



Önálló laboratórium beszámoló

Távközlési és Médiainformatikai Tanszék

készítette: **Reményi Gergely Márk**

gergo@gergo.city

neptun-kód: **KPZH44**

ágazat: **Mérnökinformatikus szak**

konzulens: **Maliosz Markosz**

markosz@tmit.bme.hu

konzulens: **Simon Csaba**

simon@tmit.bme.hu

Téma címe: Skálázás Kubernetesben egyedi HTTP metrikák alapján

<A címnek nem kell megegyeznie az eredeti téma címével,

Feladat:

A hallgató feladata egyedi metrikák alapján történő horizontális skálázás vizsgálata Kubernetes környezetben. A hallgató a félév során megismerkedik a Horizontal Pod Autoscaler (HPA) technológiával, kiépíti és konfigurálja Kubernetesben az egyedi metrikákat létrehozó csővezeték. TODO

Tanév: 2020/2021. tanév, II. félév

1. A laboratóriumi munka környezetének ismertetése, a munka előzményei és kiindulási állapota

1.1. Bevezető

1.2. Elméleti összefoglaló

2. Mérések elvégzése

2.0.1. Mérésekhez beállított alapértelmezett értékek

A minden mérés elvégzéséhez egyetlen k6 tesztet definiáltam. Ebben a tesztben a kapcsolatot kialakító virtuális felhasználók száma 10 perc alatt lineárisan növekedik 0-tól 100-ig, majd 30 másodperc alatt visszaesik 0-ra. Minden virtuális felhasználó egytized másodpercenként küld el egy HTTP GET kérést.

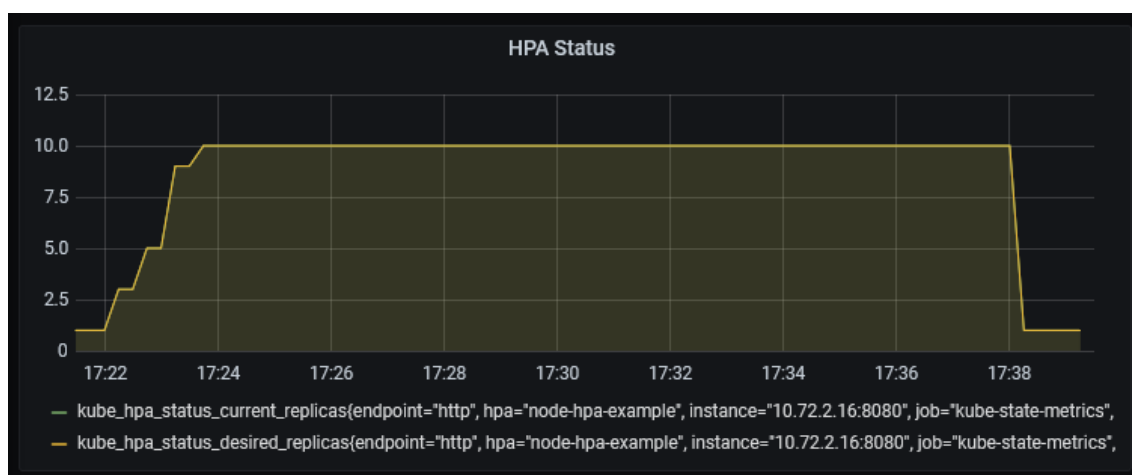
A podokat úgy állítottam be, hogy 300 millicore legyen a limitjük, vagyis a node CPU-jának maximum 3/10-ad számítási kapacitását használhatják. A podoknak még megadtam azt is, hogy csak olyan node-okon indulhatnak el, amelyeken van még 100 millicore-nyi szabad kapacitás.

2.0.2. CPU alapú skálázás rövid válaszidejű végponton

A CPU alapú skálázáshoz úgy állítottam be a HPA-t, hogy 50% alatt maradjon a podok CPU igénye. Ez az 50%-os küszöb a podok CPU limitjének kontextusában értelmezendő, vagyis ha egy pod limitje 300 millicore, akkor a HPA feladata lesz, annyi új podot indítani, hogy a podok átlagos CPU erőforrás igénye 150 millicore alatt maradjon.

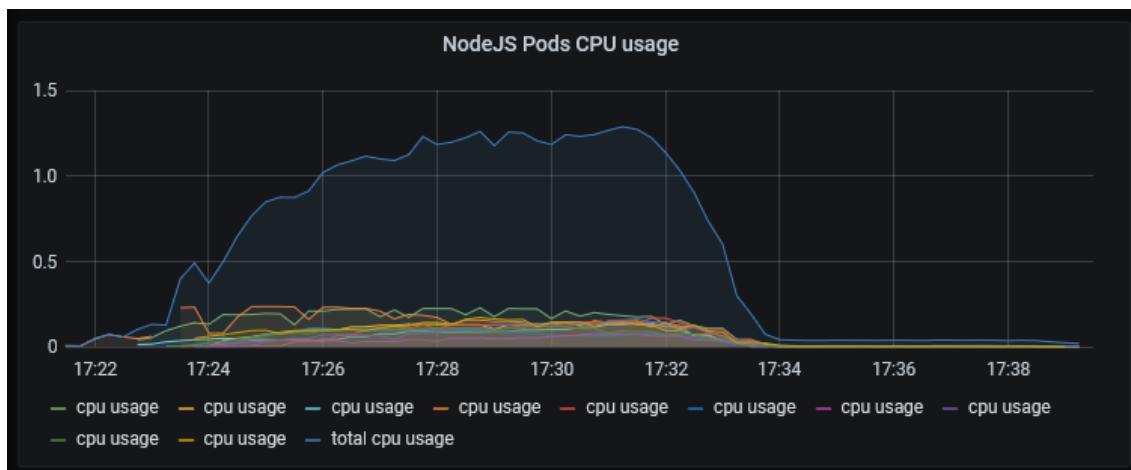
A teszt lefutásához négy diagramot hoztam létre Grafanában, amelyek szükségesek a CPU alapú skálázás vizsgálatához és amellyel össze lehet vetni az egyedi metrikák alapján skálázódó teszt adatait.

Az 1 diagramon láthatóak a HPA által kívánt podok száma (desired replicas) és a podok tényleges száma (current replicas). A diagramon csak egy adatsor látszik, mivel a két metrika együtt mozog. A diagramon látható az is, hogy a HPA nem skálázza le a podokat egyből, hanem kivárja az alapértelmezetten beállított 5 perces lehűlési időt.



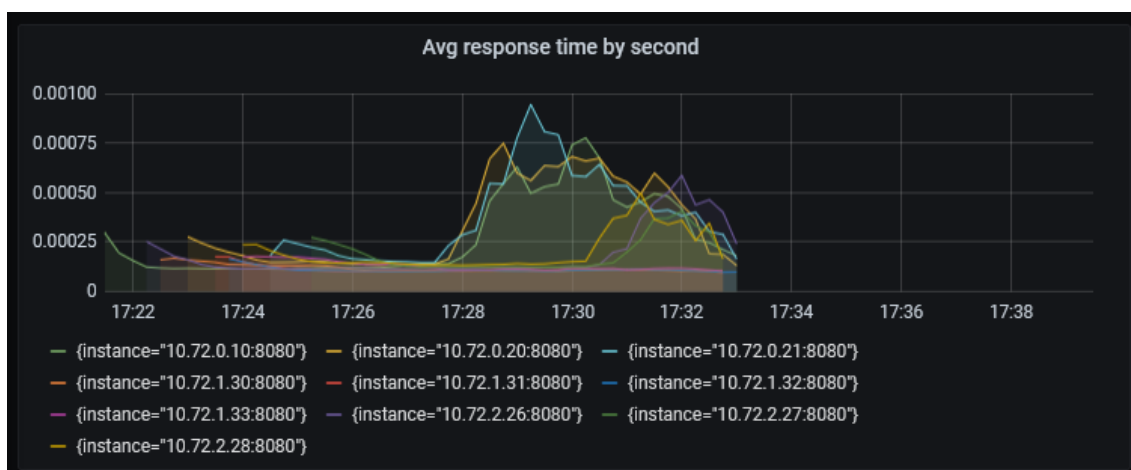
1. ábra. HPA által kívánt és tényleges podok száma

A 2 diagramon látható a skálázott podok CPU kihasználtsága. A kék színű grafikon jelöli az összes pod CPU kihasználtságát, a többi ugyanezt podokra levetítve. Az ábráról leolvasható, hogy a teljes CPU kihasználtság maximuma 1250m. Ezt leosztva a replikák számával, vagyis 10-el, 125m-et kapunk, azaz a HPA jól működött, mert az átlagos CPU kihasználtság nem ment 150m fölé. Azonban egyenként megvizsgálva a podok CPU használatát, vannak olyan podok, amelyek elérik, és sokáig a 300m-es tartományban maradnak és olyanok, amelyek a 100m-es tartományban maradnak. Ebből látható, hogy a Kubernetesben a terheléelosztás nem egyenletesen történik.



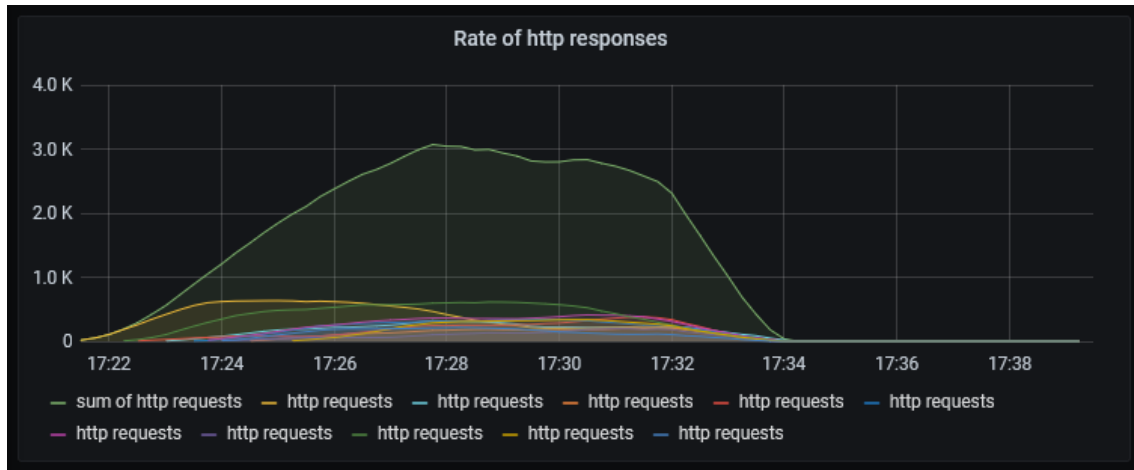
2. ábra. Kiszolgáló podok CPU kihasználtsága

A 3 diagramon az átlagos válaszidő látható podokra bontva. Szinte az összes pod kezdetben magasabb válaszidővel válaszol, ez lehet a NodeJS Express valamilyen sajátossága miatt. A válaszidők a teszt közepe felé egyforma értékeket vesznek fel, majd kettő pod kivételével az összes pod válaszáideje megnő. Ez amiatt lehet, mert a terhelésgenerálás olyan fázisba ért, ahol a kéréseket a podok nem tudják azonnal kiszolgálni.



3. ábra. Kiszolgáló podok válaszáideje 1 másodpercre vetítve

A 4 diagramon az átlagos válaszok száma látható. A zöld színű grafikon jelöli az összes pod átlagos válaszainak számát. Ez a diagram a teszt szerint elvárt módon növekszik a 17:28-as időpontig, majd a válaszok arányában stagnáció áll be. A 3 diagrammal összevetve így arra lehet következtetni, hogy a stagnáció szintén a podokra nehezedő terhelés következménye.



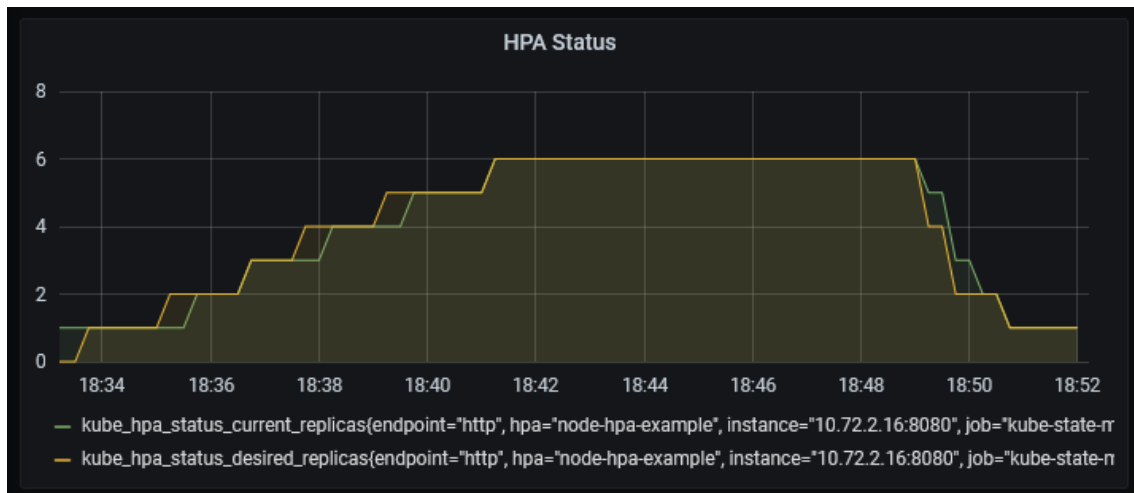
4. ábra. Kiszolgáló podok válaszainak száma 1 másodpercre vetítve

2.0.3. Skálázás az átlagos válaszok számának függvényében

Az egyedi metrikára az átlagos válaszok számát választottam. Az előző mérésből a 4 ábra alapján az átlagos válaszok számának küszöbét a HPA-ban 500-ra állítottam.

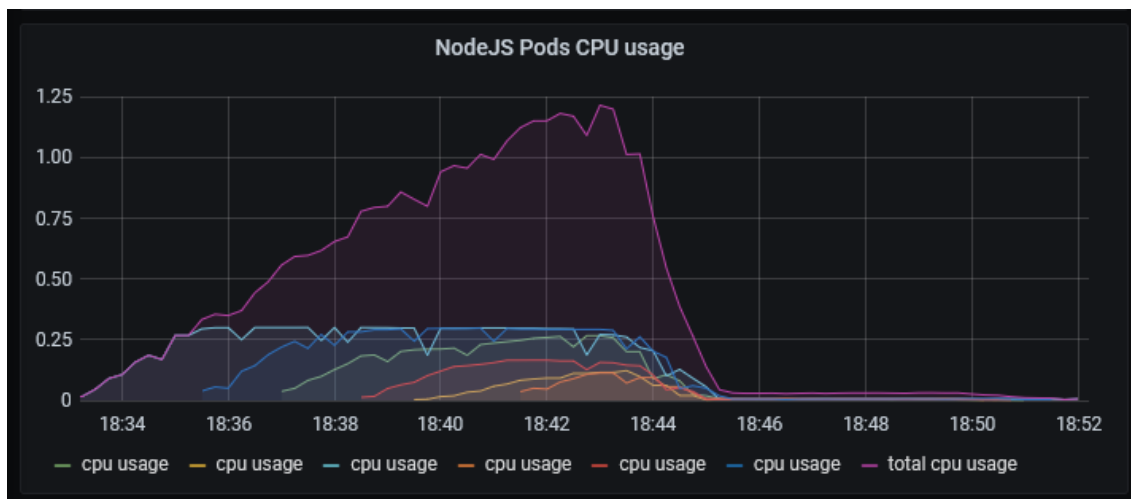
A teszt lefutásához ugyanazokat a diagramokat hoztam létre, mint a CPU alapú skálázásnál a 2.0.2 fejezetben.

Az 5 ábrán a HPA által kívánt és tényleges podok száma látható. Összevetve az 1 ábrával a podok száma sokkal lassabban növekszik, és csak hat podot használ ki a maximális tízből.



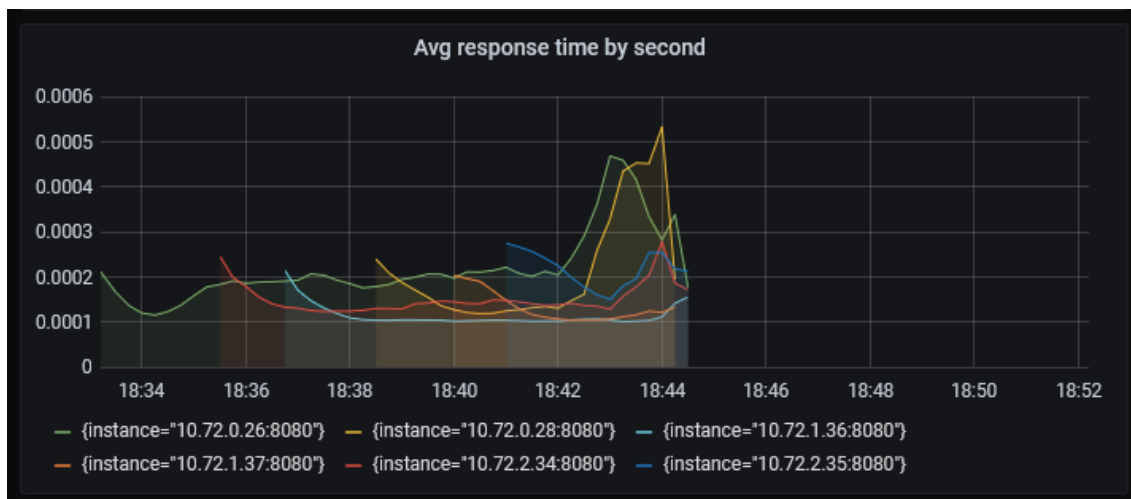
5. ábra. HPA által kívánt és tényleges podok száma

A 6 ábrán látható CPU kihasználtságot összevetve a 2 ábrán látható CPU kihasználtsággal látható, hogy az egyéni metrikákkal történő skálázásnál a CPU kihasználtság sokkal jobban követi a tesztben is definiált lineáris terhelést, valamint a CPU terhelés kisebb, mint a CPU alapú skálázás esetében.



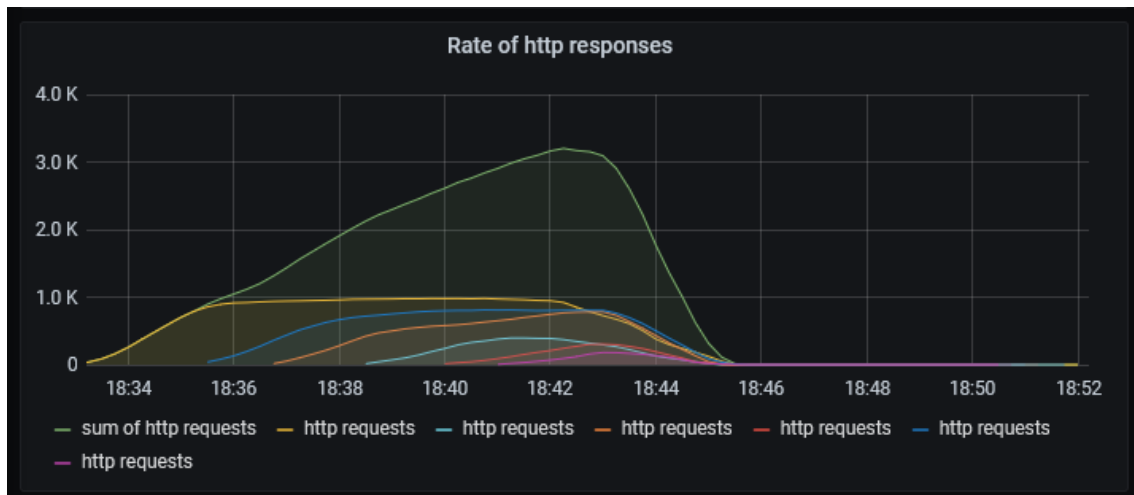
6. ábra. Kiszolgáló podok CPU kihasználtsága

A 7 ábrán látható válaszidőket összevetve a 3 ábrán látottakkal, a válaszidők a teszt elején még nagyjából megegyeznek, a 0.1-0.2 ms tartományba esnek. Az egyéni metrikákkal történő skálázásnál is megnőnek a válaszidők a teszt végére, azonban ez sokkal később történik meg, és csak két podnál emelkedik számottevően.



7. ábra. Kiszolgáló podok válaszáideje 1 másodpercre vetítve

A 8 ábrán is megfigyelhető a 6 ábrához hasonló lineáris növekedés, azaz a válaszok átlagos száma is jól követi a tesztben definiált terhelés növekedését. Azonban itt is megfigyelhető, hogy mivel a HPA a podok válaszainak egy másodpercre vetített számának átlagát veszi, a podok válaszainak száma nem lesz 500 alá szorítva.



8. ábra. Kiszolgáló podok válaszainak száma 1 másodpercre vetítve

2.1. Összefoglalás

A 2.0.2 részben bemutattam egy CPU alapú skálázást és a 2.0.3 részben egy HTTP válaszok átlagos száma alapján történő skálázást. A tesztek futásakor kapott metrikák azt mutatták, hogy a válaszok átlagos száma alapján történő skálázás CPU kihasználtságban és válaszidőben is jobb eredményeket produkált, mint a CPU alapú skálázás. A mérésekből az a következtetés vonható le, hogy a tényleges terhelést leíró metrikákat használva jobb kihasználtságot és jobb metrikus adatokat lehet kapni az erőforrás alapú skálázáshoz képest.

3. Irodalom, és csatlakozó dokumentumok jegyzéke

3.1. A tanulmányozott irodalom jegyzéke

- [1] *Tájékoztató a Műszaki Informatika Szak önálló laboratórium tantárgyainak 2008/9. tanév I. félévi lezárásáról a BME TMIT-en (VITMA367, VITMA380, VITT4353, VITT4330)*, <http://inflab.tmit.bme.hu/08o/lezar.shtml>, szerk.: Németh Felicián, 2008. november 5.
- [2] Wikipedia contributors, *Wikipedia:Academic use*, Wikipedia, The Free Encyclopedia, 2011 Nov 11. Available from:
http://en.wikipedia.org/w/index.php?title=Wikipedia:Academic_use&oldid=460041928

3.2. A csatlakozó dokumentumok jegyzéke