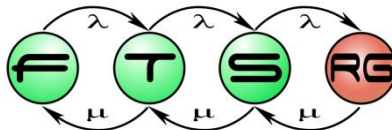


Felhő számítástechnika 2.

A kiberfizikai rendszerek (VIMIMA02) tárgyban

Kocsis Imre, ikocsis@mit.bme.hu

2017. ősz



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/>

*„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.”*

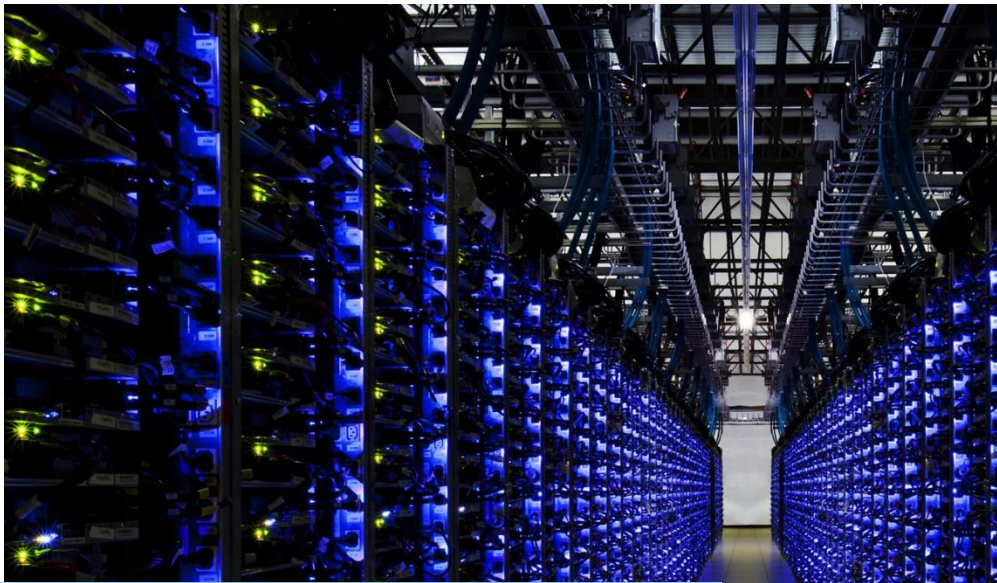
Cloud „telepítési” (*deployment*) modellek [5]

- 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...



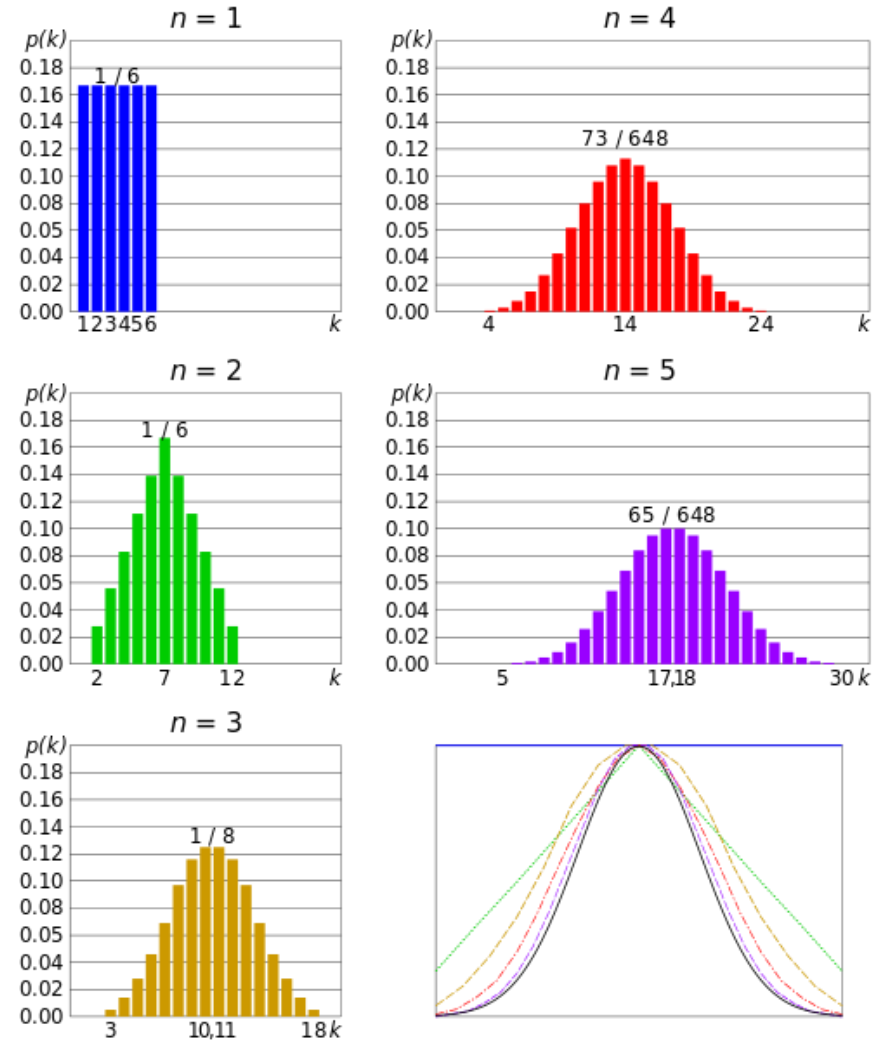
Miért éri meg a szolgáltatónak? [6]

- (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

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

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]

Basic cloud market forces – examples by simulation

- 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

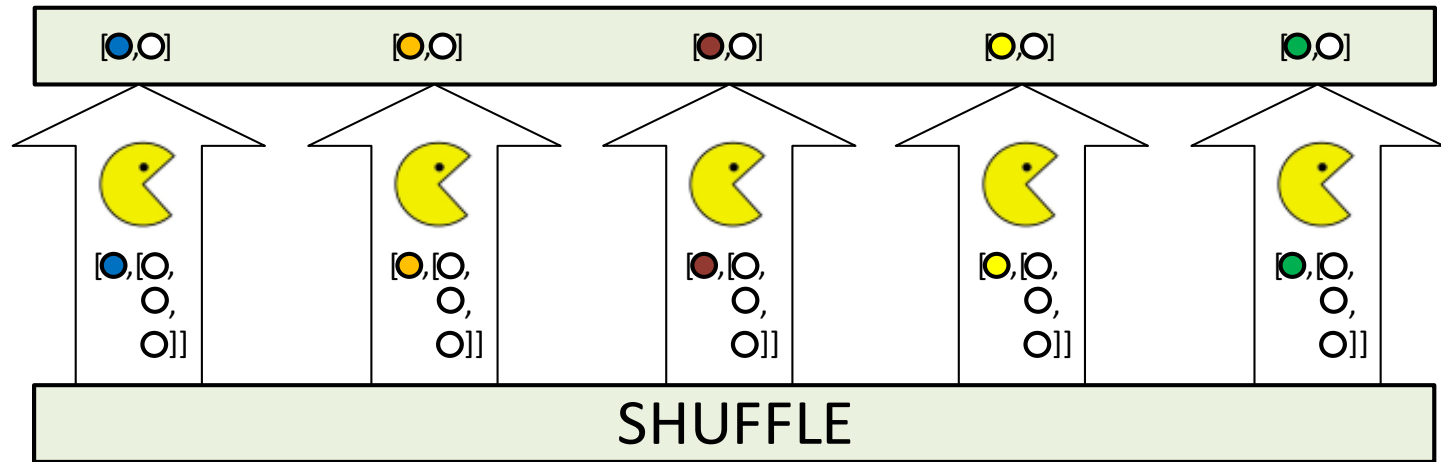
„Ingyen idő” / „ingyen gyorsítás”

Párhuzamosítható terhelések

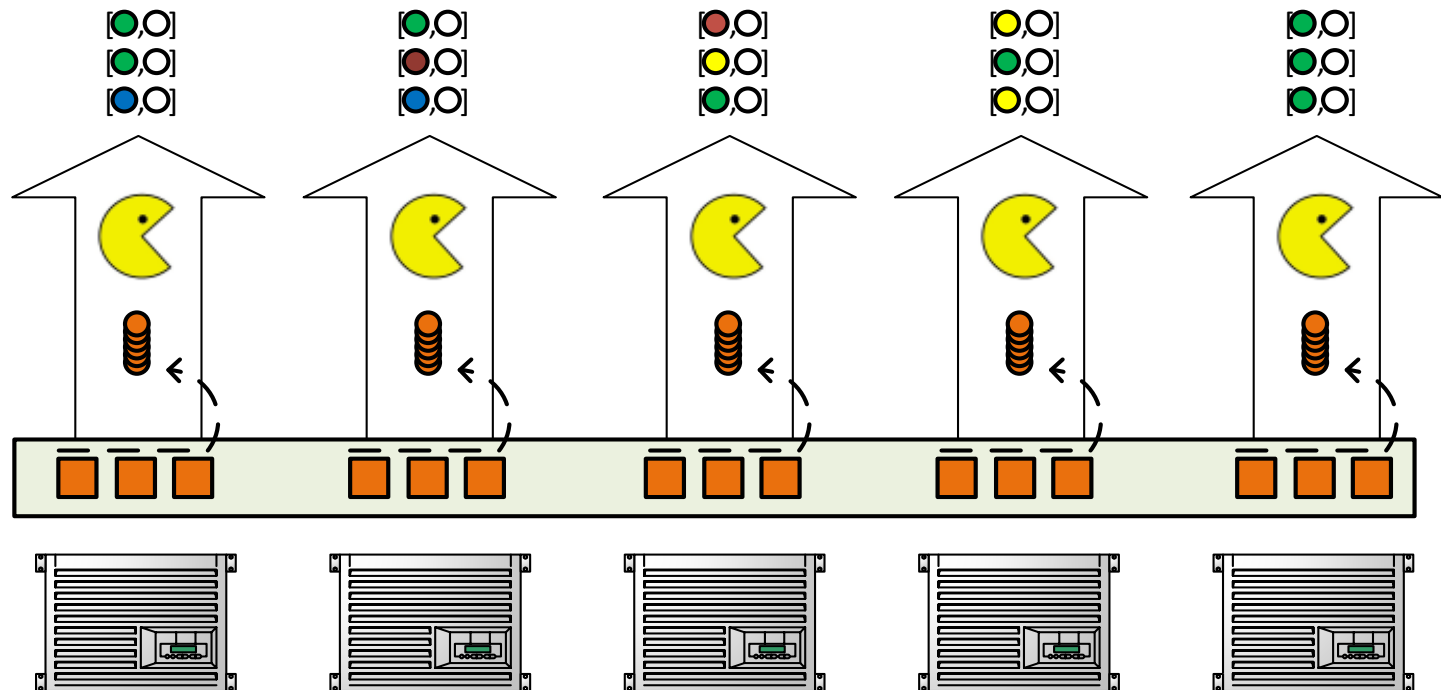
- Egyre több „zavarbaejtően” párhuzamos (*embarrassingly parallel*), „scale-out” alkalmazáskategóriánk van
- NYT TimesMachine [12]: public domain archív
 - Konverzió web-barát formátumra [13]: Hadoop, pár száz VM, 36 óra
- **A használat alapú számlázás miatt ~ugyanannyiba kerül, mint egy VM-mel**
- Praktikusan: „ingyen idő”

MapReduce (Hadoop)

„Reduce”



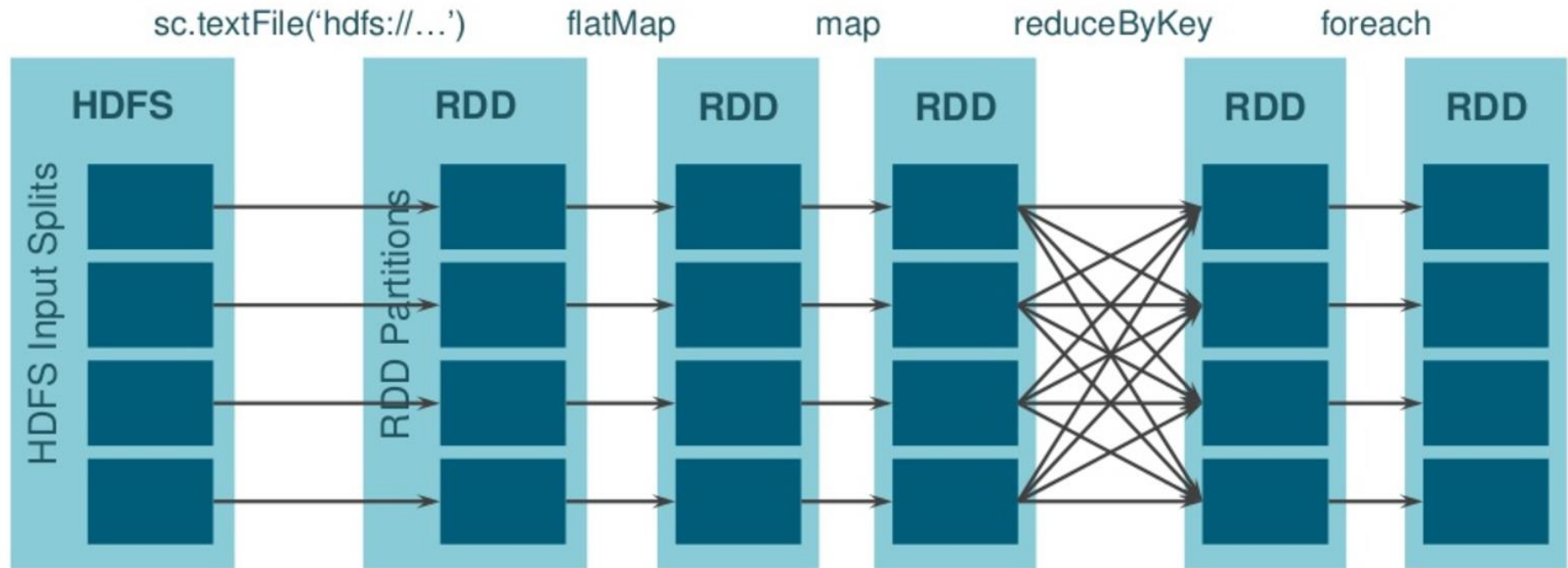
„Map”



Distributed
File System

Apache Spark

WordCount example



<https://www.slideshare.net/AGrishchenko/apache-spark-architecture>

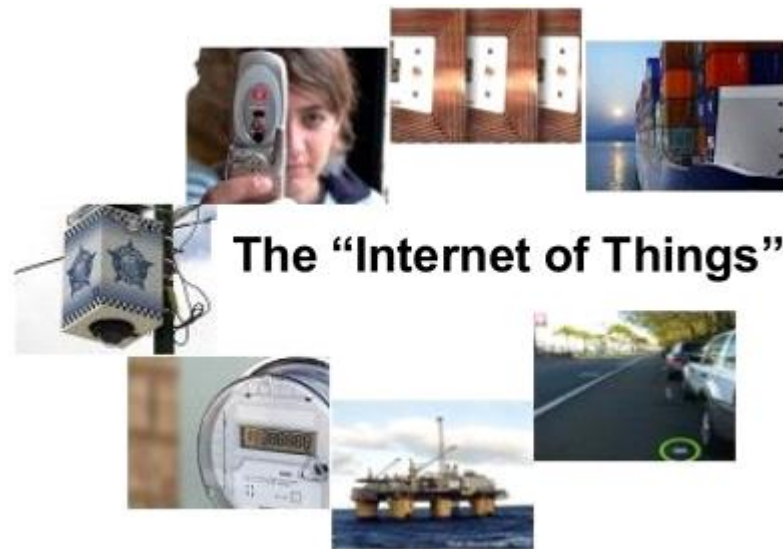
Főbb tanulságok

- Felhőben
 - Látszólag végtelen erőforrások – de scale-out, nem up
 - Kapacitások adaptálhatók, de ezt nekünk kell megtenni
- Jellemző architekturális minták
 - Minta (pattern): ~ gyakran felmerülő problémára “általában jó” válasz
 - “at rest” Big Data egy példa – ezen túl még van jópár
- CPS-ben kiemelt fontosságú:
 - stream processing
 - Pub/sub, illetve MQ integráció
 - Terep <-> edge <-> felhő

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
- ...



Traditional Computing



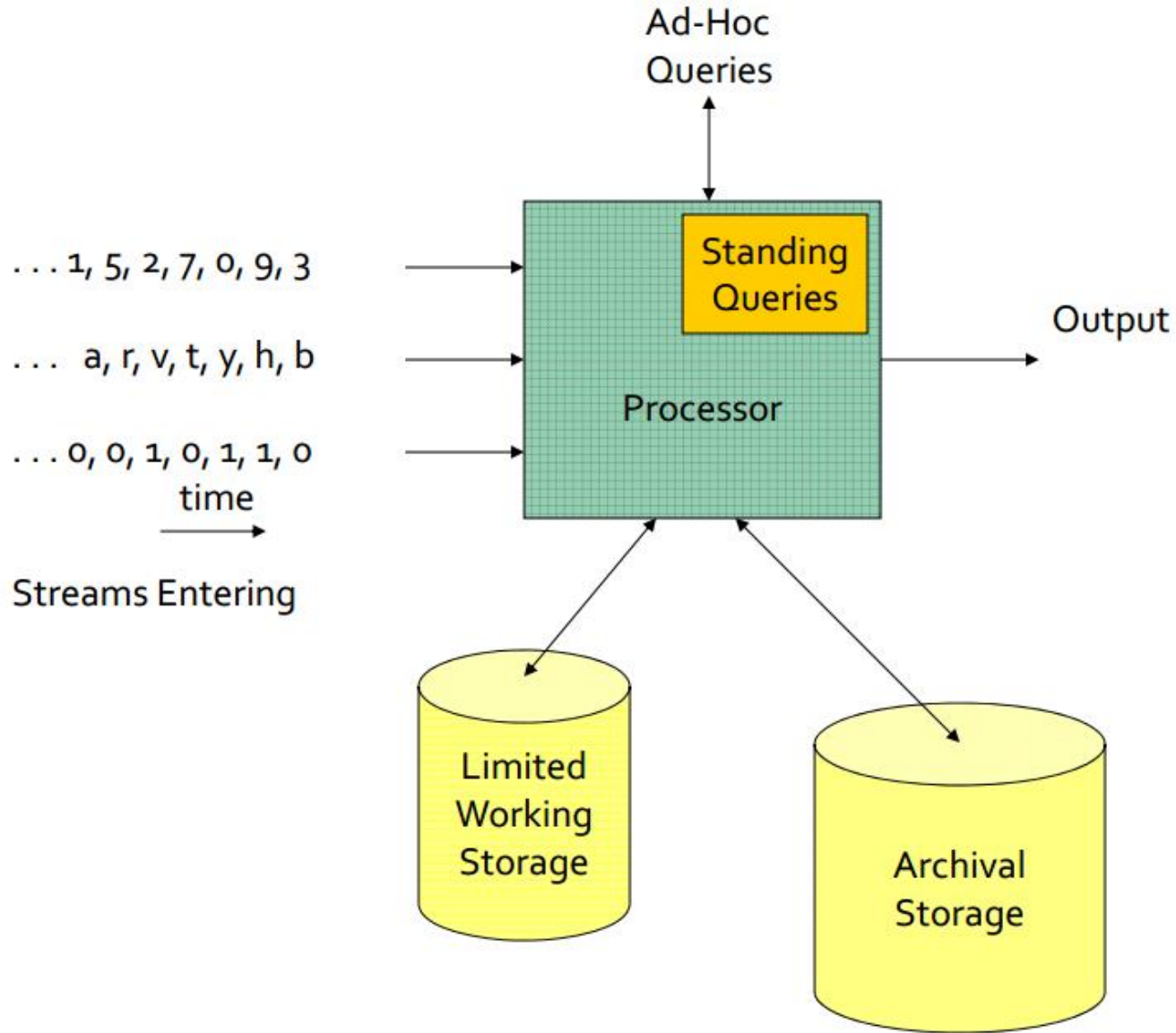
Fact finding with data at rest

Streams Computing

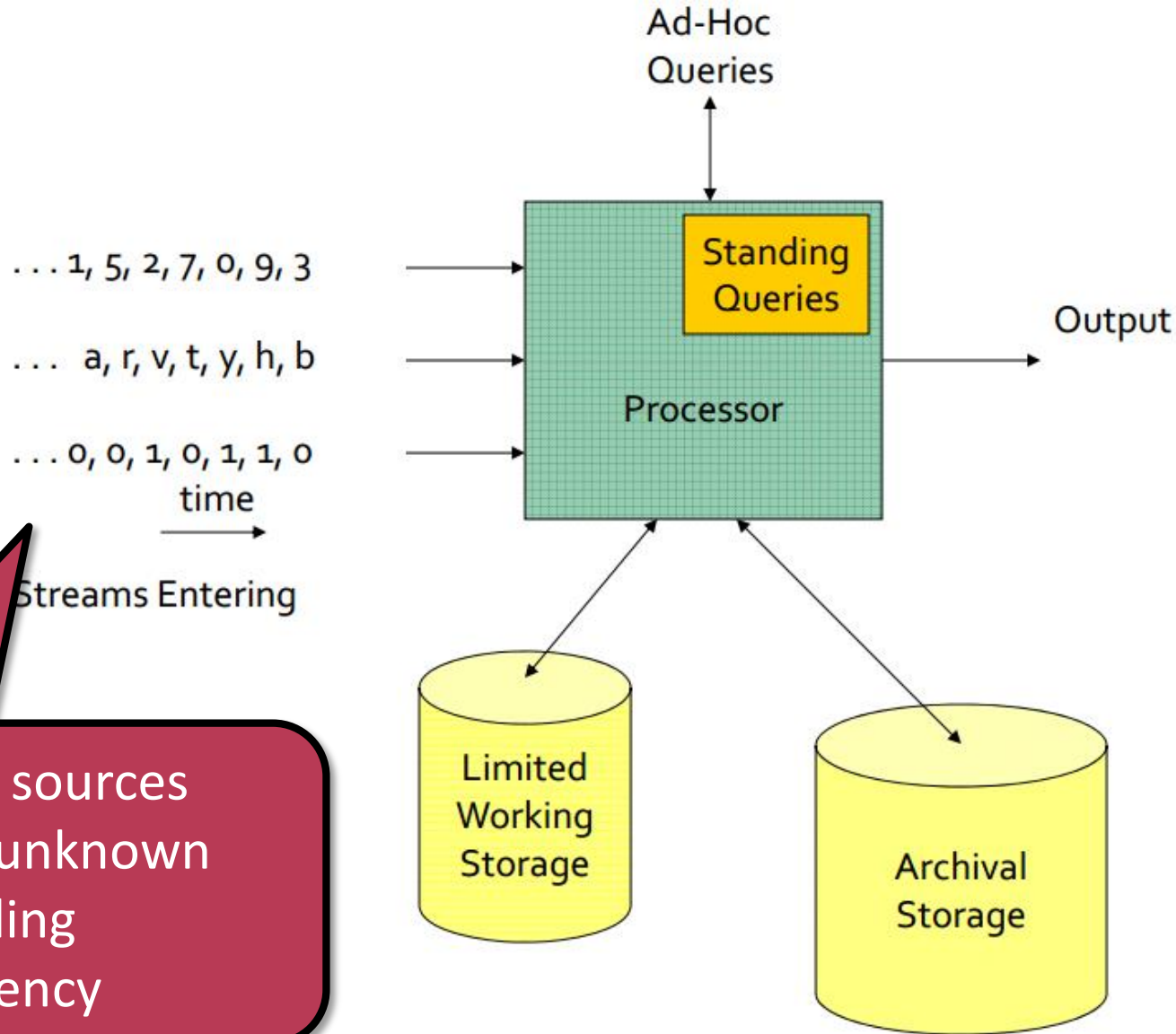


Insights from data in motion

Stream processing (vs „at rest” Big Data)

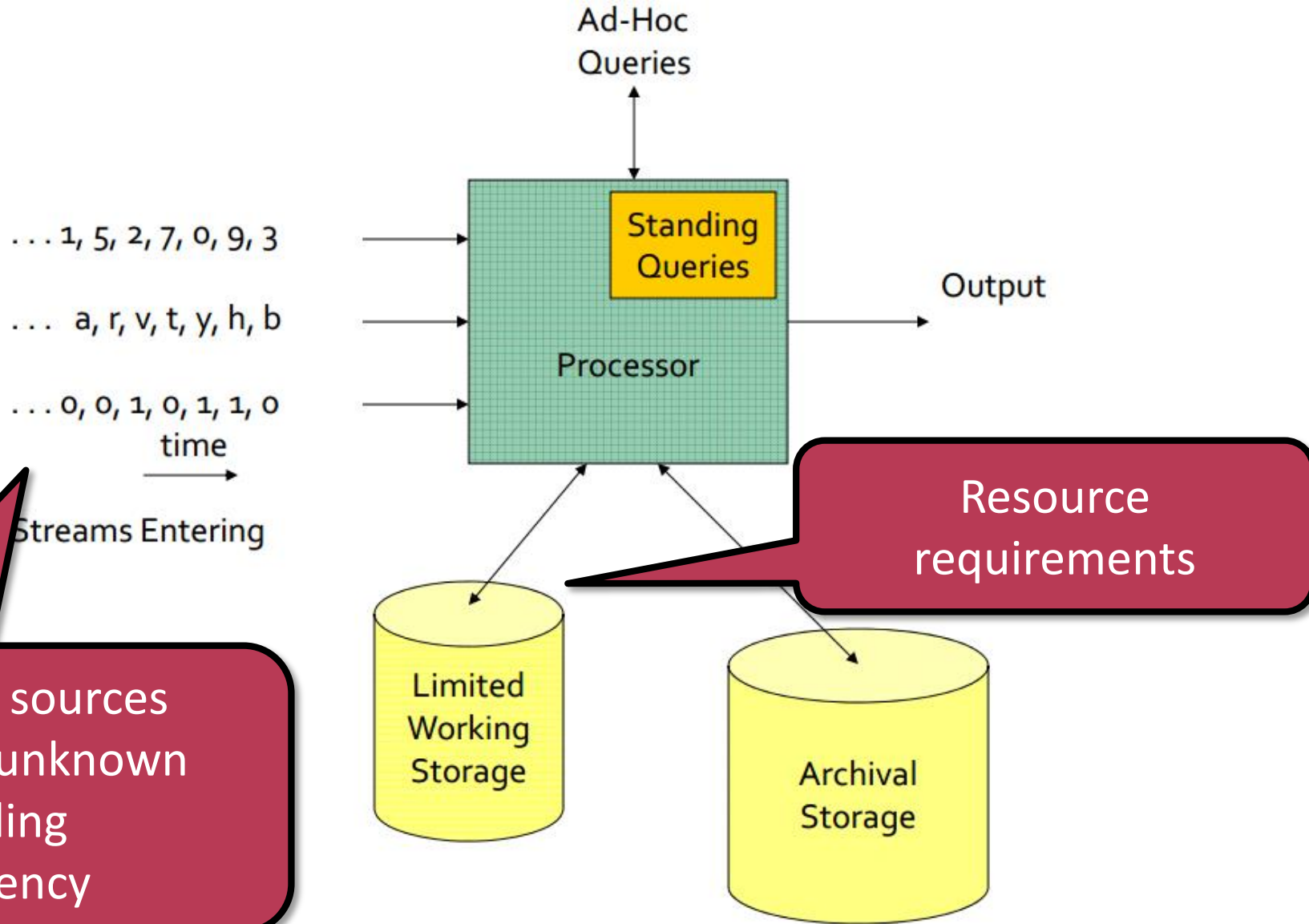


Stream processing



1. Many sources
2. With unknown sampling frequency

Stream processing



Stream processing

Once per stream:
„Local maximum?”

... 1, 5, 2, 7, 0, 9, 3

... a, r, v, t, y, h, b

... 0, 0, 1, 0, 1, 1, 0

time
→

Streams Entering

Ad-Hoc
Queries

Standing
Queries

Processor

Output

Resource
requirements

Limited
Working
Storage

Archival
Storage

1. Many sources
2. With unknown sampling frequency

Stream processing

Once per stream:
„Local maximum?”

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

... 1, 5, 2, 7, 0, 9, 3

... a, r, v, t, y, h, b

... 0, 0, 1, 0, 1, 1, 0

time
→

Streams Entering

Ad-Hoc
Queries

Standing
Queries

Processor

Output

Resource
requirements

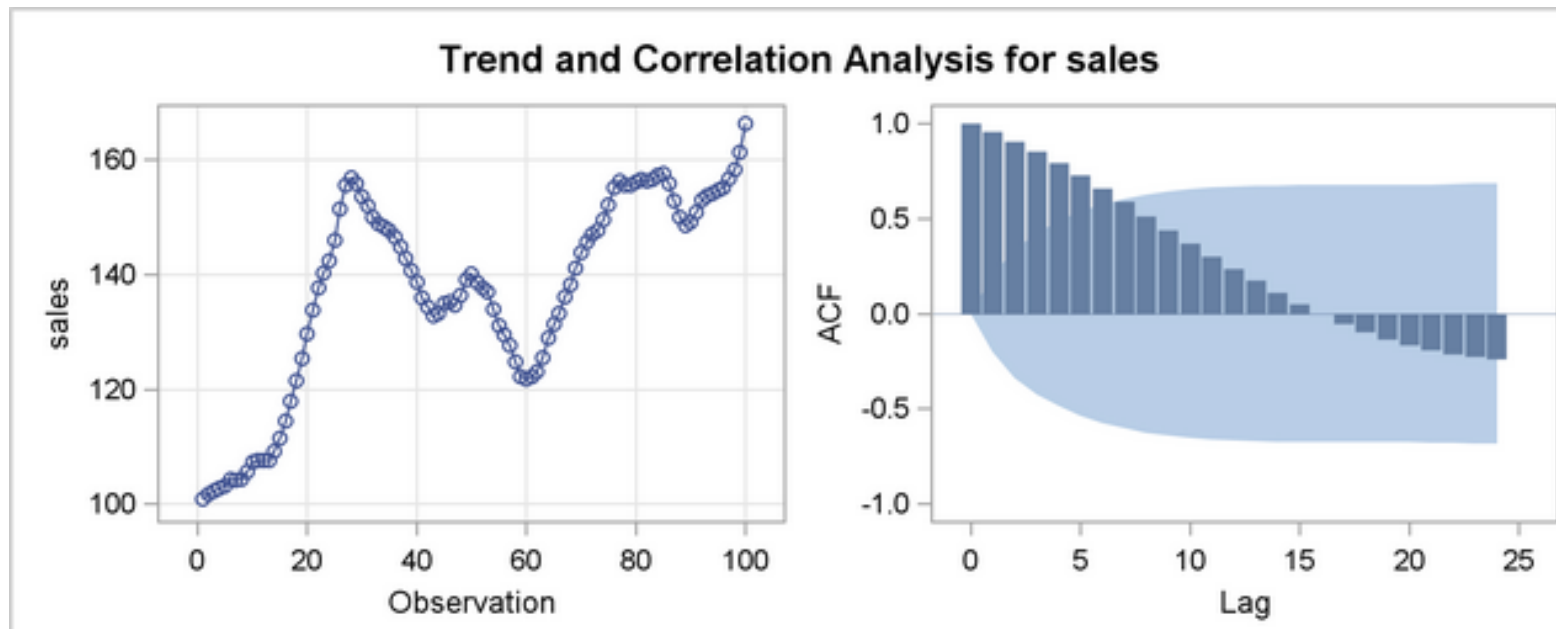
1. Many sources
2. With unknown sampling frequency

Limited
Working
Storage

Archival
Storage

Typically sliding window approaches

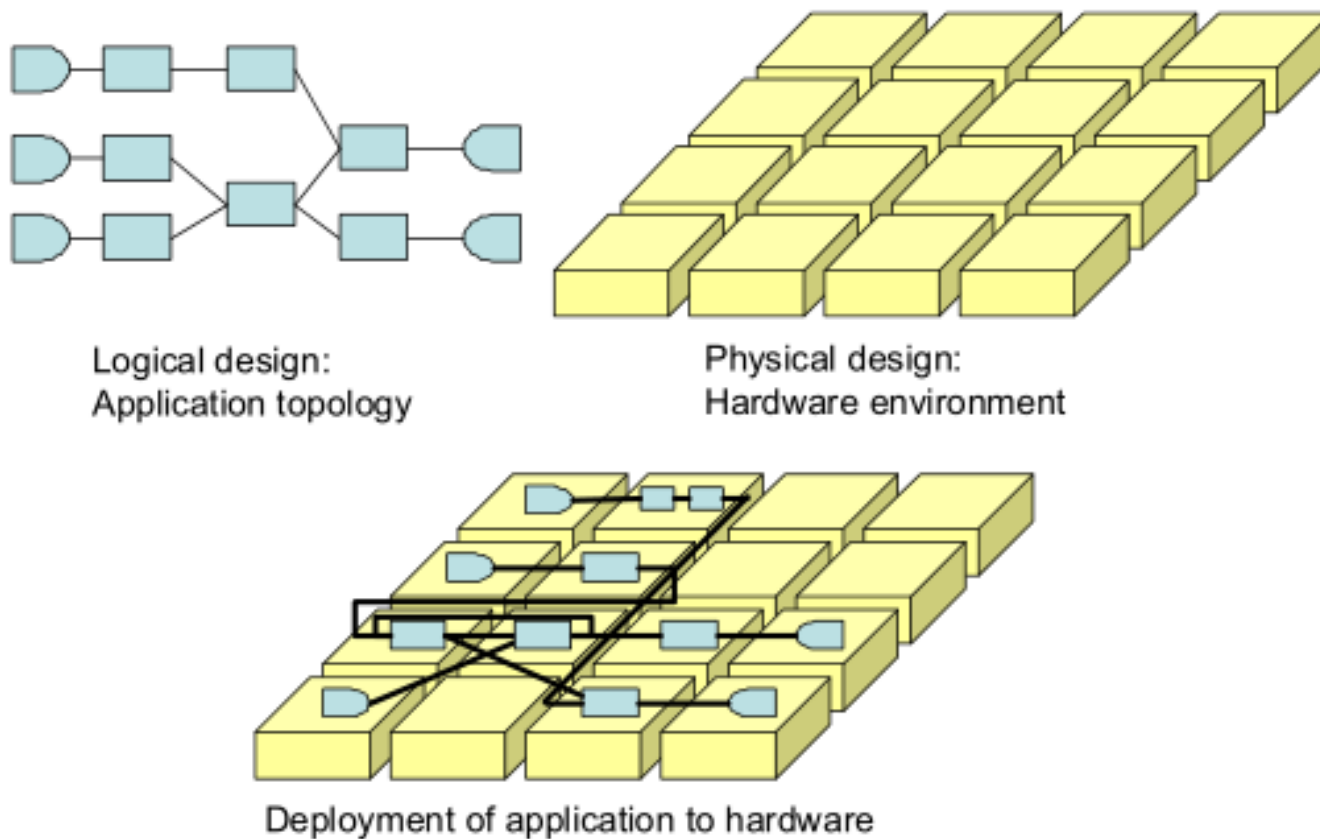
- Autocorrelation methods
 - Where do we differ from the predicted value?
 - Where does the autocorrelation model change?



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

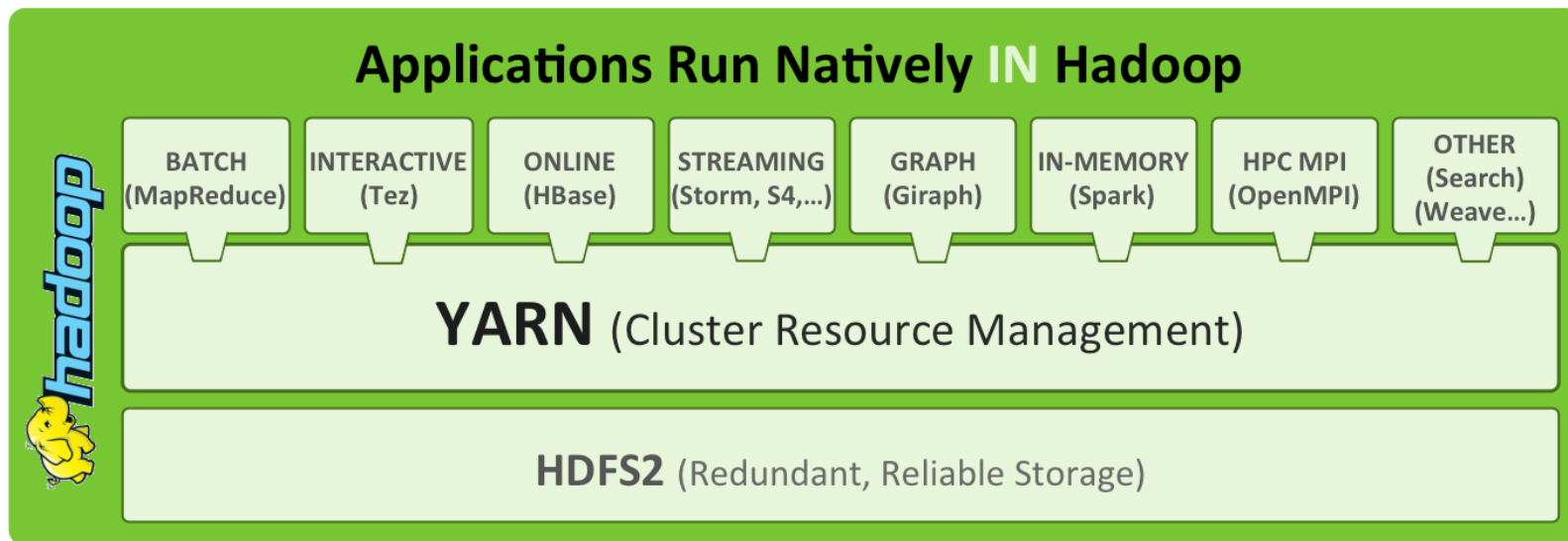


Forrás: [15], p 76

Eszközök

- Apache Spark Streaming
- Apache Storm

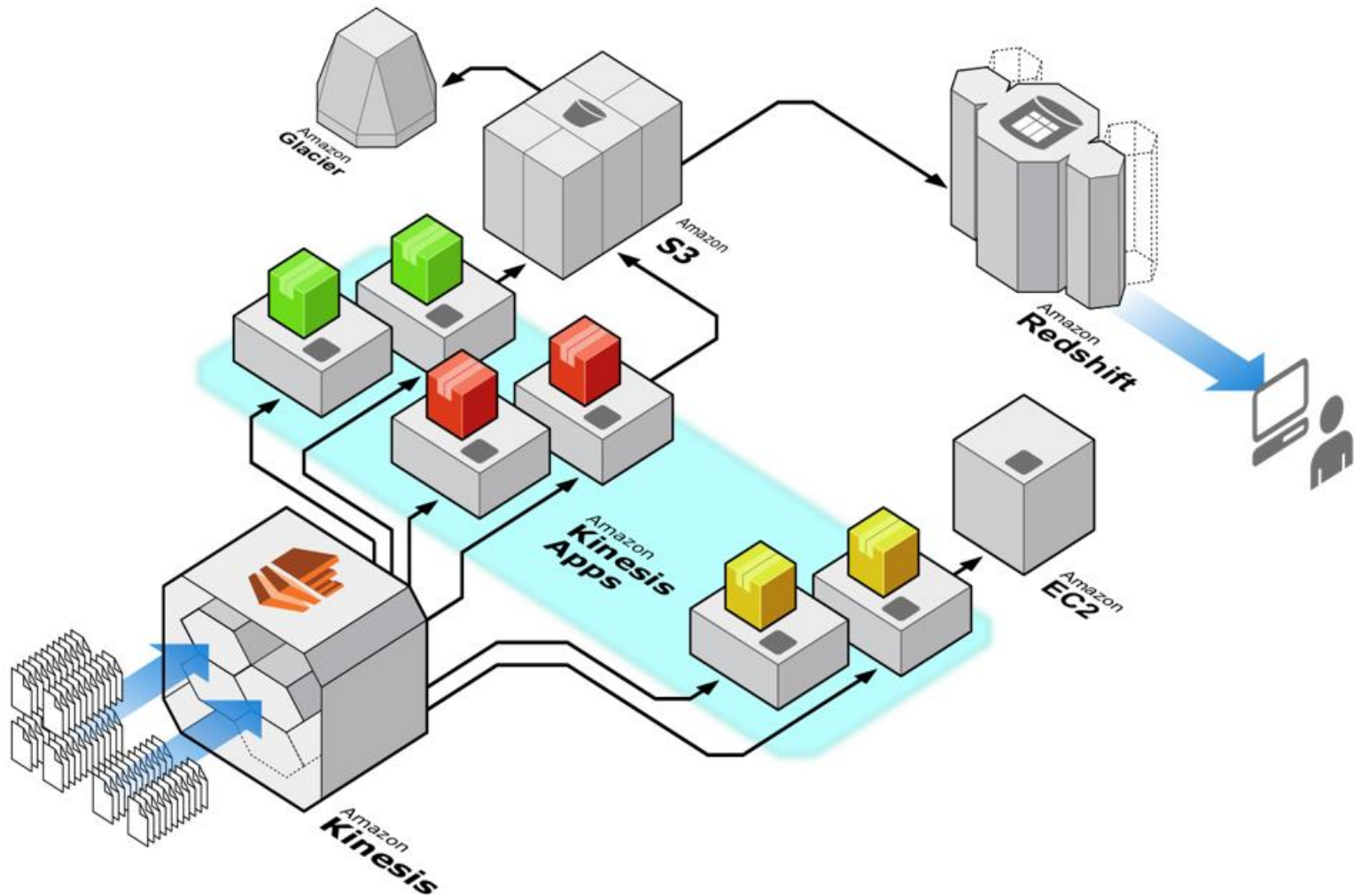
Ábra forrása: [16]



- IBM InfoSphere Streams
- Amazon Kinesis
- ~~LinkedIn~~ Apache Samza
- ...

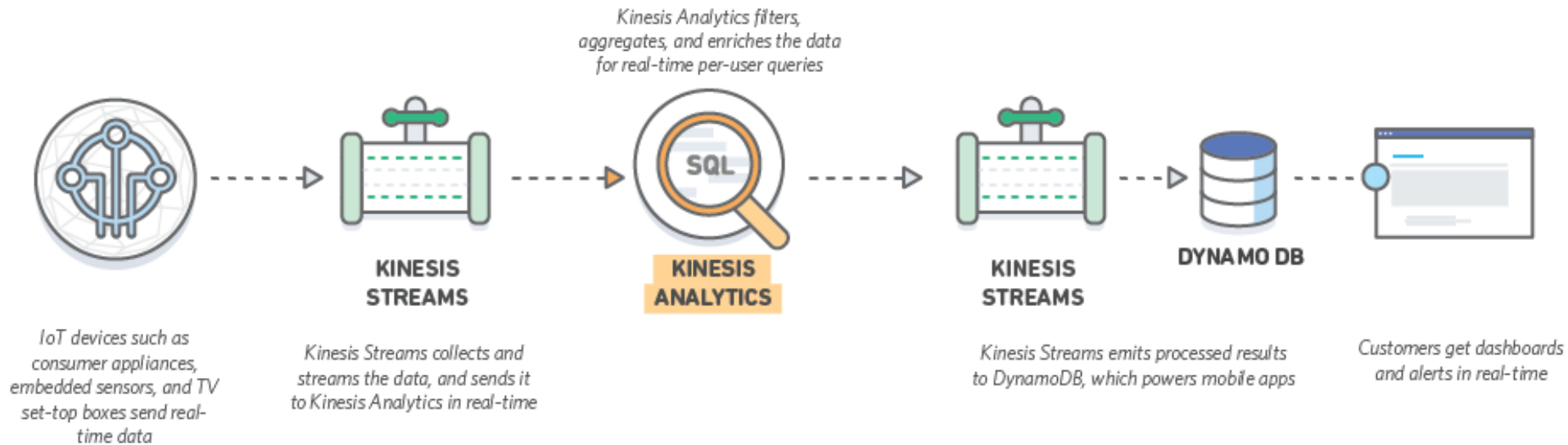
+ kapcsolódó projektek

Kinesis Streams



<https://aws.amazon.com/kinesis/streams/details/>

Kinesis Streams



Folyam-algoritmika

- A számítási modellt láttuk
- Fő korlát: adott tár + WCET, „be nem látott” adat
- Néhány tipikus probléma
 - Mintavételezett kulcstér, kulcsok minden értéke
 - „Elég jó” halmazba tartozás-szűrés kicsi leíróval
 - „Count distinct” *korlátos tárral*
 - Momentumok
- Részletes tárgyalás: Ullman et al.: Mining of Massive Datasets (mmds.org)

Mintavételezés

- Modell:
 - n komponensű elemek
 - ezek egy része *key* (pl. **user**, query, time)
 - a kulcsok felett mintavételezünk
- Probléma
 - Egy kulcsnak vagy minden értéke megjelenjen, vagy egy sem
- Megoldás
 - a/b méretű mintához a (kulcstér)méretű folyamon a kulcsot b vödörbe hasheljük
 - A hash-függvény valójában „konzisztens random-generátor”: $a < b$ esetén “tárolunk” (továbbengedünk)
 - Nem véges minta – kisebb módosítás
- Példa: „a felhasználók mekkora része ismételt megkerdezéseket” a felhasználók $1/10$ mintáján

Apache Kafka

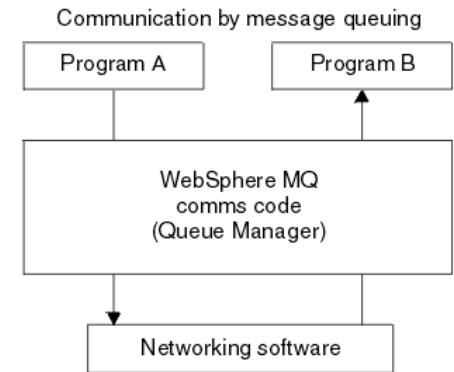
Apache Kafka

■ Valamikor réges-régen

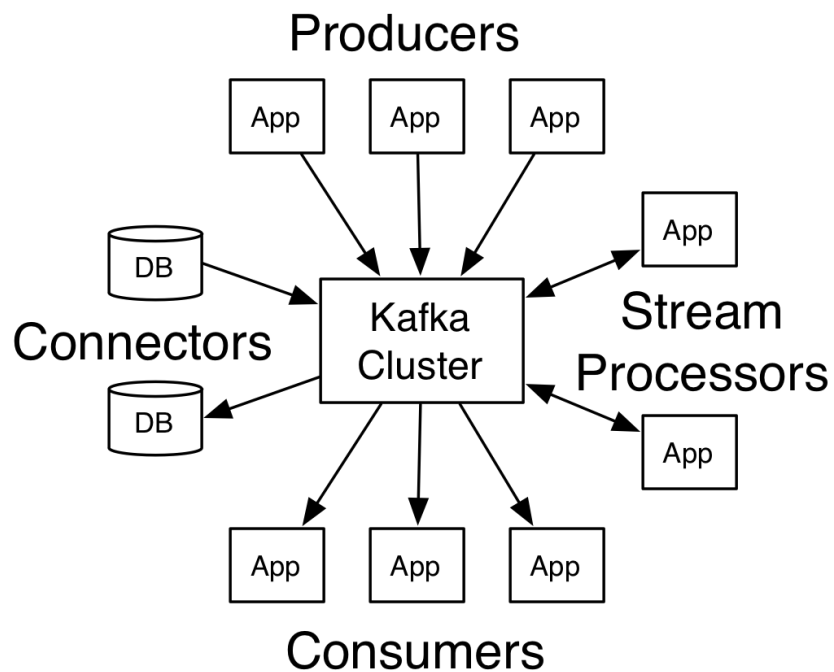
- MQ, pub/sub
- WebSphere MQ, JMS, ...

■ Innováció

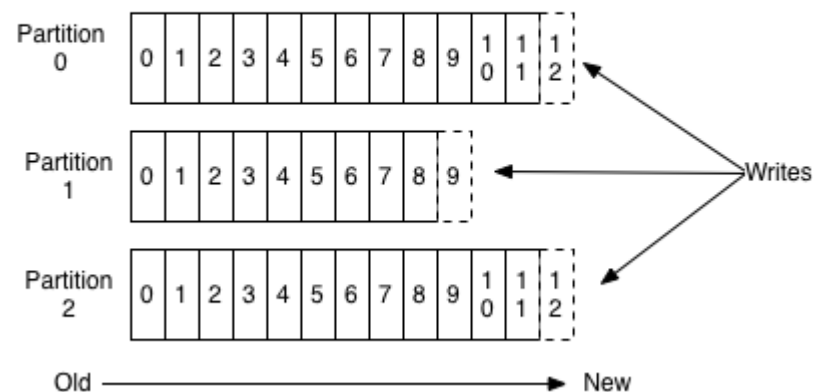
- **F/OSS**
- **DC**
 - elosztott, real time, nagy áteresztőképességű, hibatűrő, perzisztens pub/sub stream-ek
 - pl. Kafka
- **Embedded/IoT/CPS**
 - alacsony overhead, nyílt, esetleg “brokerless”
 - pl. OMG DDS



Apache Kafka

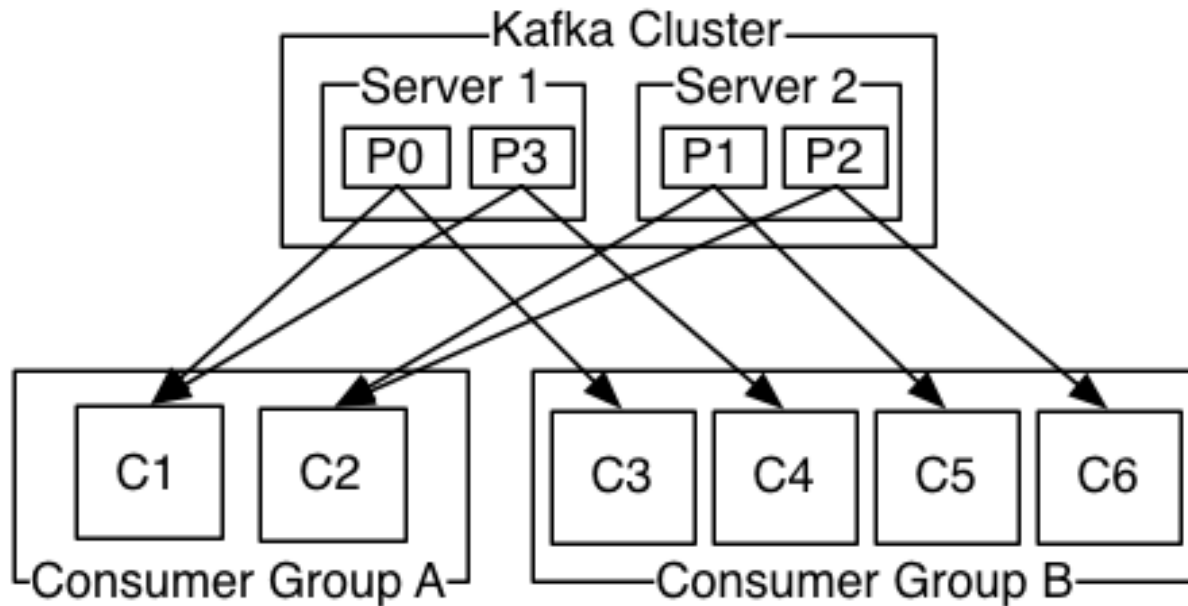


Anatomy of a Topic



Ábrák forrása: <https://kafka.apache.org/intro>

Apache Kafka



- Producer dönti el a partícióba rendelést
- Egy P-t CG-onként egy C olvashat, de több CG (sub)
- Írás és olvasás szigorúan rendezett, de csak partíciónként!
- N- szeres partíció-replikáció: N-1 hiba tűrése

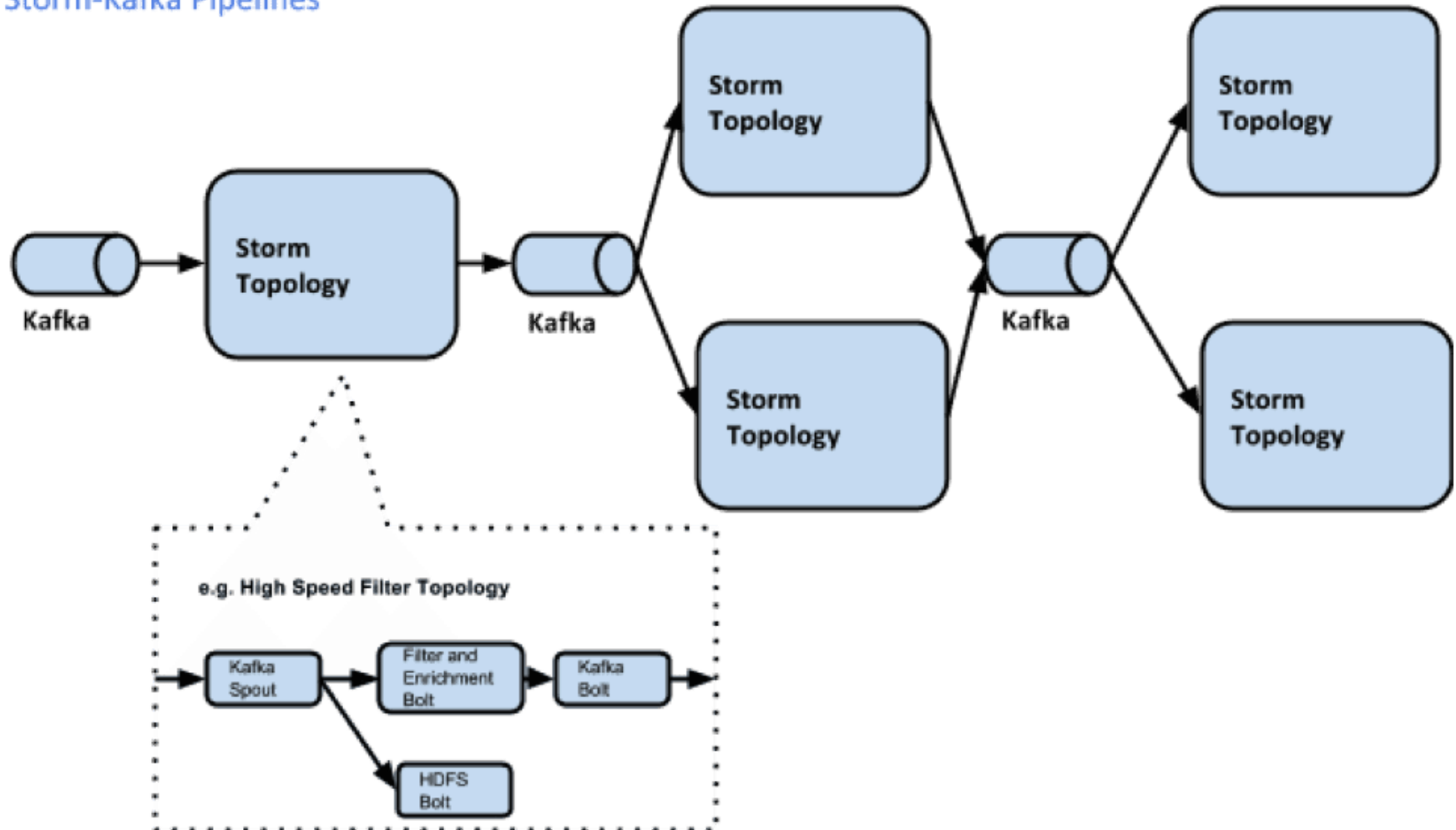
Ábrák forrása: <https://kafka.apache.org/intro>

Saját Streams API

- “Valódi” stream processing
 - Nem “microbatching”
 - Stateless, stateful, windowing
- Akár ms késletetés
- Kafka-integrált mechanizmusokkal: “exactly once”
- Csak egy Java dependency – nem brókeren fut
 - **Hol láttuk már ezt?**

Rendszerintegrációs példa: Kafka + Apache Storm

Storm-Kafka Pipelines



Ábra forrása: <https://hortonworks.com/blog/storm-kafka-together-real-time-data-refinery/>

IaaS teljesítmény

IaaS teljesítmény

Ismeretlen / nem
vezérelhető
űtemezés (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



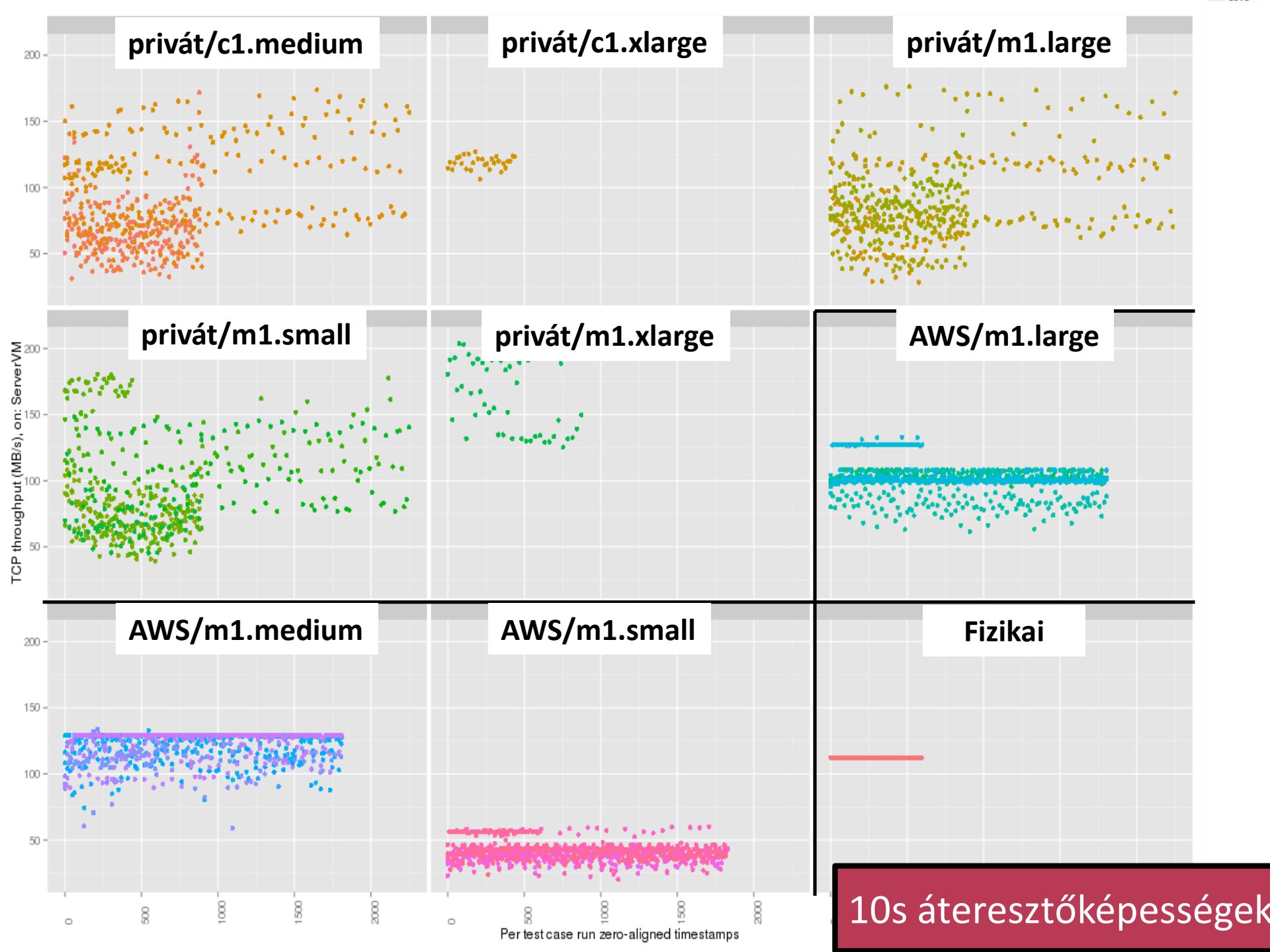
Benchmarkolás!

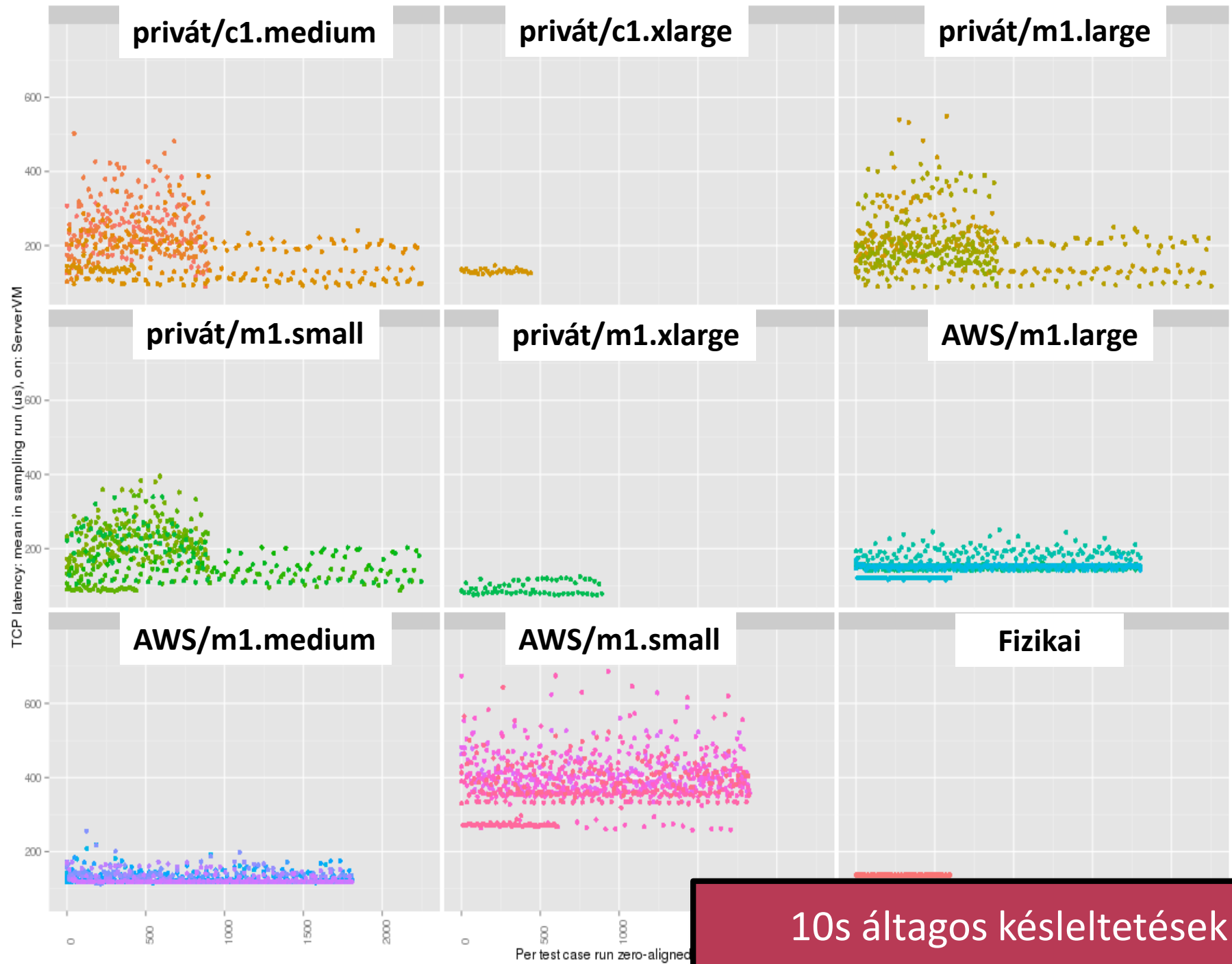
Felhő-specifikus (és fontos): variabilitás [11]

- 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

<https://youtu.be/N3-Az0OH5WM?t=464>

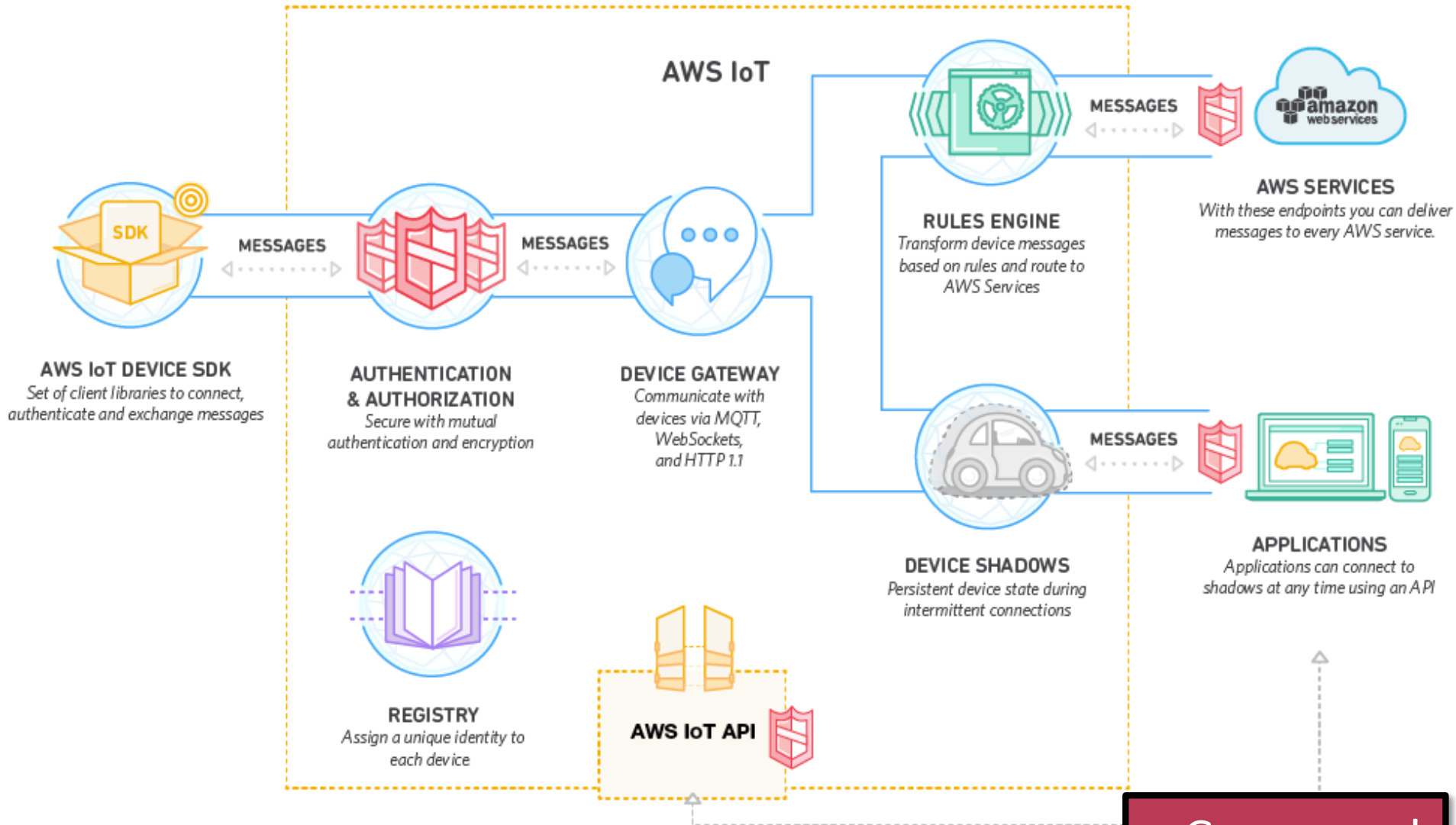






AWS IoT

AWS IoT



<https://aws.amazon.com/iot-platform/how-it-works/>

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., O'Brien, 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 *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>
- [16] <http://hortonworks.com/blog/hdp-2-0-community-preview-and-launch-of-hortonworks-certification-program-for-apache-hadoop-yarn/>