

# Alkalmazkodó terheléselosztás

## Szakedolgozat bemutató

Krajcsovits György

Győr, 2013



# Tartalom

- Bevezetés, kommunikációs hálózatok forgalmának növekedése
- Jelenlegi kiszolgáló rendszerek felépítése
- Előtét-berendezés, terheléselosztás
- A megoldandó gyakorlati probléma
- Megoldási javaslat
  - Mérési környezet megvalósítása
  - Mérések, eredmények

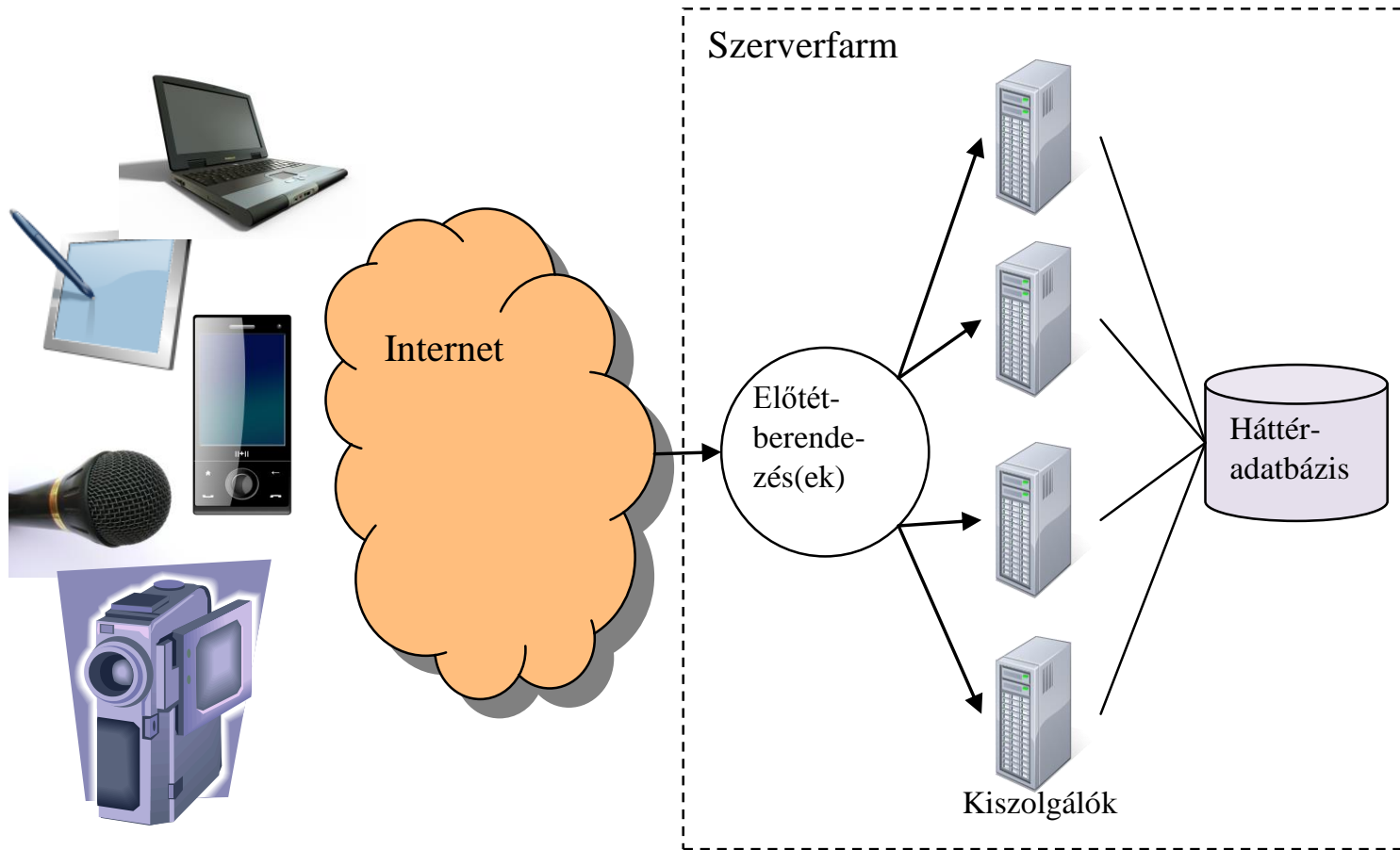
# Bevezetés I.

- A telekommunikációs hálózatok forgalma folyamatosan növekszik
- Új szolgáltatások jelennek meg (mobil fizetés, szociális háló, gép-gép kommunikáció)
- Rengeteg ügyfél, aki elvárja, hogy szinte azonnal működjön a választott szolgáltatás (különösen az olyan, amiért pénzt fizet vagy pénz múlik rajta)

# Bevezetés II.

- Telekommunikációs szolgáltatók és interneten szolgáltatást nyújtók kihívásai
  - Nagy teljesítmény
  - Alacsony válaszidő
  - Folyamatos rendelkezésre állás

# Kiszolgáló rendszerek



# Előtét-berendezés

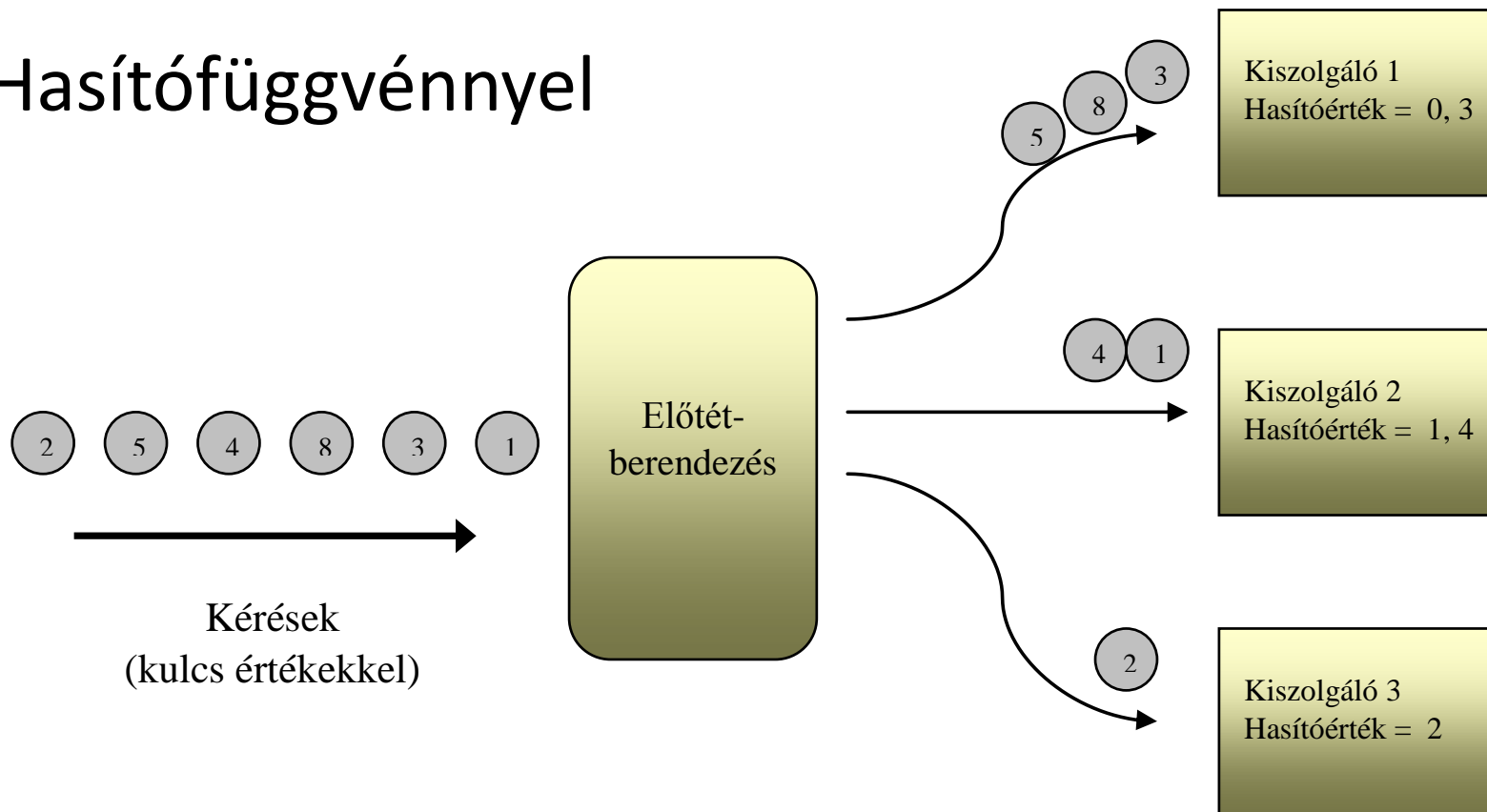
- Feladata, hogy a kliensektől érkező kérések eljussanak a kiszolgálóig
- Egyenletes terhelés biztosítása (terheléselosztás)

# Terheléselosztás I.

- Kérések adatai, rendszer jelenlegi állapota, rendszer múltbéli állapotai alapján
  - Állapottartó
  - Állapot nélküli
  - Statikus
  - Dinamikus

# Terheléselosztás II.

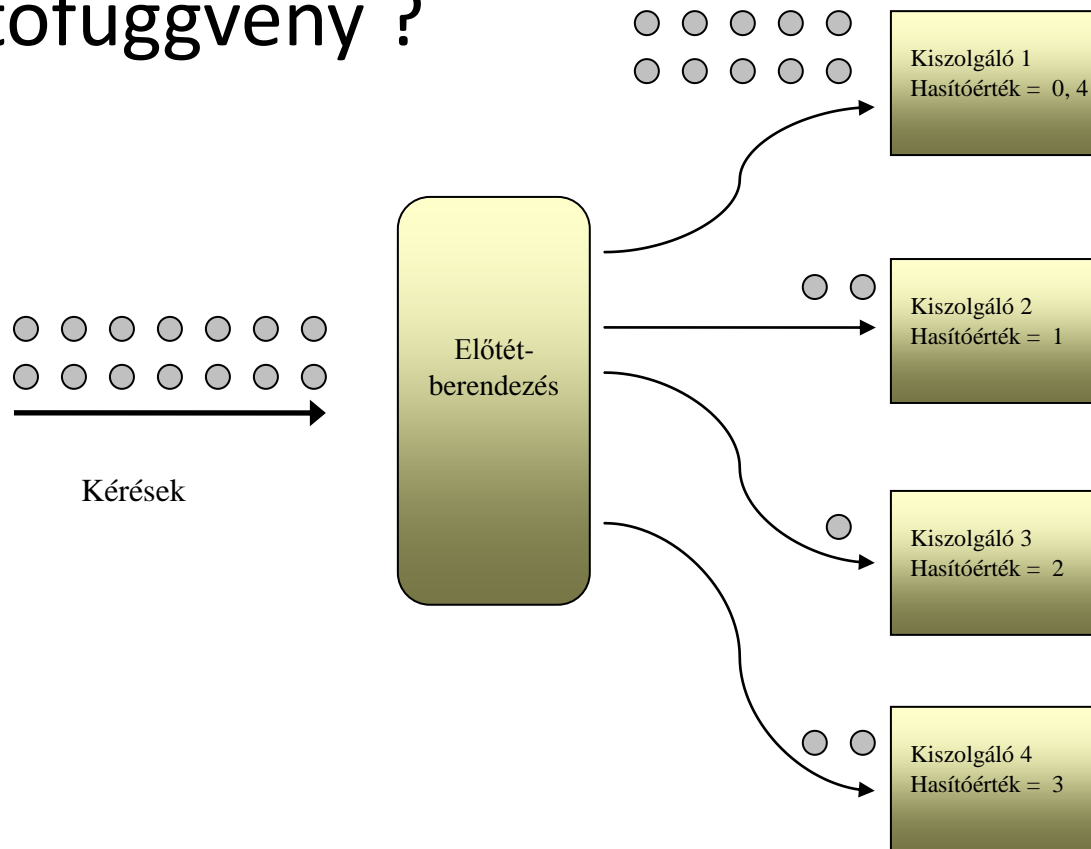
## Hasítófüggvénnyel





# Gyakorlati probléma

- Mit tehetünk, ha nem ad egyenletes eloszlást a hasítófüggvény ?



# Megoldási javaslat

- Az algoritmus mérő részének működése a következő:
  1. Minden kérés továbbításakor hozzáadok egyet a kiválasztott kiszolgáló terheléséhez és a számolt hasítófüggvény érték terheléséhez.
- Ezzel tulajdonképpen meg is van a mérés, hogy melyik kiszolgáló és hasítóérték mennyire terhelődik, használódik.
- A kiegyenlítési algoritmust viszont idő alapon futtatom, a programban minden 10. órajelre.
  1. Megkeresem, hogy melyik a leginkább és melyik a legkevésbé terhelt kiszolgáló, legyenek ezek MAX és MIN.
  2. Ha MAX és MIN azonos, vagy nincs legalább kétszeres különbség a terhelésük közt, akkor lépek 5.-re.
  3. Megkeresem, és terhelés alapján növekvő sorrendbe rakom azokat hasítóértékeket, amelyek a MAX kiszolgálóhoz vannak rendelve.
  4. Átállítom a hasítóértékek célját MIN-re, amíg ezzel javul az egyensúly a MIN és MAX között.
  5. Nullázom a számolt terheléseket.

# Mérési környezet

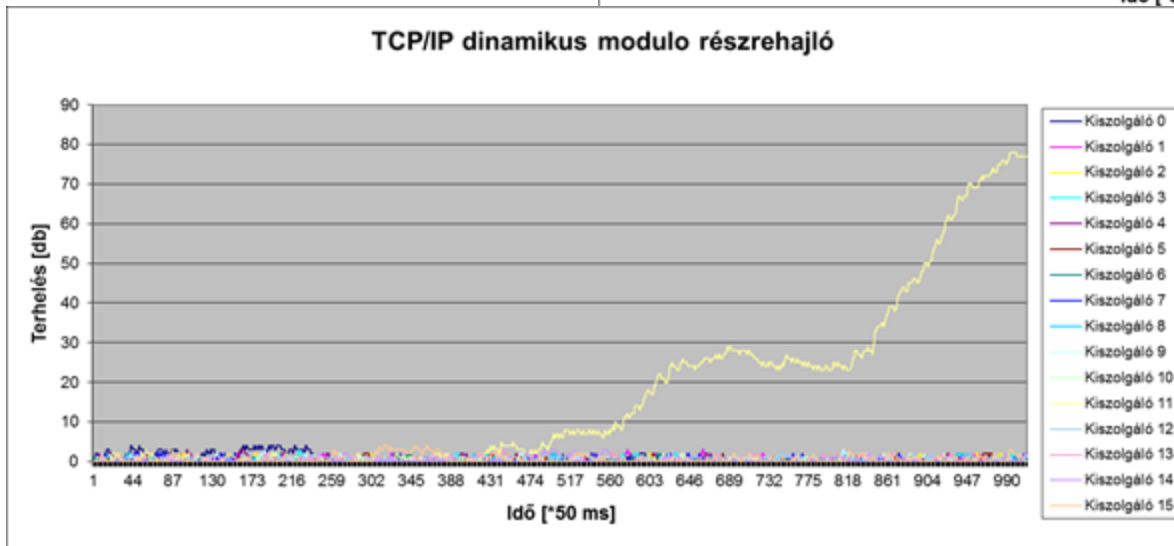
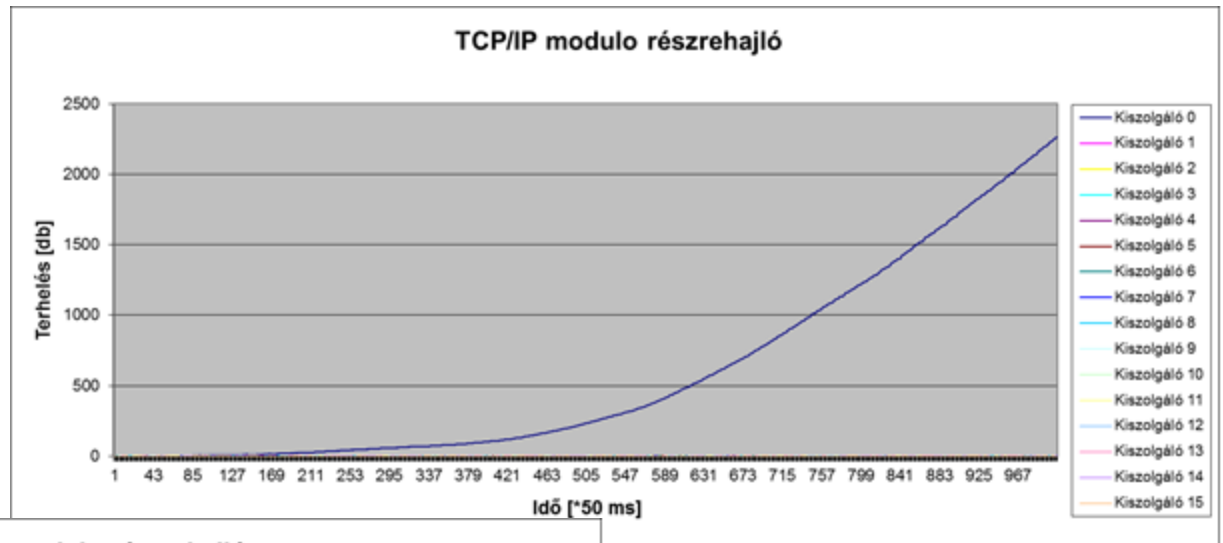
- Linux, C++ programok
- Képes fájlokon működni szimulált diszkrét időben vagy hálózaton (TCP) valós időben
- Alprogramok:
  - Kérések generálása
  - Kérések lejátszása
  - Kérések terheléselosztása
  - Kiszolgáló szimulálása

# Mérési eredmény I.

- Kétféle forgalom mintára:
  - Egyenletes
  - Részrehajló
- Háromféle terheléselosztó algoritmus
  - Round-robin
  - Hasítófüggvény = maradékos osztás
  - Saját algoritmus maradékos osztás felett

# Mérési eredmény II.

Hasítófüggvény,  
2000x terhelés



Hasítófüggvény,  
saját  
algoritmussal,  
80x terhelés