

# Felhő alapú alkalmazások teljesítményének kiértékelése és modellezése

Konzulens: Dr. Rétvári Gábor

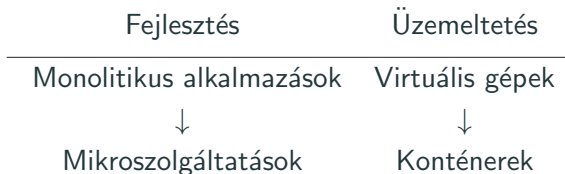
---

Tutkovics András

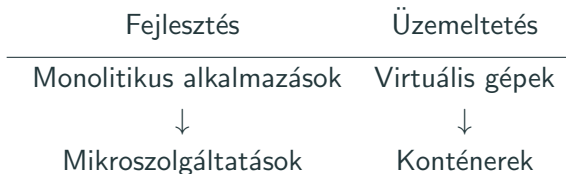
2020.01.08. 14:00

Budapesti Műszaki és Gazdaságtudományi Egyetem

Változó igények, technológiák



Változó igények, technológiák



**Cél:** költségek optimalizálása

(pl: emberi erőforrás (üzemeltető), használt cpu/memória)

- Kubernetes megismerése
  - Irodalomkutatás**, ismeretek bővítése
  - Erőforrás limitálás**, ütemezés
  - Skálázási módszerek** támogatottsága, használata
- Mérési **környezet összeállítása**
- **Vezérlő program** összeállítása
- **Eredmények kiértékelése**
- **Modell keresése** a skálázás becslésére
  - Modell **implementálása, ábrázolás**

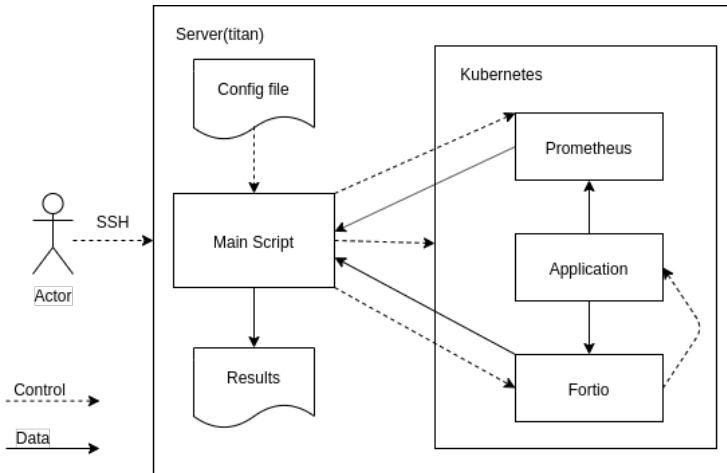
- Konténer orkesztrációs platform  
(konténerek indítása, skálázása, monitorozása, forgalomelosztás...)
- Absztrakciós szintek (objektumok)

Konténer > Pod > Deployment ← Service

- Erőforrások limitálása (ütemezés)

```
- name: nginx-container
  image: nginx:1.17
  imagePullPolicy: IfNotPresent
  resources:
    limits:
      cpu: "200m"
      memory: "64Mi"
    requests:
      cpu: "200m"
      memory: "64Mi"
```

# Környezet összeállítása



## 1. Nginx

- Egyszerű használat → könnyebb tervezés/fejlesztés
- Statikus (*Hello World*) oldal

## 2. NodeJS - Prímszámoló

- Nagy számításigényű alkalmazás
- Változtatható nehézség (*build* után is)
- Eratoszthenész algoritmus

## 3. Apache

- Nagy, statikus webalkalmazás
- ~11 Megabájt szöveges tartalom
- Memóriaigényezés figyeléséhez

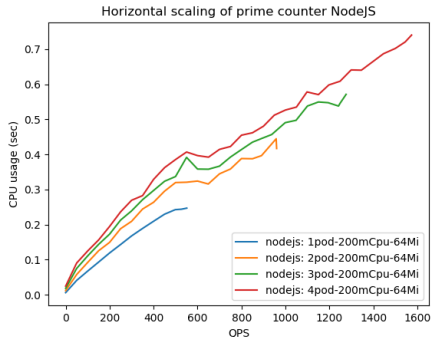
## 4. NodeJS - *Hello World*

- Már meglévő *image*
- Azonos alap, más programmal

# Horizontális skálázás

- Azonos QPS mellett
- Több konténer → magasabb CPU

QPS : 600



#pod	CPU (s)	Válasz (ms)
2	0.32	3.366
4	0.4	1.786

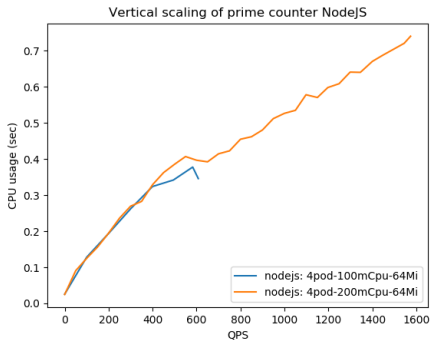
Általában a több egység több erőforrást használ, de van alkalmazás, ami kivételt jelent.



# Vertikális skálázás

- Azonos QPS mellett
- Azonos erőforrás felhasználás
- Max QPS megnő

#pod : 4, QPS : 300



Limit (mCPU)	Válasz (ms)
100	3.489
200	2.014

Semelyik alkalmazás esetében nem jelent (jelentősen) több erőforrás igényt adott terhelés mellett.

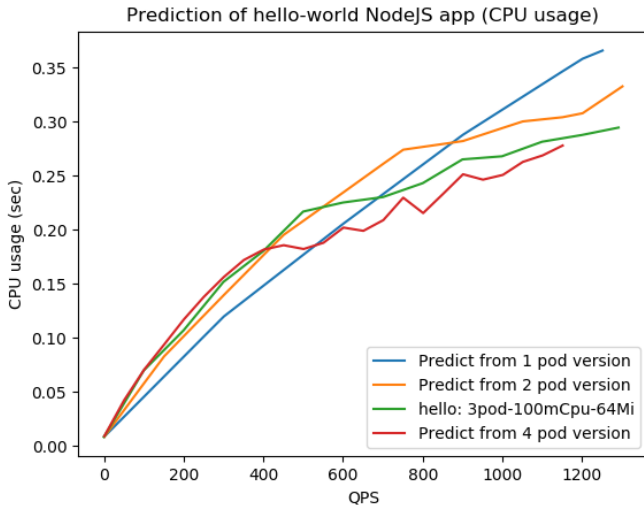
1 pod  $\rightarrow$  általános:

$$f_1(q) : q \in [0; Q] \rightarrow \tilde{f}_k(q) = kf_1\left(\frac{q}{k}\right) : q \in [0; kQ] \quad (1)$$

$\downarrow$

x pod  $\rightarrow$  általános:

$$f_x(q) : q \in [0; Q] \rightarrow \tilde{f}_k(q) = \frac{k}{x} f_x\left(q \frac{x}{k}\right) : q \in [0; \frac{k}{x} Q] \quad (2)$$



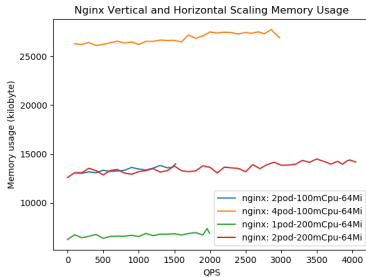
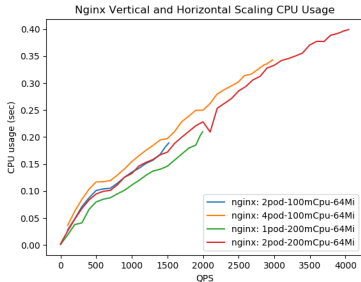
- Horizontális skálázás vagy vertikális skálázás?

<i>Horizontális</i>	<i>Vertikális</i>
K8s alaptól támogatja	K8s nem támogatja alaptól
Több erőforrást használ(hat)	Azonos mennyiségű erőforrás
Nő a rendszer megbízhatósága	Változatlan marad

- Előnyök + hátrányok → tervezői döntés
- Modell
  - Konceptió jó, további kutatást igényel
  - Vertikális skálázásra is lehet keresni

## Bírálóí kérdés (1/2)

Mutassa be egy Ön által választott alkalmazásra a két stratégia különbségét vagy egyezőségét a CPU és memória használat valamint válaszidő paraméterekre.



1pod-200mCPU → 2pod-200mCPU  
0.702ms → 0.635ms

2pod-100mCPU → 4pod-100mCPU  
2.830ms → 1.011ms

## Bírálói kérdés (2/2)

Hogyan lehet az, hogy a 14. és 15. ábrán az elméleti maximum CPU használat felett is vannak mérési eredmények?

