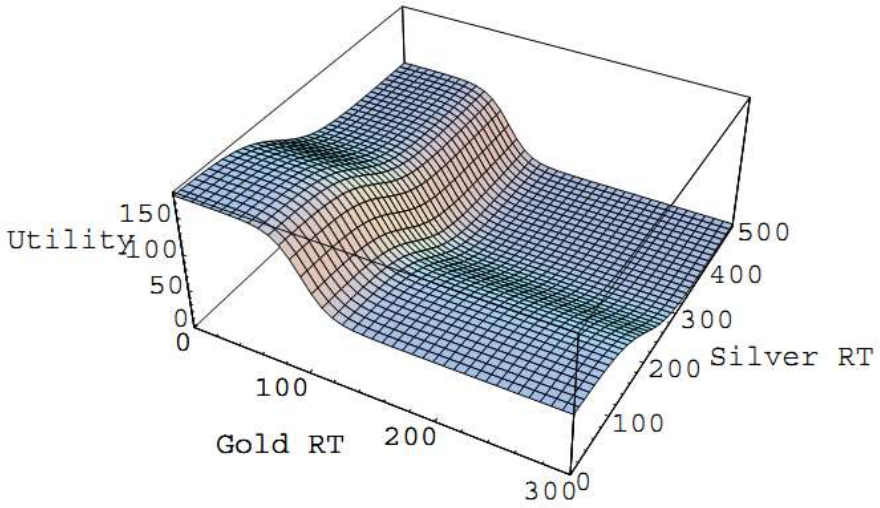


Adott terhelés és erőforráskészlet mellett

$$T_i = \frac{1}{\alpha C_i - \lambda_i}$$

Példa: hasznosság alapú policy





- Pl. SLA alapján
- Vezérelhet cél alapú policyt, pl. erőforrás menedzser szintjén
  - Egyszerű specifikáció, komplex döntési logika

Kihívások, feltételezések

- A hasznosság előre ismert
  - Rossz specifikáció: Silver osztály „éhezik”
  - Nincsenek kiugróan fontos/hosszú tranzakciók
- Taszkváltás hatása elhanyagolható
- Válaszidő egyértelműen mérhető
  - Átlag? Max?
- Az erőforrásmenedzsment hatékony
  - Nem ront a helyzeten az átkonfigurálás

Példa: tanulságok

- Eredmény:
  - Hihetően működő
  - automatikus
  - (valamennyire … erősen) deklaratív
  - újrakonfigurációs logika
  - ami SLA-k sértése ellen véd
  - (persze nem tökéletes)

- Figyeljük meg: matematikai apparátus…

Autonóm rendszerek összehasonlítása

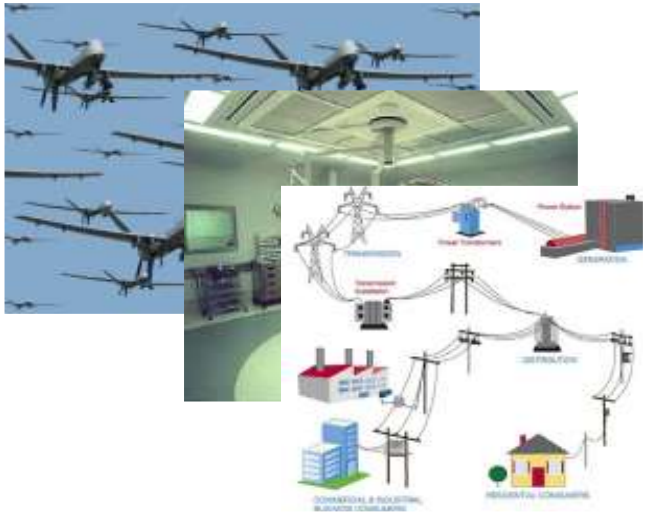
- QoS
- Költség
- Rugalmasság/Granularitás
- Autonómia foka
- Adaptivitás
- Reakcióidő
- Érzékenység
- Stabilitás

Self-optimization példa: kooperatív adatközpont-optimalizálás

Imre Kocsis, Zoltán Ádám Mann, Dávid Zilahi,

“Optimal deployment for critical applications in Infrastructure as a Service” (ICACON 2015) alapján

Emerging cloud applications: NFV, CC and CPS



Network Function Virtualization

Carrier Cloud

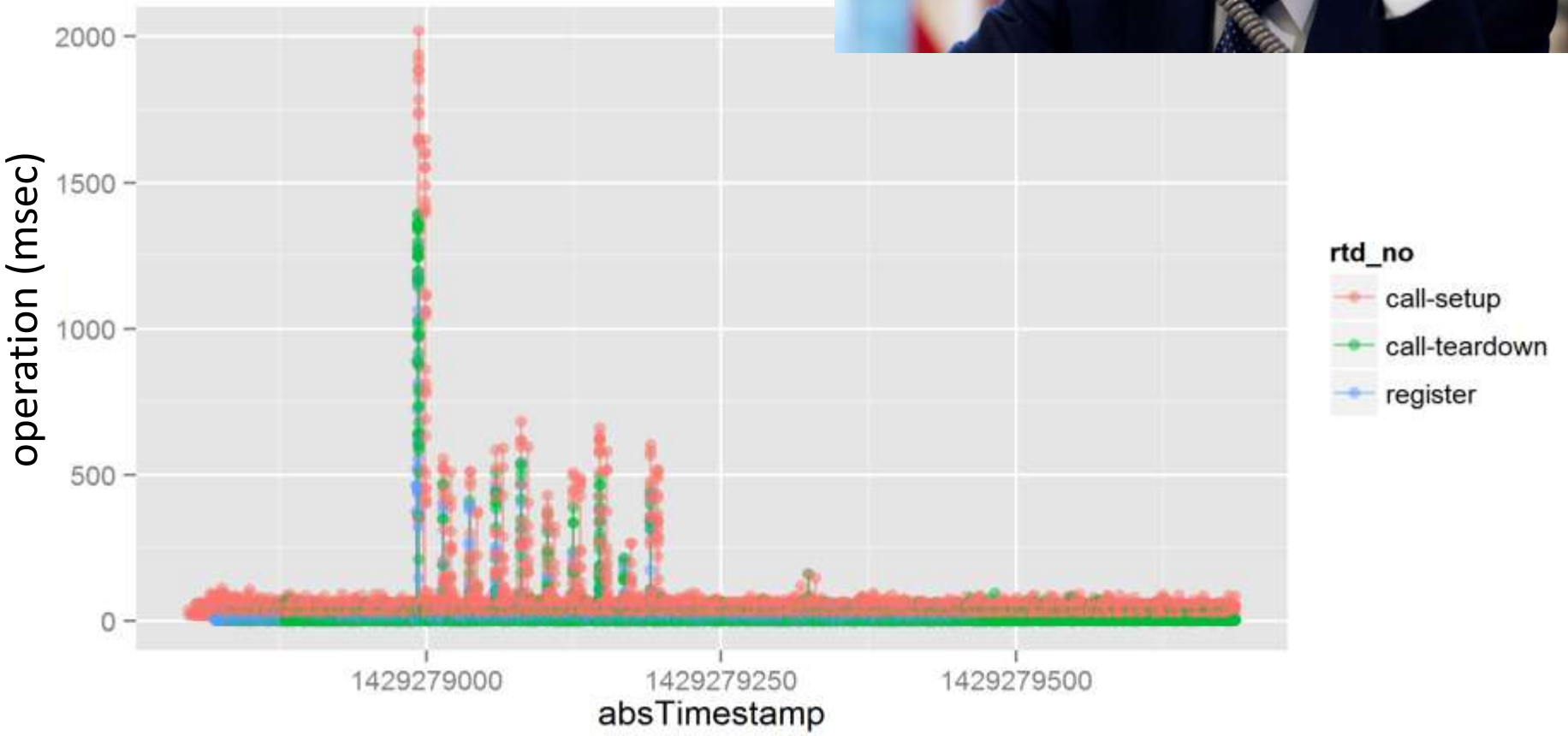
Cyber-Physical Systems

Instead of dedicated resources: IaaS!

What's the problem?

Noisy neighbor VM (or operator scheduling)

Call setup time ☹

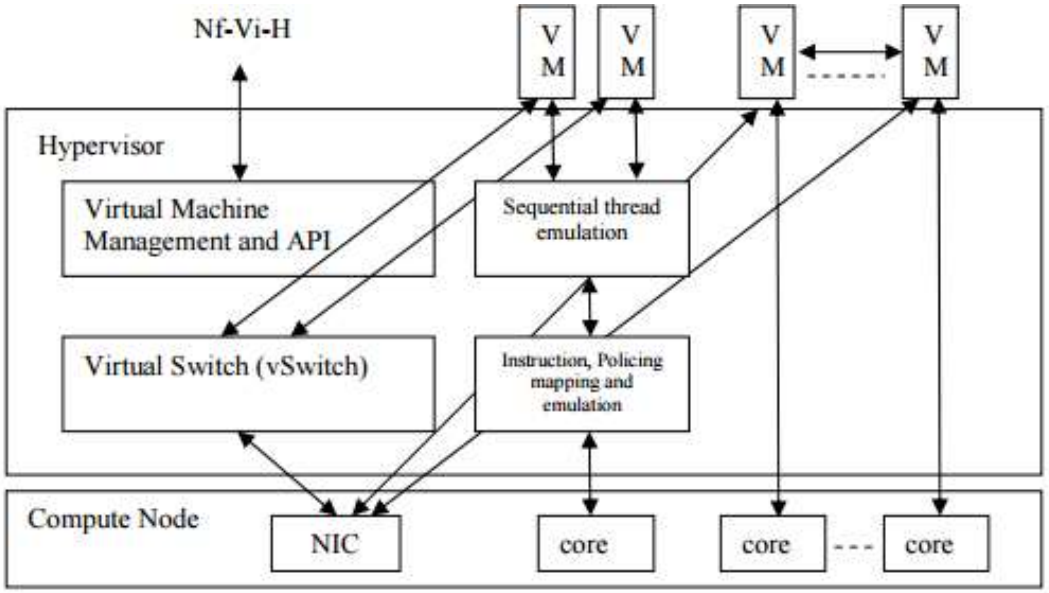


We have the toolbox…

- Example: CPU
- Limits, reserves, shares, schedulers…
- Core affinity, „pinning”
- Dedicated: still option!

- No access from cloud
  - ***Yet!*** (coming)
- Standards ***mandate*** the capability

ETSI NFV Architectural Framework



NFV as a model of cloud-CPS

- Critical service
- Interactive/real-time
- No tolerance for network impairments
  - w/o additional dependability mechanisms

General-purpose IaaS for critical applications?

- Performance isolation
  - Sub-par for the purpose
  - Uncontrollable channels
  - No/not disclosed
- Performance: instability and heterogeneity
- Dependability: in application

- For the operator: DC density is king
  - For on-line capacity – **HVAC**!
  - Heavy optimization – of **op. cost**

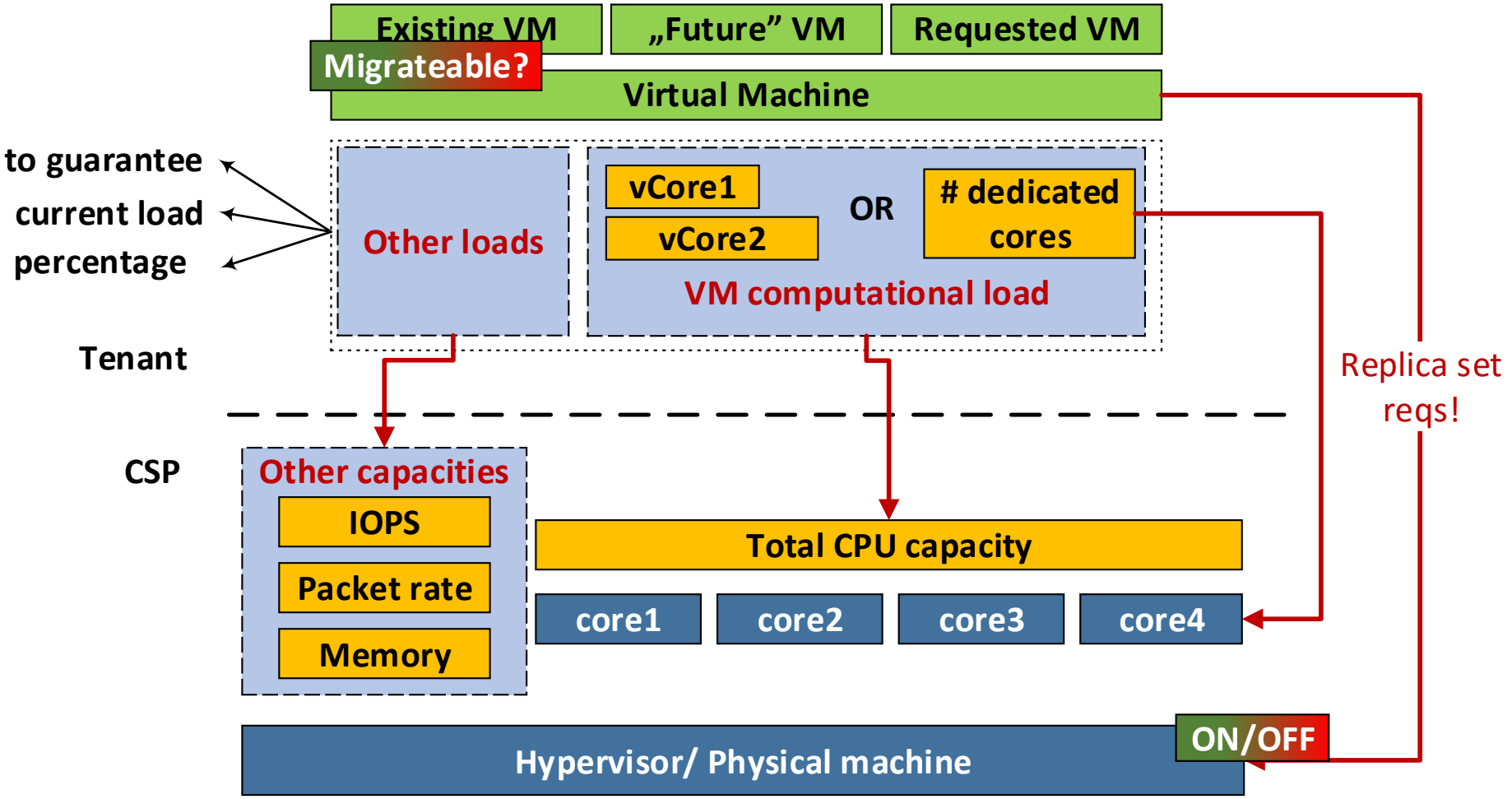
Our goal

Deployment optimization

that complies with

tenant deployment policies

Deployment modeling approach



Objectives

- **OPEX** - amount of switched-on machines
- **RESERVES** - immediately servable future VM reqs
- **QoE** - single-tenant impact of physical fault

Optimization model

Basic packing formulation

- VMs: v_i $(i = 1, \ldots, n)$
- PMs: p_j $(j = 1, \ldots, m)$
- Resource types: $r = memory, egressPacketRate, ingressPacketRate, IOPS$
- Binary variables:

$$Alloc_{i,j} = \begin{cases} 1 & \text{if } v_i \text{ is allocated on } p_j \\ 0 & \text{otherwise} \end{cases}$$

$$Active_j = \begin{cases} 1 & \text{if } p_j \text{ is active} \\ 0 & \text{otherwise} \end{cases}$$

Basic packing formulation

$$\text{minimize} \quad \sum_{j=1}^{m} Active_j$$

$$\text{subject to} \quad \sum_{j=1}^{m} Alloc_{i,j} = 1, \qquad \forall i$$

$$Alloc_{i,j} \leq Active_j, \qquad \forall i,\ \forall j$$

$$\sum_{i=1}^{n} vload(v_i, r) \cdot Alloc_{i,j} \leq cap(p_j, r), \qquad \forall j,\ \forall r$$

$$Alloc_{i,j} \in \{0,1\}, Active_j \in \{0,1\}, \qquad \forall i,\ \forall j$$

CPU constraints

- PM p_j has $pcn(p_j)$ pCPUs with $pcc(p_j)$ computational capacity per pCPU
- VM v_i has $vcn(v_i)$ vCPUs with $vcc(v_i)$ computational capacity requested per vCPU
- V_{ded}: set of VMs with dedicated CPU cores
- Additional constraints:

$$vcc(v_i) \cdot Alloc_{i,j} \leq pcc(p_j), \qquad \forall i,\ \forall j$$

$$\sum_{v_i \in V_{ded}} vcn(v_i) \cdot Alloc_{i,j} \leq pcn(p_j), \qquad \forall j$$

$$\sum_{i=1}^{n} cpu\_load(i,j) \cdot Alloc_{i,j} \leq pcn(p_j) \cdot pcc(p_j), \qquad \forall j$$

- where

$$cpu\_load(i,j) = \begin{cases} vcn(v_i) \cdot pcc(p_j) & \text{if } v_i \in V_{ded} \\ \sum_{vc \in vC(v_i)} vcl(v_i, vc) & \text{if } v_i \notin V_{ded} \end{cases}$$

Additional constraints for critical VMs

- If v_i must not be migrated and it is currently mapped to p_j:

$$Alloc_{i,j} = 1$$

- If at most k of the VMs in $\overline{V}$ can be on the same PM:

$$\sum_{v_i \in \overline{V}} Alloc_{i,j} \leq k \qquad \forall j$$

Additional optimization objectives

Best-effort allocation of reservations for future VM requests:

- Additional constraint:

$$\sum_{j=1}^{m} Alloc_{i,j} \leq 1 \qquad \forall v_i \in V'$$

- Additional objective:

$$\text{maximize} \quad \sum_{v_i \in V'} \sum_{j=1}^{m} Alloc_{i,j}$$

Additional optimization objectives

Minimizing the impact of the failure of a PM on a tenant:

- Additional variable: C (highest number of VMs of the same tenant on the same PM)
- Additional constraint:

$$\sum_{v_i \in VT(t)} Alloc_{i,j} \leq C \quad \forall t \in T, \forall j$$

- Additional objective:

$$\text{minimize} \quad C$$

The resulting cost function

$$\text{minimize} \quad \alpha \cdot \sum_{j=1}^{m} Active_j - \beta \cdot \sum_{v_i \in V'} \sum_{j=1}^{m} Alloc_{i,j} + \gamma \cdot C$$

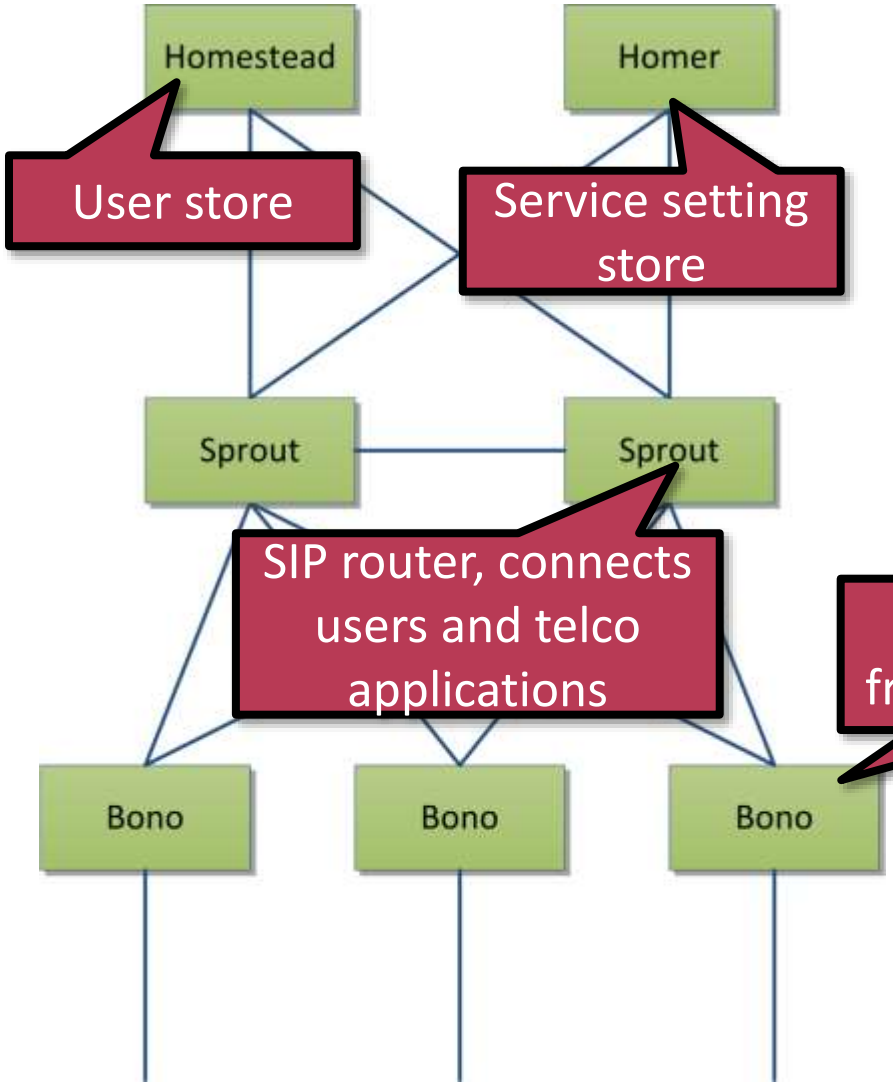
Number of active PMs

Number of fulfillable future requests

Max. number of VMs of the same tenant on the same PM

Case study

Case study: Clearwater



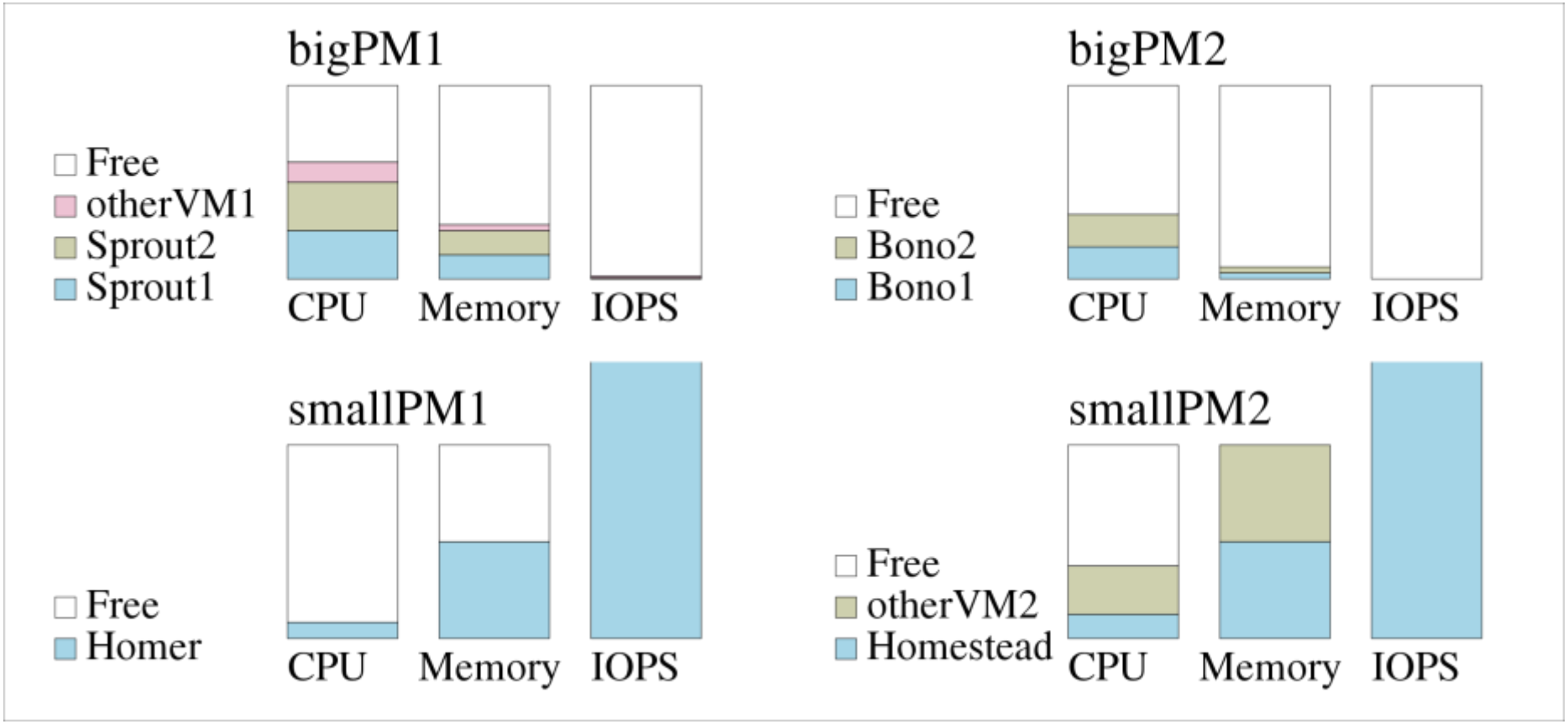
- Open-source implementation of the IMS (IP Multimedia Subsystem) standard
- Considered to be deployed in NFV IaaS

Case study: Clearwater

PM resources	Cores	Core capacity	Memory	IOPS	Number
bigPM	6	8	32	10000	2
smallPM	4	6	8	350	2

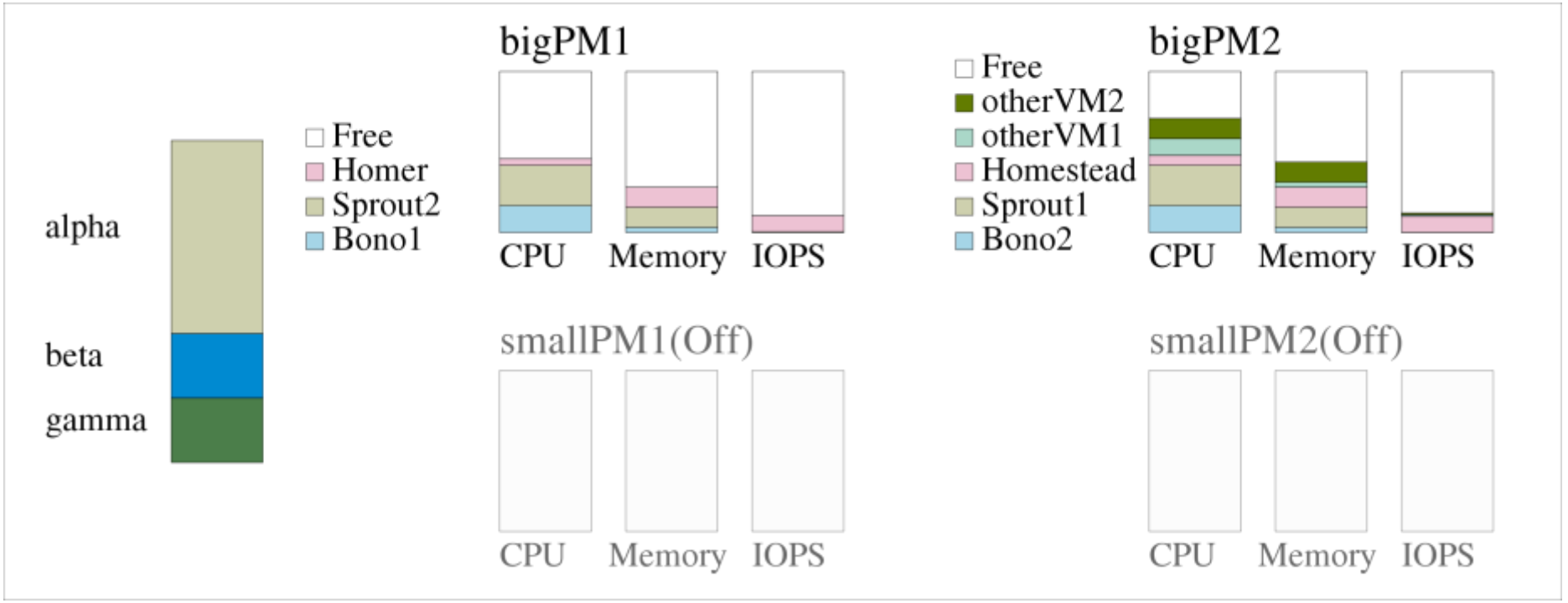
VM type	Cores / dedicated	Capacity / guaranteed	Memory / guaranteed	IOPS / guaranteed	Number / reservation
Bono	1 / y	4 / y	1 / y	100 / n	2 / 1
Sprout	2 / n	6 / y	4 / y	100 / n	2 / 0
Homer / Homestead	1 / n	4 / n	4 / y	1000 / y	1 / 0
other1	2 / n	4 / n	4 / n	100 / n	1 / 0
other2	4 / n	2 / n	4 / n	50 / n	1 / 0

Shortfalls of a manual deployment



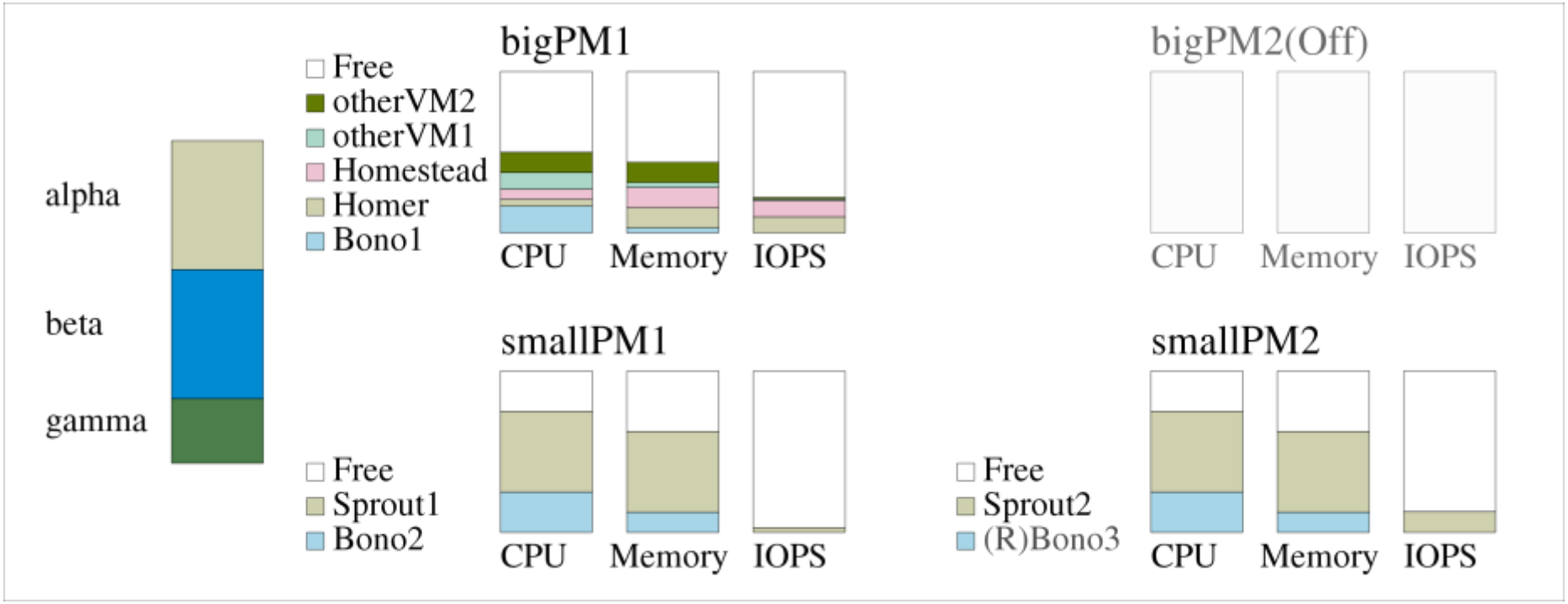
- × All PMs are on → high energy consumption
- × Some PMs overloaded, others lightly utilized
- × All instances of a VM group on the same PM

Automated solution – cost-sensitive



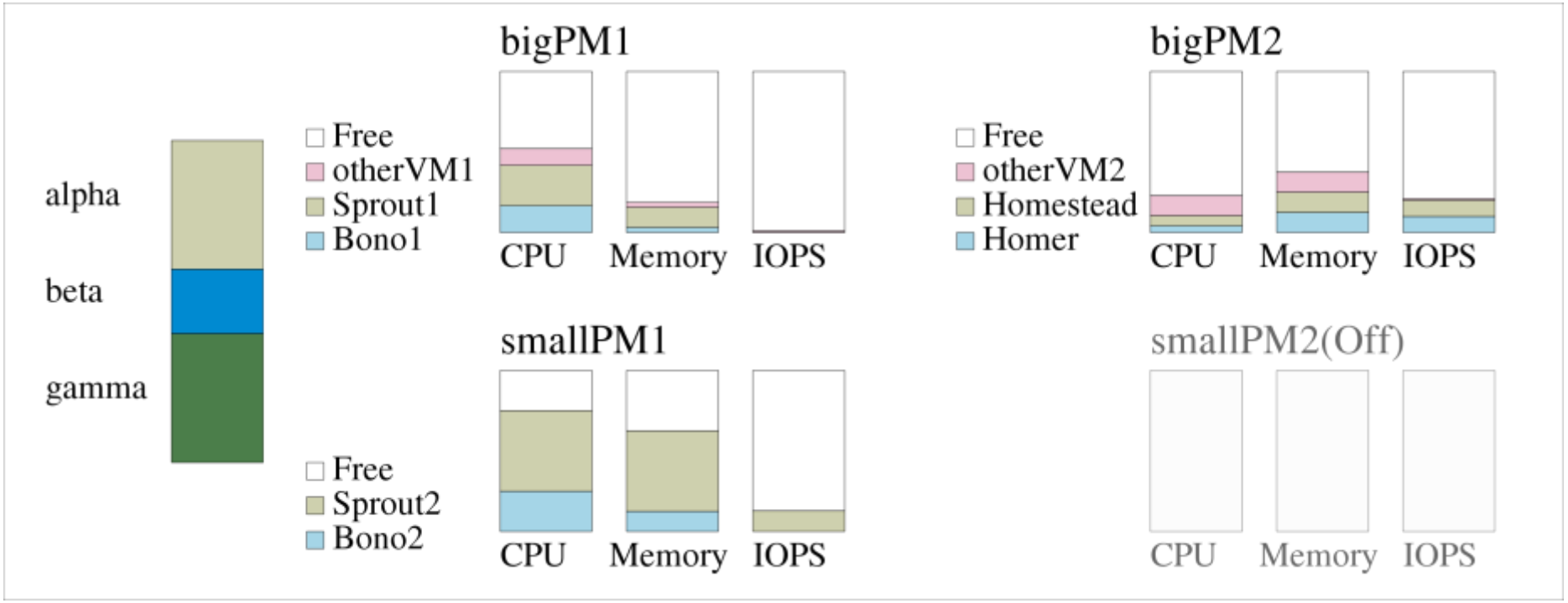
- ✓ Two PMs turned off → significant reduction in energy consumption
- ✓ Instances of a VM group are on different PMs
- × Limited reserve for future VM requests
- × Failure of a PM would have significant impact on a tenant

Automated solution – reserve-oriented



- ✓ One PM turned off → some reduction in energy consumption
- ✓ Instances of a VM group are on different PMs
- ✓ Indicated reserve available for future VM requests
- × Failure of a PM would have significant impact on a tenant

Automated solution – balancing fault impact



- ✓ One PM turned off → some reduction in energy consumption
- ✓ Instances of a VM group are on different PMs
- × Limited reserve for future VM requests
- ✓ Failure of a PM would lead to the loss of only 2 VMs of a tenant, not 3

Initial scalability assessment