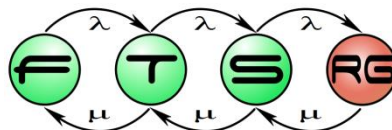


Kritikus rendszerek integrációja laboratórium

Bergmann Gábor,
Honfi Dávid, Nagy András Szabolcs,
Salánki Ágnes, Semeráth Oszkár,
Szárnyas Gábor



A tárgy célja

■ Oktatott téma

- Szolgáltatás alapú rendszerintegráció
 - Komponensek leírása, vezérlése, kommunikációja
 - Monitorozás, elemzés, vizualizáció
- Elosztott üzleti alkalmazások fejlesztése
- Modell alapú folyamattervezés
- Párhuzamos programozás készítése

■ Előfeltétel

- Java alapismeretek
- Angol dokumentáció megértése

■ Oktatási módszer:

- Egyes technológiák kipróbálása egy egyszerű feladaton

A tárgy célja

- Féléven végighúzódó implementációs példafeladat
 - „Szöveghasonlóság-vizsgálat üzletileg kritikus ajánlórendszerben”
 - Részletes specifikáció a weben:
<https://github.com/FTSRG-CSI-2015/csi-lecture-notes/wiki/Running-Example>
 - Az egyes mérések építenek a korábbiakon elkészült megoldásra!
- Segítséget adunk
 - Témánként bevezető segédanyag (nem tankönyv)
<https://github.com/FTSRG-CSI-2015/csi-lecture-notes/wiki>
 - Személyes konzultáció a mérési alkalmak során
 - Virtuális gép telepített fejlesztő- és futtatókörnyezettel
 - Mérőpáronként GitHub repository
 - Automatikus builddel, integrációs tesztel

Követelmények

- Csak olyan mérésre kapható érdemjegy,
 - amelyen a Hallgató részt vesz,
 - a mérés előtt az írásos ellenőrző kérdéseket legalább elégséges szinten megválaszolja,
 - a beugrót komolyan vesszük! Aki felkészületlenül érkezik, az nem kezdheti el a mérést, és pótmérésen kell pótolnia!
 - a mérésen (mérőtárssal közösen) elvégzett munkáját a laborvezető személyes bemutatás alapján jóváhagyja,
 - és az elkészült megoldásról kellő részletességű jegyzőkönyvet ad be

Követelmények

- A félévközi ellenőrzés módja a laboratóriumi mérési feladatok sikeres elvégzése.
- Értékelés:
 - Minden laboratóriumi mérést külön pontszámmal értékelünk, ez
 - a laboratóriumi tevékenység, és
 - a jegyzőkönyv együttes értékeléséből adódik.
 - A félévközi jegy megszerzéséhez minden mérést legalább elégséges szinten (40%) kell teljesíteni!
 - A félévközi jegyet a méréseken szerzett pontszámok átlagából számítjuk.
 - 40% – 55% – 70% – 85%

Feladatok beadása

- A mérőpár részére kiadott Git repositoryba
- Forrás (program kód, konfigurációs leírók, modellek)
- Jegyzőkönyv → ~~Feladat.pdf~~
 - Sima GitHub Markdown is jó, de persze lehet PDF
 - Igényes forma (követhető, rendezett)
- Tartalma:
 - A feladat (röviden)
 - Telepítési útmutató („Getting Started”)
 - Fejlesztői útmutató („Developer’s Guide”):
 - architektúra
 - kihívások és ezek megoldása
 - Technológia és labor értékelése, jobbító javaslatok
 - Beadás: pull request küldése határidőig (vasárnap)
- Előtte a kész megoldás személyes bemutatása!
 - Ideálisan még a mérésen, de legkésőbb péntekig

Pótlás

- Pótlási célból
 - a szorgalmi időszakban egy pótmérési alkalmat biztosítunk,
 - ezen túl a pótlási időszakban is lesz egy pótmérési alkalom.
- A pótmérésen vagy a pótlási időszakban **összesen egy** mérés pótolható.
- A pótmérésre a megadott határidőig jelentkezni kell!

Időzítés

- Első alkalom (laborismertető): **szeptember 7.**
- Hallgatói mérések:

#	Téma	Mérésvezető	Időpont
1.	Munkafolyamatok megvalósítása Java nyelven	Semeráth Oszkár	09.14.
2.	Felderítő adatelemzés	Salánki Ágnes	09.21.
3.	Megbízható üzenetküldés MQ alapon (RabbitMQ)	Honfi Dávid	09.28.
4.	Megbízható üzenetküldés MQ alapon (JMS, JMX)	Honfi Dávid	10.05.
5.	Integráció szabványos sorosítási platformon	Bergmann Gábor	10.12.
6.	Aktor modell konkurens alkalmazások készítésére	Szárnyas Gábor	10.19.
7.	Deklaratív futtatóplatformok	Nagy András Szabolcs	10.26.
8.	Platformfüggetlen modellezés kódgenerálással	Semeráth Oszkár	11.02.

- Pótmérés a szorgalmi időszakban: **november 23.**
 - Pót-pótmérés (reméljük ilyen nem lesz): pótlási hét.

Időzítés

- Egy mérési alkalom ütemezése

H	K	Sze	Cs	P	Szo	V
Segédlet elérhető kb. egy hétrel előbb						
MÉRÉS 8 ³⁰ -12 ⁰⁰ Beugró + munka				Személyes bemutatás határideje (ha a mérésen nem lett kész)		Végleges kód, jegyzőkönyv beadása (pull request)

MÉRÉSEK BEMUTATÁSA

1. MUNKAFLYAMATOK MEGVALÓSÍTÁSA JAVA NYELVEN

Mérésvezető: Semeráth Oszkár

Munkafolyamatok megvalósítása Java nyelven

■ Követelmények

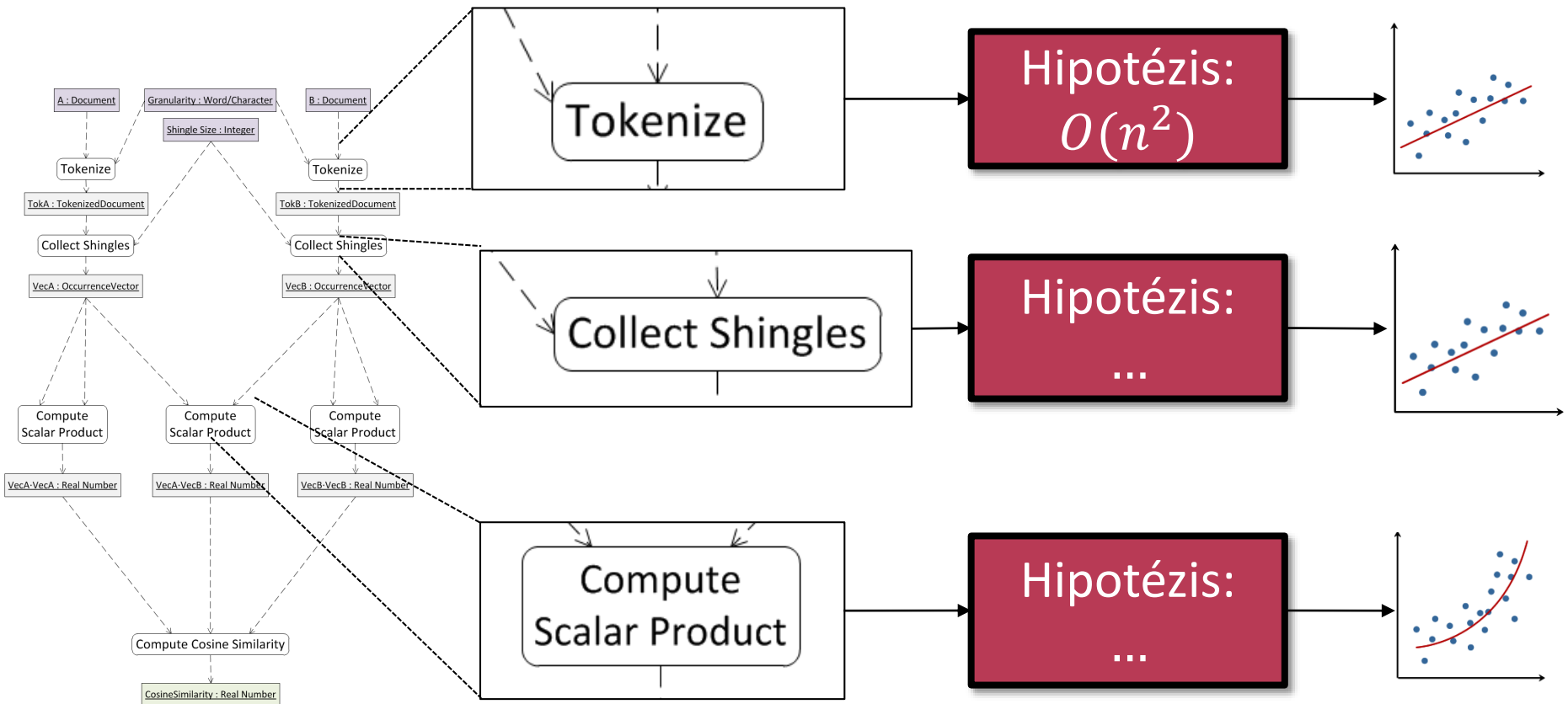
- Minden tevékenység külön szálon fusson
- Kommunikáció a szálak között
 - Elosztottan vagy központi szállal
 - Várakozás vagy értesítést
 - concurrent package
- Adatáramlás – egyszerre több adat is mehet a munkafolyamatban
- Kivételek legyenek kezelve
- Működés naplózása (folyamatok kezdése és befejezése)
- Fork és join műveletek szálak közötti kommunikációval

2. FELDERÍTŐ ADATELEMZÉS

Mérésvezető: Salánki Ágnes

Mérésalapú elemzés

■ Implementációk teljesítményének mérése



Mérésalapú elemzés

■ Mérés céljai

- Hipotézis ellenőrzése
- Szűk keresztmetszetek azonosítása
- Implementációk összehasonlítása

■ Környezet

- Statisztikai elemzés R-ben
- Alapvető vizualizációs technikák
- Regresszió



Felkészülés

- Implementáció/mérés rész elkészítése
 - Eredmény CSV formátumba
- Segédanyag elolvasása
 - A valószínűségszámítás és a statisztika alapfogalmait ismertnek feltételezzük, de linkelünk forrásokat is
- R szintaxisának megtanulása
 - [DataCamp](#)



DataCamp

Amire szükségünk lesz

- **Némi statisztika**

- Kísérlettervezési alapok
- Nagy számok törvénye
- Hipotézistesztelés alapjai

- **Kód instrumentáció**

- **Workflow automatikus lejátszása adott késleltetéssel**
- Szimuláció → megismételhetőség!

- **R**

3. MEGBÍZHATÓ ÜZENETKÜLDÉS RABBITMQ ALAPON

Mérésvezető: Honfi Dávid

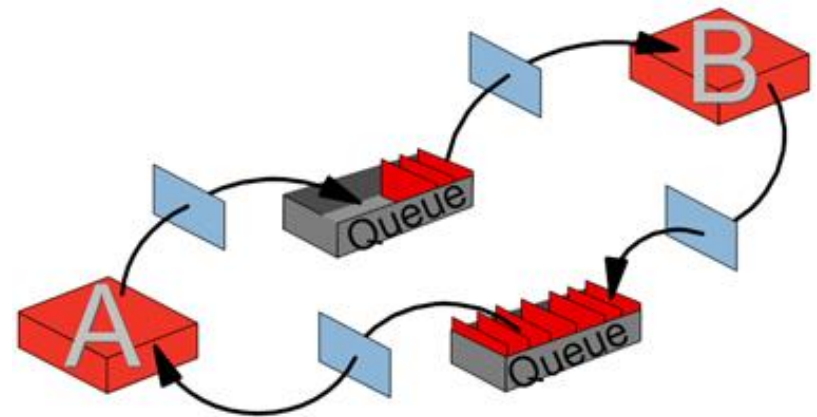
Megbízható üzenetküldés RabbitMQ alapon

■ A mérés céljai

- Üzenetsorok használatának elsajátítása
- RabbitMQ eszköz megismerése

■ Feladatok

- Munkafolyamat kommunikációjának átalakítása üzenet alapúvá
- RabbitMQ használata az üzenetkezeléshez



 RabbitMQ™

4. KOMMUNIKÁCIÓ JMS ÉS JMX TECHNOLÓGIA SEGÍTSÉGÉVEL

Mérésvezető: Honfi Dávid

Kommunikáció JMS és JMX segítségével

- Előző mérésből RabbitMQ csak AMQP protokoll alapú kommunikációt biztosít
 - **Probléma:** Enterprise környezetben *hibrid rendszerek*
- Megoldás lehet *Java Enterprise* segítségével
 - **JMS API:** szabványos üzenetsor-kezelés (Elfedi az implementációs részeket.)
 - **JMX** segítségével konfigurálhatóság
 - **JNDI** segítségével címezhetőség
- **Feladatok**
 - Kommunikáció elkülönítése a kódban
 - Lépések összekötése ki- és bemeneti sorokkal
 - JMX segítségével követhető és vezérelhető üzenetkezelés

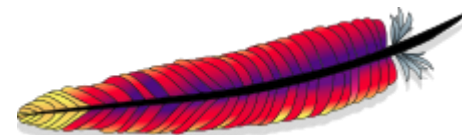


5. INTEGRÁCIÓ SZABVÁNYOS SOROSÍTÁSI PLATFORMON

Mérésvezető: Bergmann Gábor

Apache Thrift

- Platformfüggetlen szolgáltatás-leírónyelv (.thrift)
 - Adatstruktúrák
 - Szolgáltatás interfészek (RPC)
- Kódgenerátor
 - Adatstruktúra megvalósítása
 - Sorosítás és kommunikáció megvalósítása
 - Szerver és kliens váza
 - Különféle esetek (pl. szinkron / aszinkron, nemblokkoló stb.)
 - Célplatformok: Java, C++, C#, Python, Erlang, JS, ...
 - Szerver és kliens nyelve eltérhet!

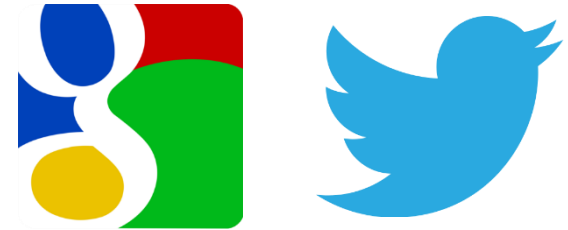


6. AKTOR MODELL KONKURENS ALKALMAZÁSOK KÉSZÍTÉSÉRE: AKKA

Mérésvezető: Szárnyas Gábor

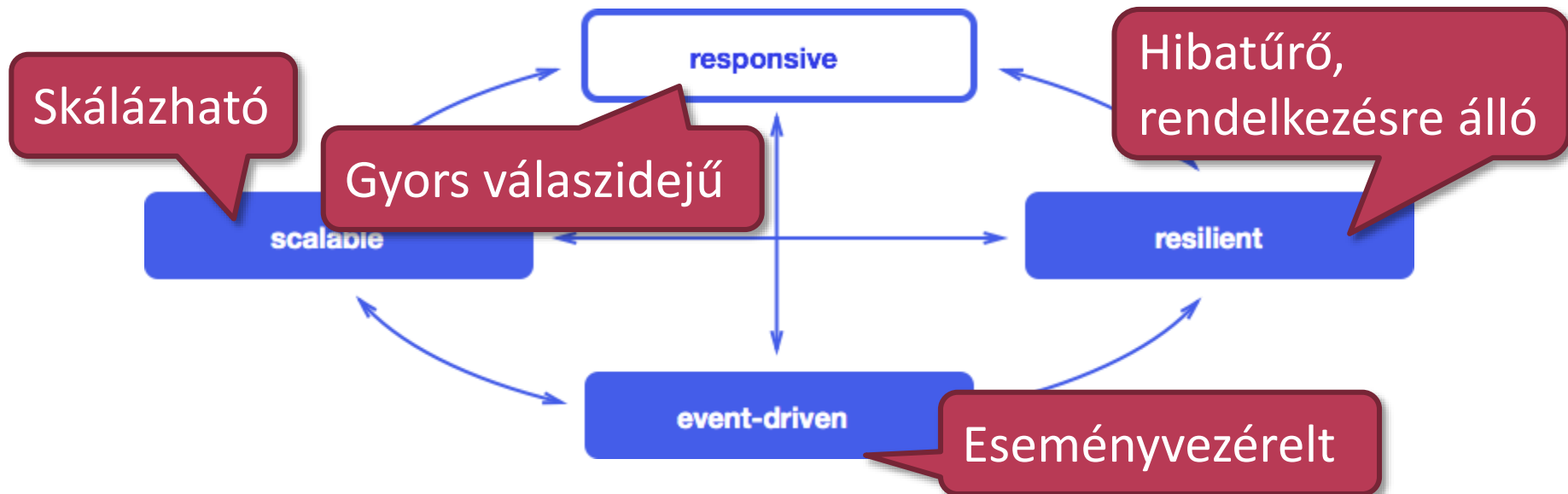
Reaktív alkalmazások

- Alkalmazások
 - Web 2.0: Google, Twitter
 - Pénzügy, telekommunikáció



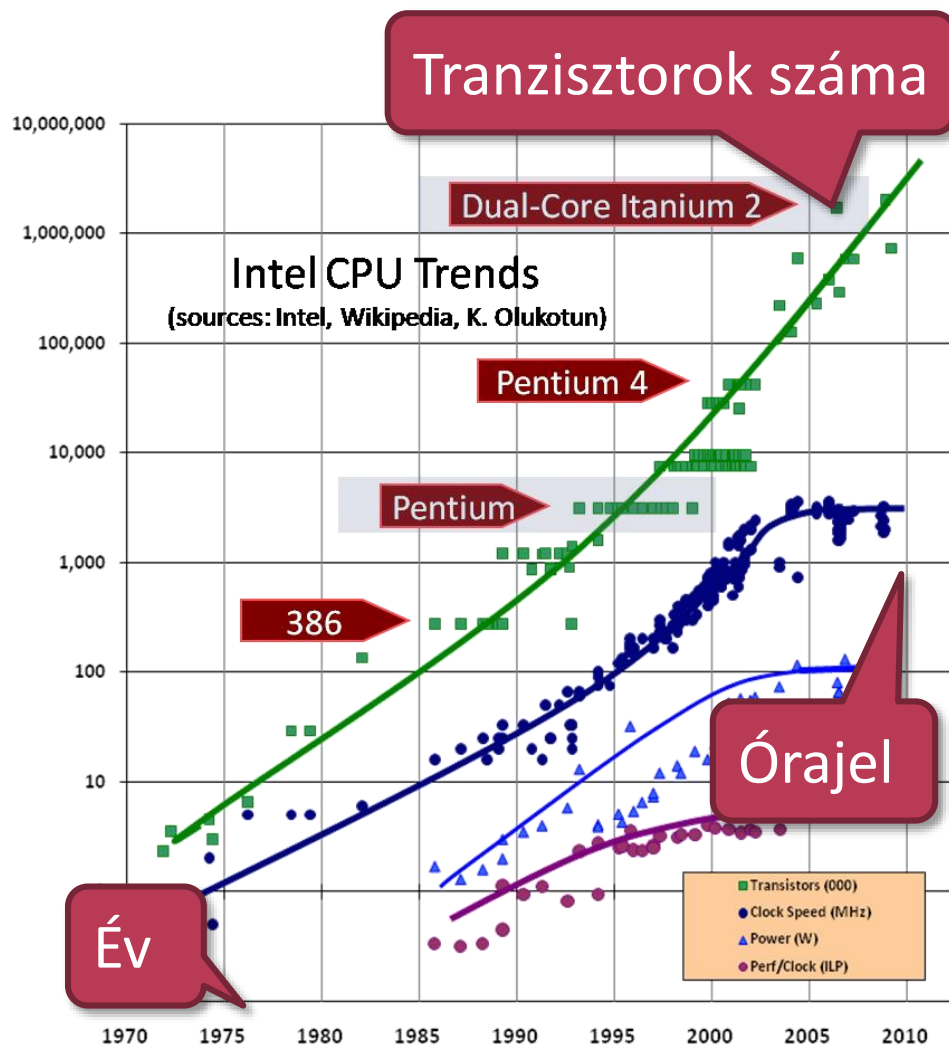
- Reaktív rendszer:

<http://www.reactivemanifesto.org/>

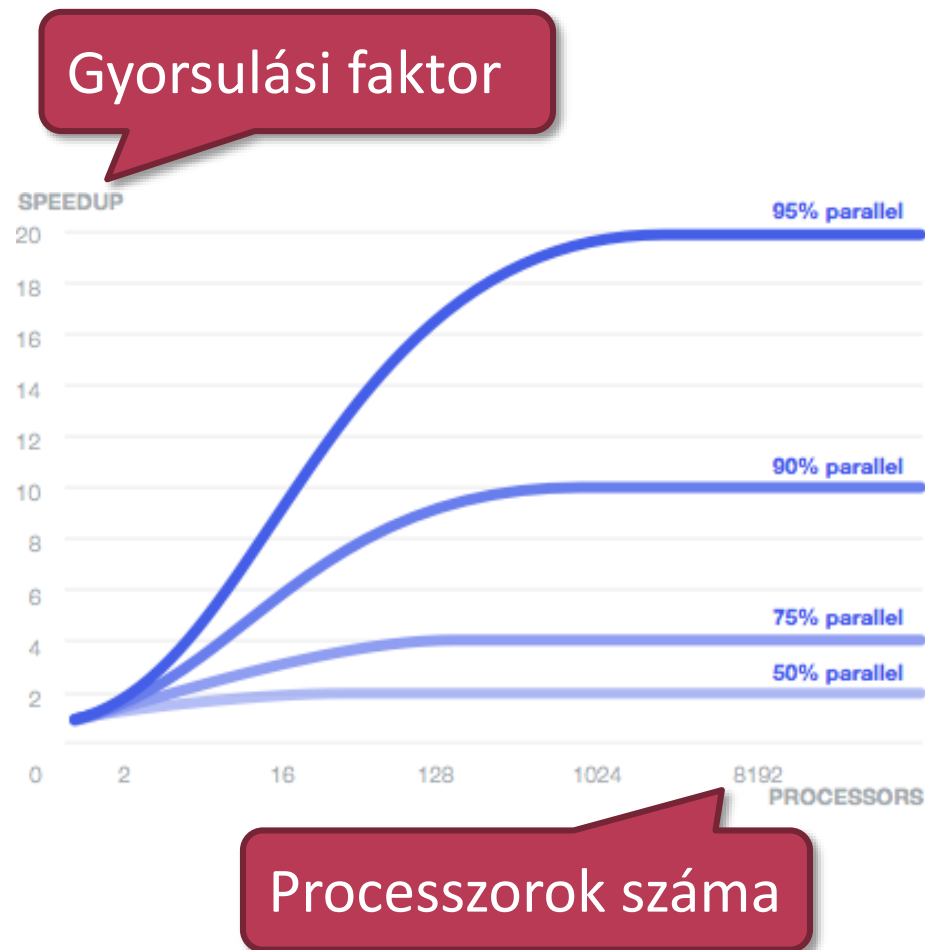


Skálázhatóság

Moore-törvény



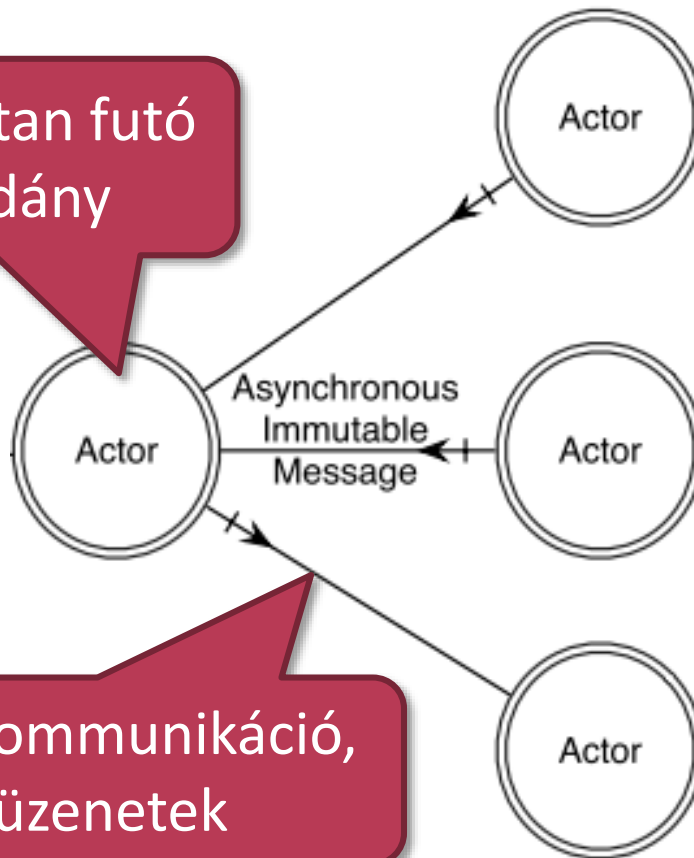
Amdahl-törvény



Párhuzamosítás

- Megszokott módszer: zárok alkalmazása
- Alternatív módszer: **aktor modell** (1973)

Aktor: izoláltan futó
programpéldány



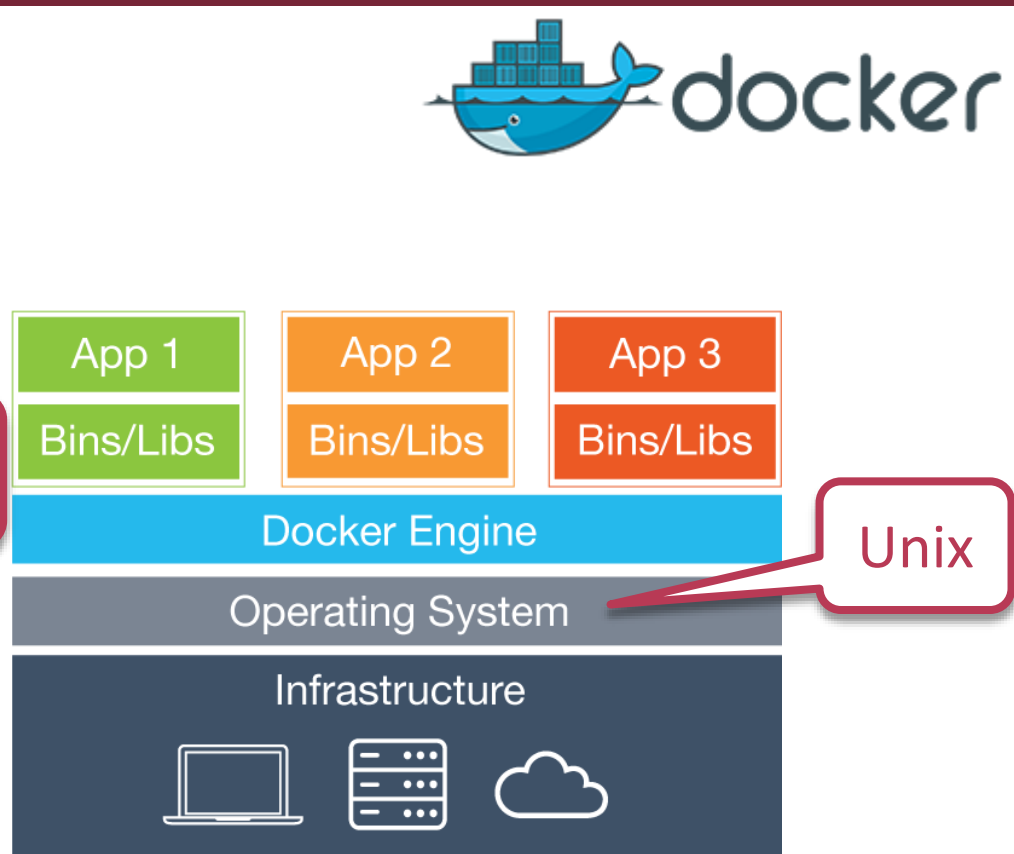
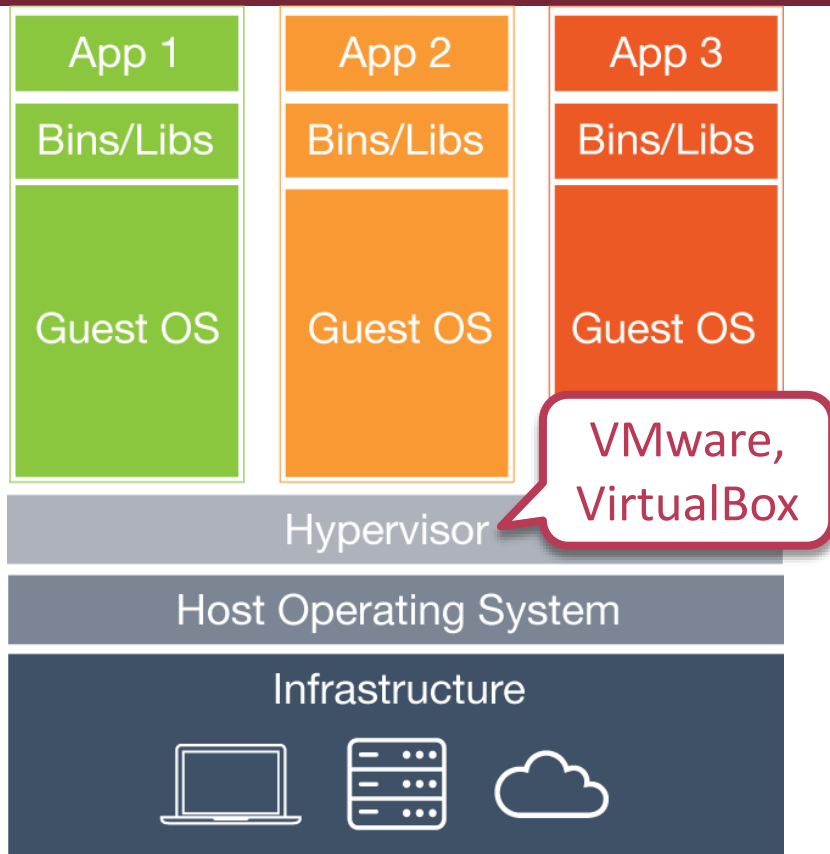
Aszinkron kommunikáció,
immutable üzenetek

The logo for Akka, featuring a stylized blue mountain range above the word 'akka' in a large, bold, black sans-serif font.

7. DEKLARATÍV FUTTATÓPLATFORMOK

Mérésvezető: Nagy András Szabolcs

Docker



Platform virtualizáció
+ Teljes izoláció
- Nagy overhead

Alkalmazásszintű virtualizáció
+ Minimális vagy nulla overhead
- OS függő

Docker

■ Docker alapismeretek

- Nyílt forráskódú, Go nyelven készült
- Dockerfile: egy képfájlt definiál
- Konténerek parancssorból építhetők és indíthatók
- Docker Hub: központi hely képfájlok tárolására

■ Előző feladat

- Elosztott munkafolyamat Akka segítségével

■ Új feladat

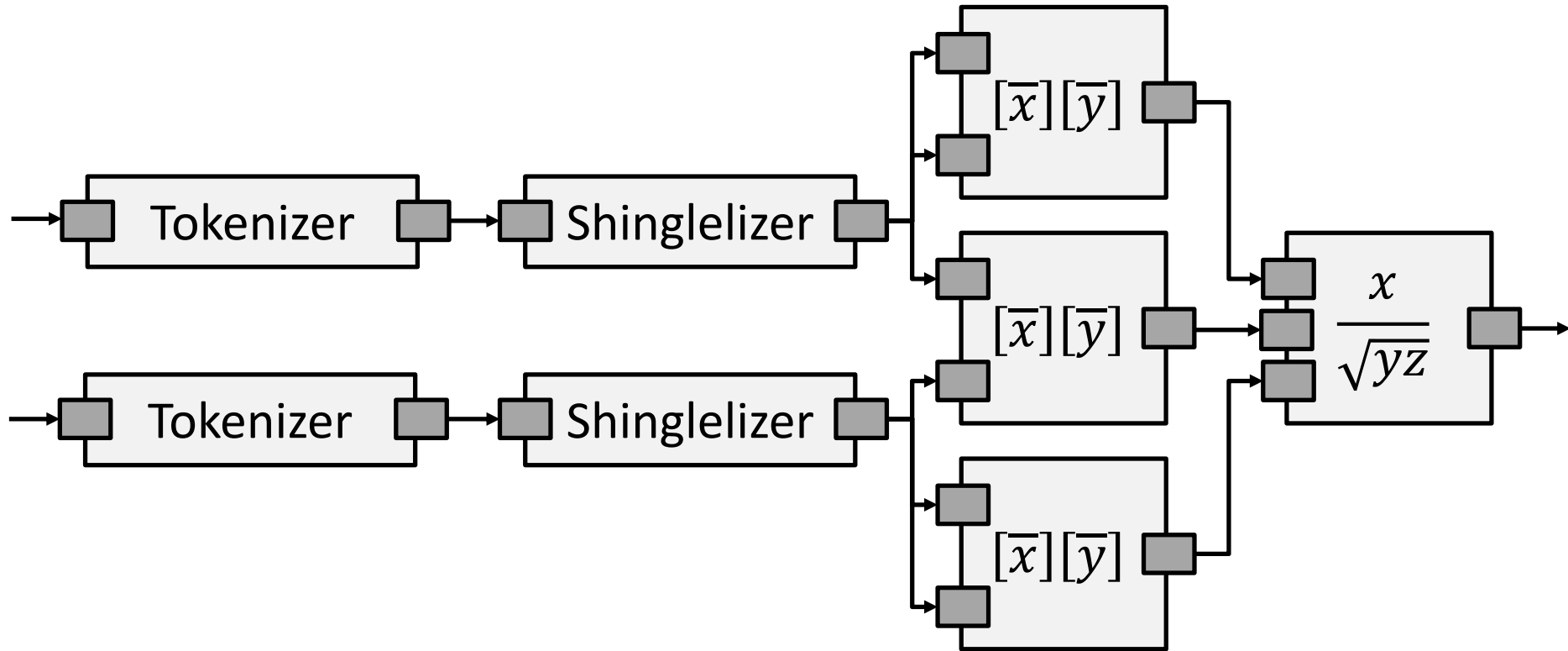
- Docker képfájlok készítése aktorok futtatásához
- Ezek telepítése és futtatása (deployment)

8. PLATFORMFÜGGETLEN MODELLEZÉS GENERATÍV PROGRAMOZÁSSAL

Mérésvezető: Semeráth Oszkár

- Célok
 - Különböző technológiák integrációja/lecserélése
 - Folyamat átstrukturálása automatikusan
- Bonyolult + sok + ismétlődő összekötő kód
- Feladat: Folyamatmodellező nyelv + kódgenerálás
 1. Szétválasztani: generálható $\Leftrightarrow$ manuális kódrészeket
 2. Modellezni: a generálható részt
 3. Kódgenerátort készíteni: a modellező nyelvhez
 4. Összekötni: generált + manuális részeket
- Technológiák: EMF, Xtend, Eclipse fájlkezelés

Platformfüggetlen modellezés generatív programozással



AKTUÁLIS TEENDŐK

Aktuális teendők

- Mérőpárok alakítása
- GitHub repository, build rendszer megismerése
 - Id. mai óra hátralévő része
- Jövő hétig:
 - megoldandó feladat elolvasása, megértése
 - (mint minden hétre)
segédanyag elolvasása, felkészülés a beugróra