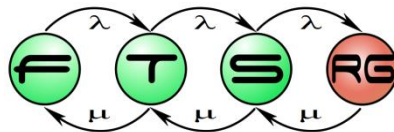


Kritikus rendszerek integrációja laboratórium

Dr. Bergmann Gábor,
Klenik Attila, Dr. Semeráth Oszkár,
Dr. Szatmári Zoltán, **Dr. Szárnyas Gábor**

<https://inf.mit.bme.hu/edu/courses/krilab>



A tárgy célja

- Oktatott téma
 - Szolgáltatás alapú rendszerintegráció
 - Komponensek leírása, vezérlése, kommunikációja
 - Monitorozás, elemzés, vizualizáció
 - Elosztott üzleti alkalmazások fejlesztése
 - Modell alapú folyamattervezés
 - Párhuzamos programozás
- 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/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/csi-lecture-notes/wiki>
 - Személyes konzultáció a mérési alkalmak során
 - Virtuális gép telepített fejlesztő- és futtatókörnyezettel
 - Mérőpáronként GitHub repository
 - Automatikus builddel, integrációs teszttel

Kapcsolódó tárgyak

- MSc 1. félév
 - Modell alapú szoftvertervezés
 - Szolgáltatásorientált rendszerintegráció
- MSc 2. félév
 - Kiberfizikai rendszerek
 - Szoftver- és rendszerellenőrzés
 - *Kritikus rendszerek integrációja laboratórium*
- MSc 3. félév
 - Kritikus architektúrák laboratórium

Követelmények

A mérési alkalom elfogadási feltétele: a mérőpár...

- ...elvégzett munkáját a laborvezető személyes bemutatás alapján jóváhagyja
- ...az elkészült megoldásról kellő részletességű jegyzőkönyvet készít
- ...a kész megoldást és a jegyzőkönyvet a verziókezelőbe feltölti, tageli

Mit adunk:

- Mérési útmutató, felkészülési anyagok
- A mérés időpontjában rendelkezésre álló mérésvezető
 - „Tech support” vagy rögtön a megoldás bemutatása

Beugró, jelenlét

- Beugró, jelenlét „feltételül szabható”
 - Kipróbáljuk egyelőre ezek nélkül
 - Laboralkalmon megjelenés nem kötelező
 - Saját felelősségre
 - Ha nem működik → félév közben visszaállíthatjuk
- Ennek ellenére érdemes...
 - ...a segédanyagból felkészülni a megoldás előtt
 - ...az ellenőrző kérdéseket átgondolni
 - ...a mérésvezető segítségét igénybe venni
 - Csak a laboralkalmon garantált!
 - Órán kívül segítséget nem ígérhetünk

Követelmények

- A félévközi ellenőrzés módja a laboratóriumi mérési feladatok sikeres elvégzése.
- Értékelés:
 - Minden laboratóriumi mérést külön pontszámmal értékelünk, ez
 - Az elkészült megoldás / forráskód minősége, 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 (ld. később)
 - Minden mérési feladat külön Git tag!
 - Elnevezési konvenció: labX, pl. lab1
 - Beadás: Git tag elhelyezése határidőig (vasárnap)
 - A javítás részletes eredményét GitHub issue-ban közöljük.
- Előtte a kész megoldás személyes bemutatása!
 - Ideálisan még a mérésen, de legkésőbb péntekig
 - A jegyzőkönyv itt még nem kell
- Travis CI működése kötelező (2 pont)
 - GitHub / Travis / Maven repository outage-ot leszámítva

Jegyzőkönyv

- Dokumentáció: docs / könyvtárba
 - Pl. `1-basic-java-implementation.md`
 - Plain text, vagy GitHub Markdown a javasolt
 - Igényes forma (követhető, rendezett), magyar/angol
- Tartalma, lényegretörően:
 - A feladat
 - Telepítési útmutató („Getting Started”)
 - Fejlesztői útmutató („Developer’s Guide”):
 - architektúra
 - kihívások és ezek megoldása
 - Technológia és labor értékelése, jobbító javaslatok

Pótlás

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

Időzítés

- Első alkalom (laborismertető): **szeptember 9.**
- További mérések:

#	Téma	Mérésvezető	Időpont	Tag
2.	Java alkalmazások párhuzamosítása	Szárnyas Gábor	09.16.	lab2
3.	Munkafolyamat keretrendszer megvalósítása	Semeráth Oszkár	09.23.	lab3
4.	Aktor modell konkurens alkalmazások készítésére	Szárnyas Gábor	10.07.	lab4
5.	Integráció szabványos távoli eljárás hívás platformon	Bergmann Gábor	10.14.	lab5
6.	Deklaratív alkalmazáskonténer platform	Szatmári Zoltán	10.21.	lab6
7.	Teljesítményelemzést segítő adatelemzési technikák	Klenik Attila	10.28.	lab7
8.	Platformfüggetlen modellezés, kódgenerálás 1.	Semeráth Oszkár	11.04.	lab8
9.	Platformfüggetlen modellezés, kódgenerálás 2.	Semeráth Oszkár	11.11.	-

- Pótmérés a szorgalmi időszakban: **november 25.**
 - Pót-pótmérés (reméljük ilyen nem lesz): pótlási hét, dec. 16.

Időzítés

- Egy mérési alkalom ütemezése

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

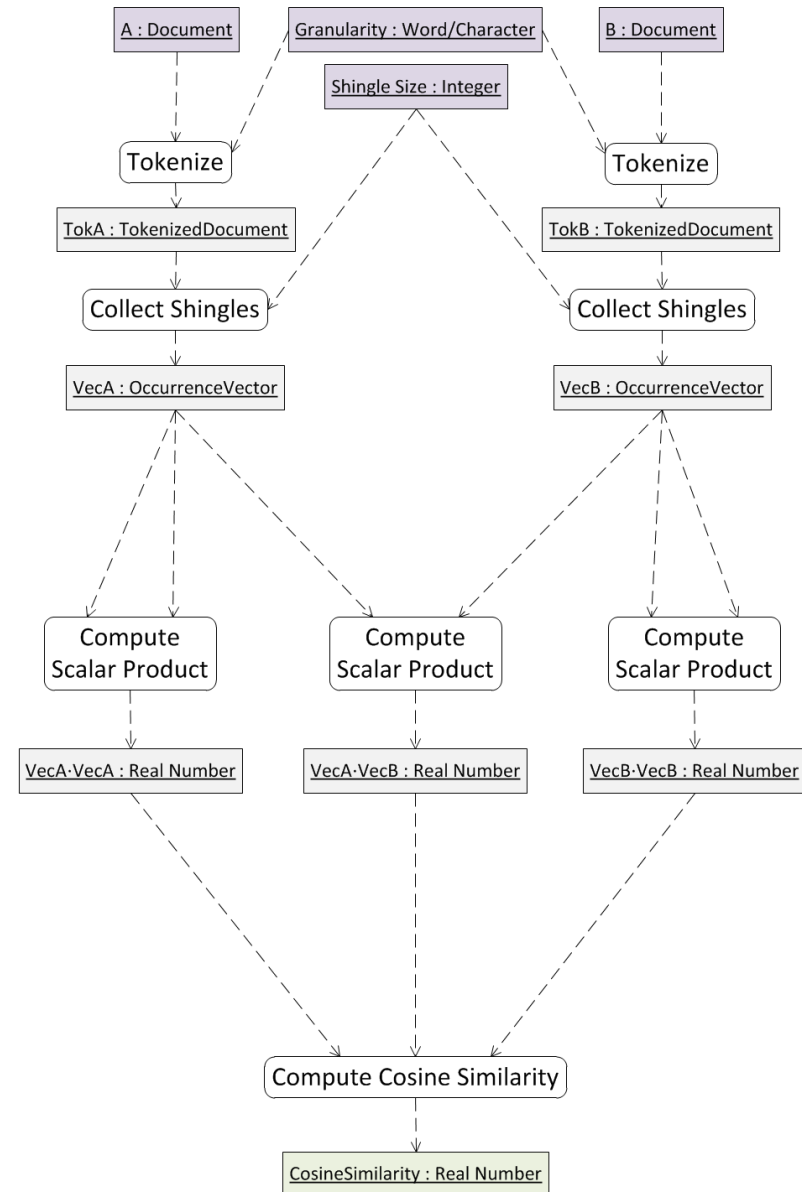
MÉRÉSEK ÁTTEKINTÉSE

1. PÉLDAFELADAT MEGVALÓSÍTÁSA JAVA NYELVEN

Mérésvezető: Szárnyas Gábor

Példafeladat

- Részletes specifikáció:
<https://github.com/FTSRG-CSI/csi-lecture-notes/wiki/Running-Example>



2. JAVA ALKALMAZÁSOK PÁRHUZAMOSÍTÁSA

Mérésvezető: Szárnyas Gábor

Java alkalmazások párhuzamosítása

■ 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

Párhuzamos programozás

- Concurrency: It's Harder (and Easier) Than You Think: <https://vimeo.com/130390892>
- Simonyi Konferencián is lement „Your multi-threaded code is broken (and I can prove it)” címmel: <https://bss.sch.bme.hu/video/simonyi-konferencia-2015-your-multi-threaded-code-is-broken-and-i-can-prove-it>

3. MUNKAFOLYAMAT KERETRENDSZER MEGVALÓSÍTÁSA

Mérésvezető: Semeráth Oszkár

Munkafolyamatok megvalósítása Java nyelven

■ Követelmények

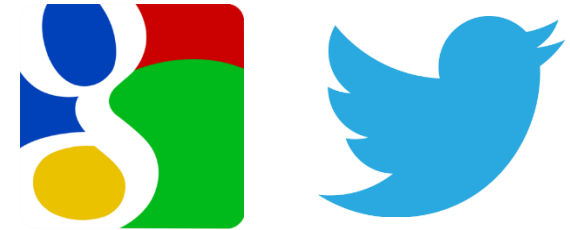
- Munkafolyamat keretrendszer megvalósítása
 - Többszálú megoldás (ld. 2. labor)
 - Többszálú viselkedés elfedése a csomópontok implementációjában
 - Átkonfigurálhatóság, újrafelhasználhatóság
- Java Generics megismerése
 - Típushelyes megoldás
 - Indokolatlan típusmegszorítások nélkül
 - Type cast nélkül
 - Warning nélkül (@SuppressWarnings nem ér 😊)

4. AKTOR MODELL KONKURENS ALKALMAZÁSOK KÉSZÍTÉSÉRE: AKKA

Mérésvezető: Szárnyas Gábor

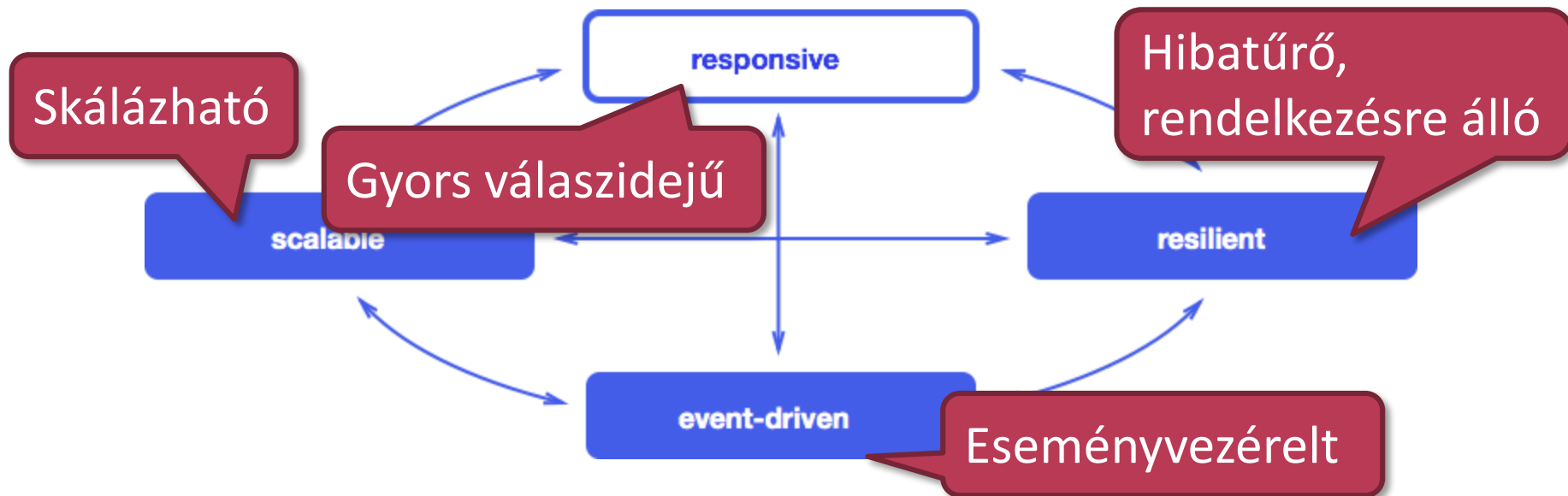
Reaktív alkalmazások

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



- Reaktív rendszer:

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



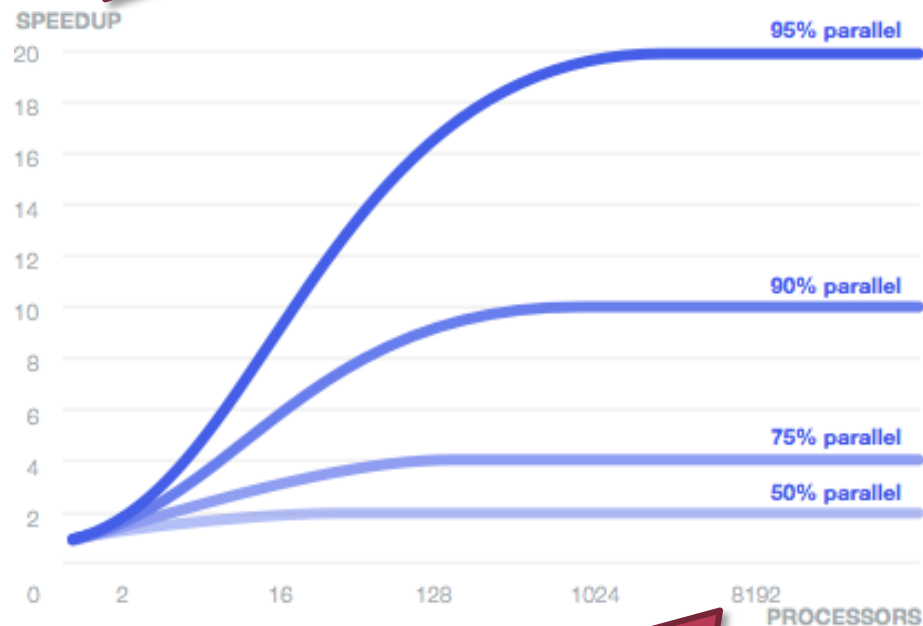
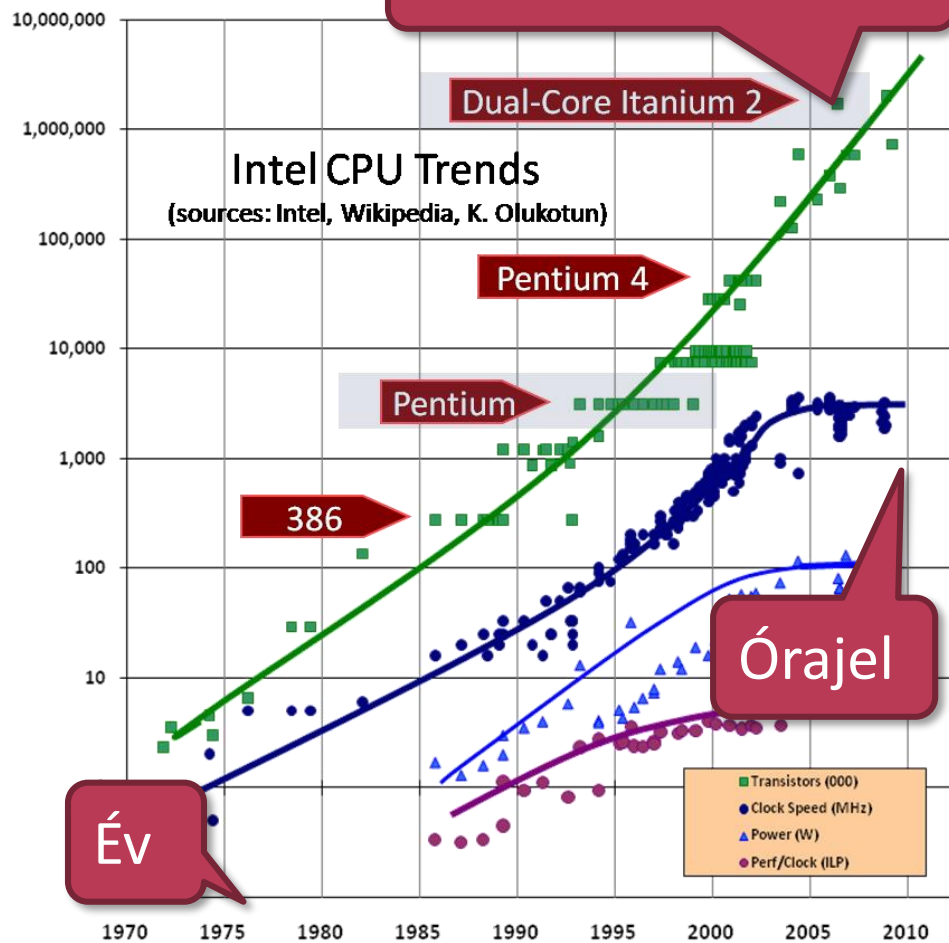
Skálázhatóság

Moore-törvény

Amdahl-törvény

Tranzisztorok száma

Gyorsulási faktor

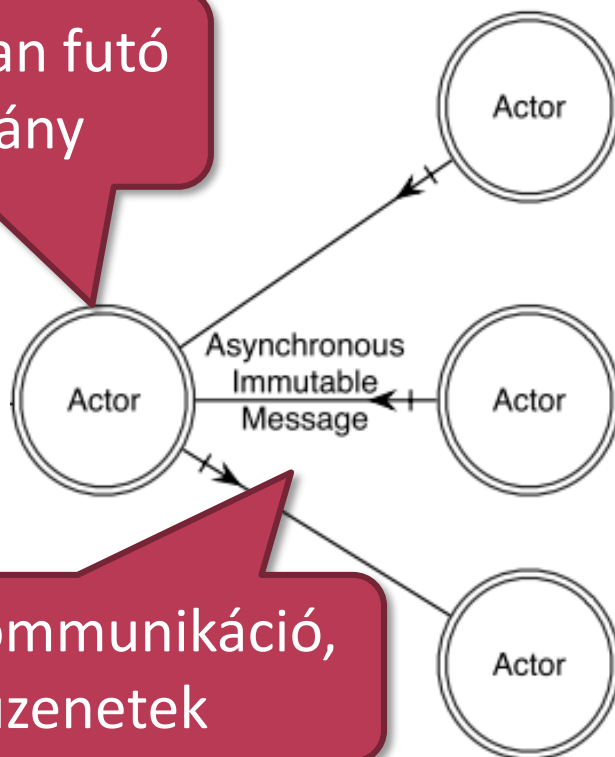


Processzorok száma

Párhuzamosítás

- Megszokott módszer: zárok alkalmazása
- Alternatív módszer: **aktor modell** (1973)
- Az aktorok **elosztottan** is futtathatók

Aktor: izoláltan futó
program példány



Aszinkron kommunikáció,
immutable üzenetek



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

5. INTEGRÁCIÓ SZABVÁNYOS TÁVOLI ELJÁRÁSHÍVÁS PLATFORMON

Mérésvezető: Dr. Bergmann Gábor

Apache Thrift

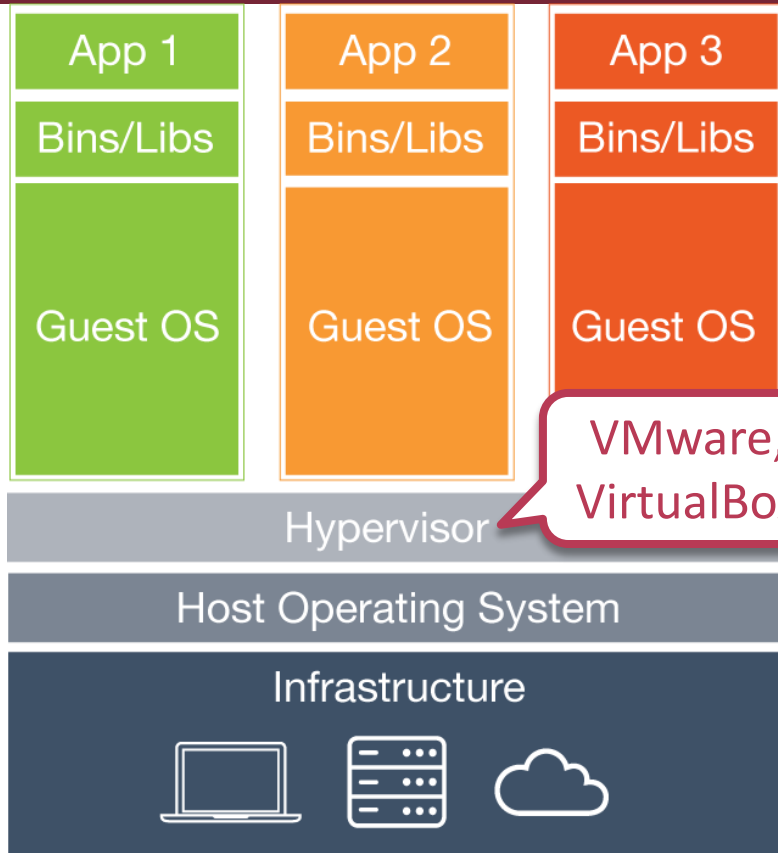
- Platformfüggetlen szolgáltatás-leírónyelv (.thrift)
 - Adatstruktúrák
 - Szolgáltatás interfészek (RPC)
- Kódgenerátor
 - Adatstruktúra megvalósítása
 - Sorosítás és kommunikáció megvalósítása
 - Szerver és kliens váza
 - Különféle esetek (pl. szinkron / aszinkron, nemblokkoló stb.)
 - Célplatformok: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



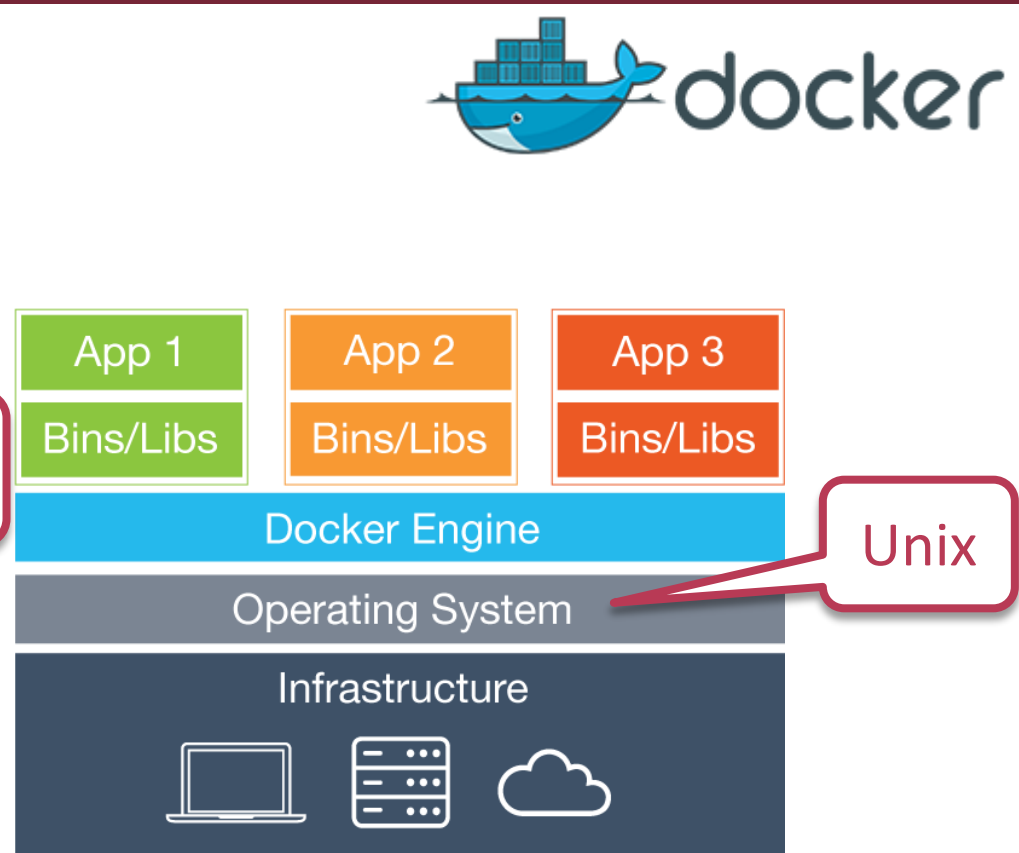
6. DEKLARATÍV ALKALMAZÁSKONTÉNER PLATFORM

Mérésvezető: Szatmári Zoltán

Docker



VMware,
VirtualBox



Unix

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
 - Docker Hub: központi hely képfájlok tárolására
- Konténerek parancssorból építhetők és indíthatók
 - Compose: container orchestration

■ Korábbi feladat

- Elosztott munkafolyamat Akka/Thrift segítségével

■ Új feladat

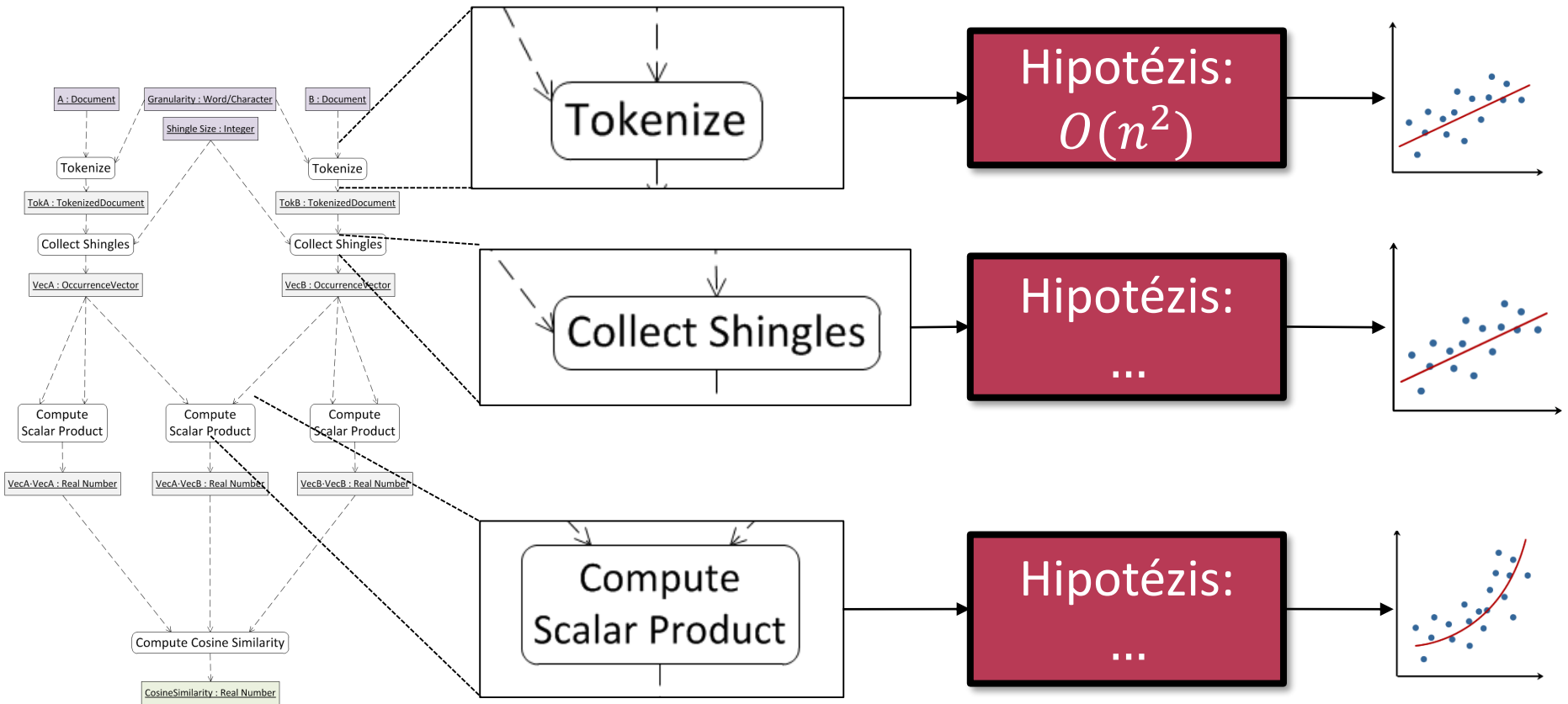
- Docker képfájlok készítése szolgáltatások futtatásához
- Ezek telepítése és együttes futtatása

7. TELJESÍTMÉNYELEMZÉST SEGÍTŐ ADATELEMZÉSI TECHNIKÁK

Mérésvezető: Klenik Attila

Mérésalapú elemzés

■ Implementációk teljesítményének mérése



Amire szükségünk lesz

■ Némi statisztika

- Kísérlettervezési alapok
- Nagy számok törvénye
- Hipotézistesztelés alapjai

■ Kód instrumentáció

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

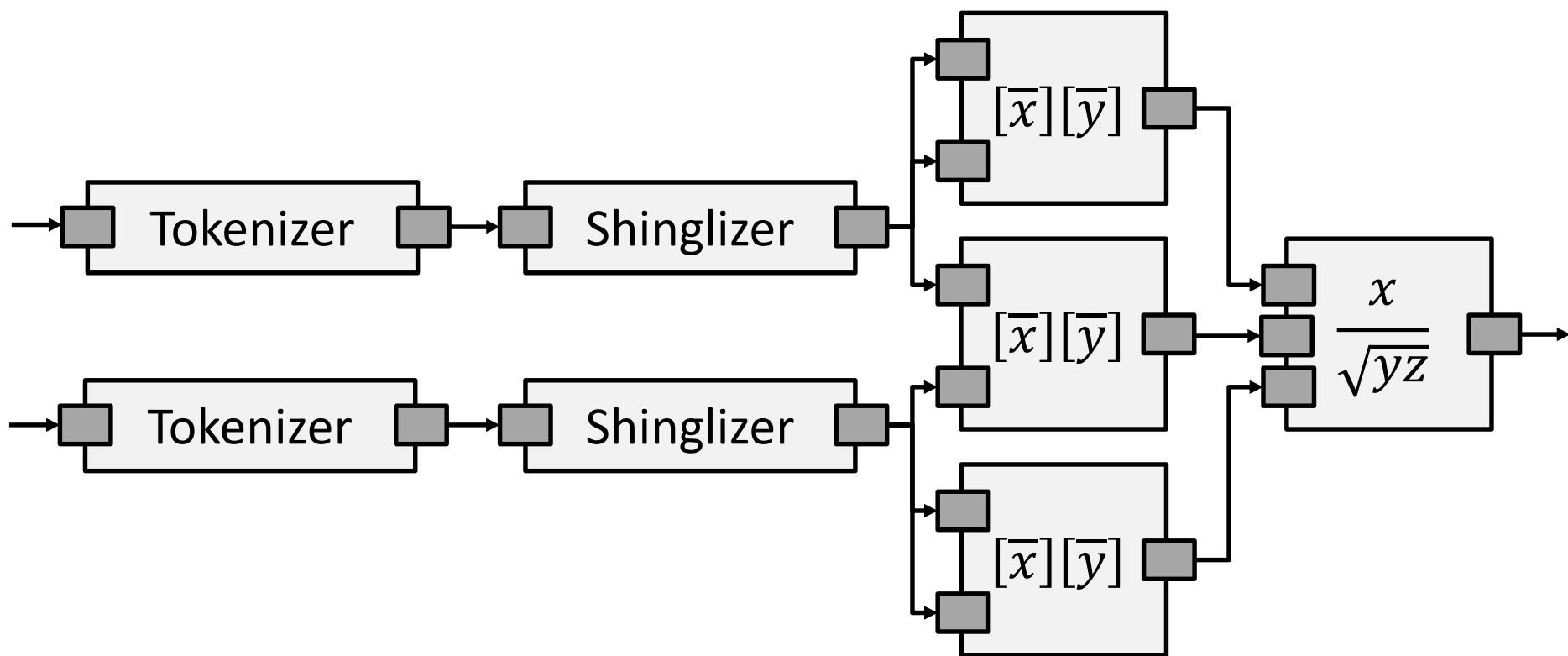
■ Python

8-9. PLATFORMFÜGGETLEN MODELLEZÉS GENERATÍV PROGRAMOZÁSSAL

Mérésvezető: Dr. Bergmann Gábor

- 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

Platformfüggetlen modellezés generatív programozással



AKTUÁLIS TEENDŐK

Aktuális teendők

- Mérőpárok alakítása: <http://tiny.cc/csi-lab>
- Mai óra hátralevő része
 - GitHub repository, build rendszer megismerése
 - Megoldandó feladat elolvasása, megértése
 - Erősen javasolt: megoldás elkezdése
 - Sima, egyszálú Java program elég
- Jövő héttől minden alkalomra:
 - segédanyag elolvasása
 - megoldás elkészítése (akár laboron)
 - feltöltés, egyeztetett időpontban bemutatás