

TOWARDS GREEN COMPUTING IN ERLANG¹

ERLANG PROGRAMOK ENERGIAFOGYASZTÁSÁNAK VIZSGÁLATA

MÉSZÁROS ÁRON ATTILA
NAGY GERGELY



ELTE IK, Programtervező informatikus BSc

Országos Tudományos Diákköri Konferencia

2019. április 17.

Budapest

¹A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg (EFOP-3.6.3-VEKOP-16-2017-00002).

BEVEZETÉS – MOTIVÁCIÓ

➤ Környezettudatosság

- Energiafogyasztás minimalizálása
- Számítógépes eszközöknél is
- Green computing

➤ Miért Erlang?

- Népszerűbb nyelvek – sok kutatás
 - C++, Java, Haskell
- Erlang
 - Széleskörűen használt, iparban
 - Még nem volt ilyen kutatás
- RefactorErl



BEVEZETÉS – EREDMÉNYEK

- Eszköz energiafogyasztás mérésére
 - RAPL, rapl-read.c
 - Erlang keretrendszer
 - Python megjelenítő
- Magasabb rendű függvények hatása
- Lista vagy tömb
- Párhuzamosítás
 - Process pool
 - Token ring
- 2 konferencián előadás,
egy megjelent folyóiratcikk,
egy cikk elbírálás alatt

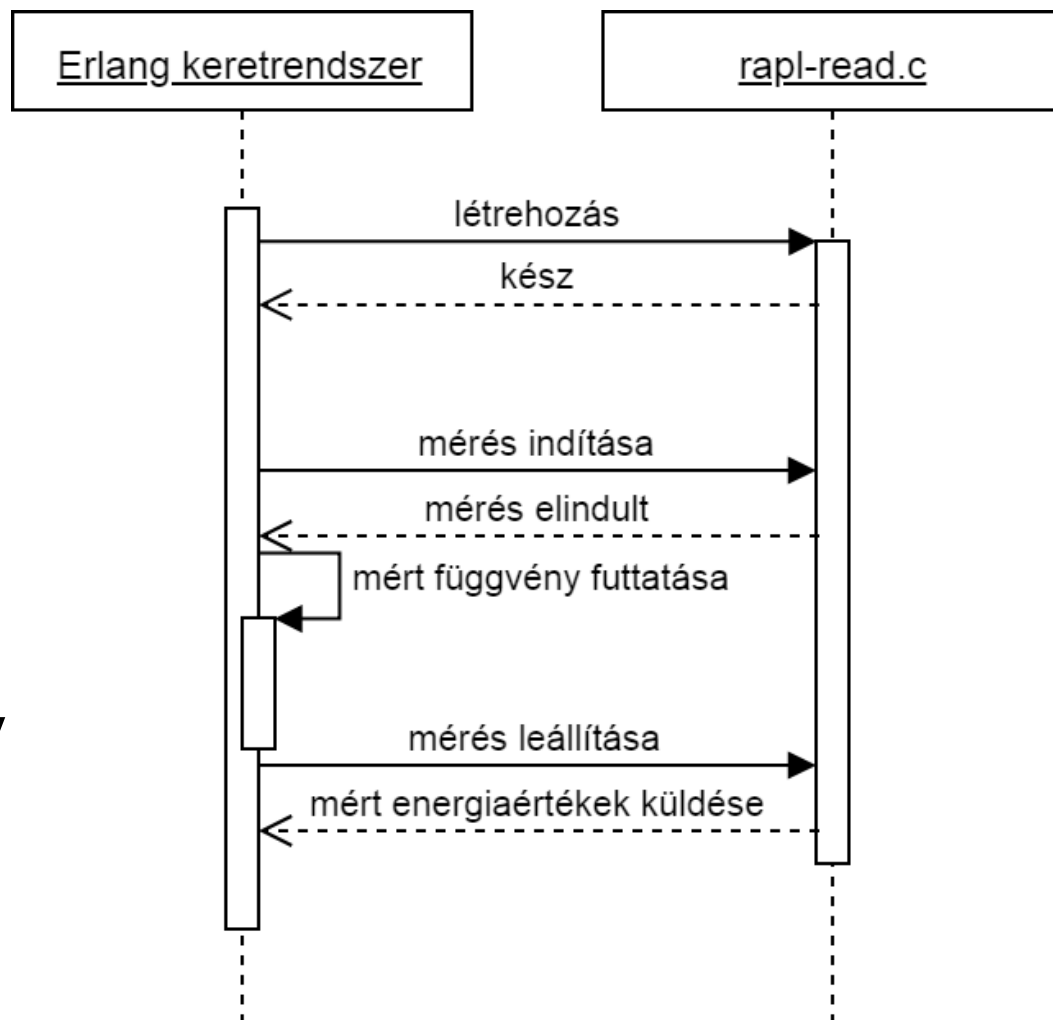
MÓDSZERTAN – MÉRÉS

➤ Running Average Power Limit (RAPL)

- Intel
- rapl-read.c
- 3 féle módszer
- 4 mérhető domain

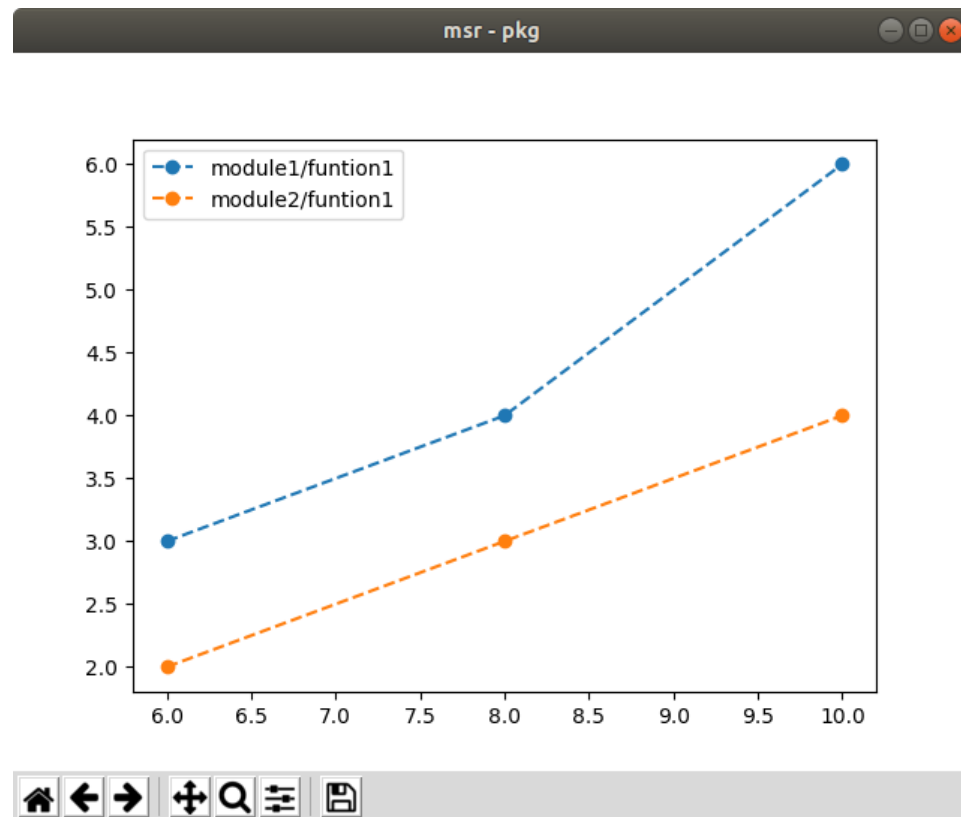
➤ Erlang keretrendszer

- Kommunikáció
- Mérni kívánt függvény futtatása
- Mérési eredmények összegyűjtése

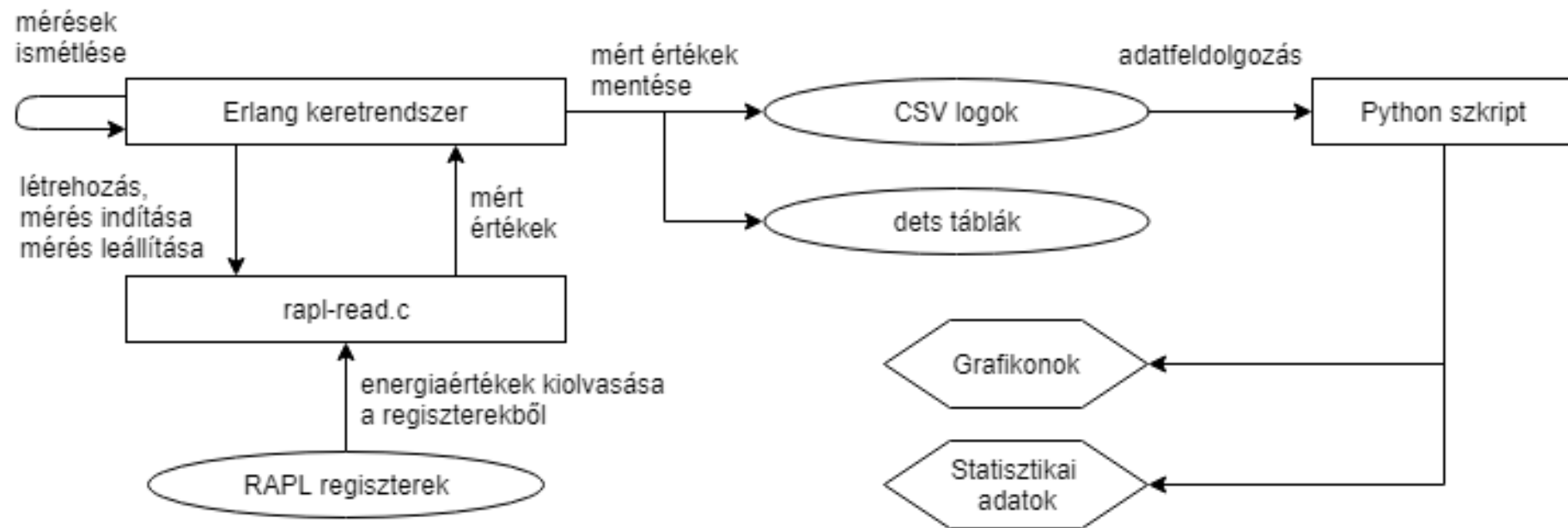


MÓDSZERTAN - ADATFELDOLGOZÁS

- Mérési módszertan
 - 10 db mérés átlaga
 - Minimális és maximális értékek eldobása
 - Futási idő mérése
- Vizualizáció
 - Python – matplotlib
- Felvett teljesítmény számítása
- Korreláció számítása energia és idő között
- Egyéb statisztikák



MÓDSZERTAN - ÖSSZEFOGLALÁS



MÉRÉSI SZEMPONTOK

- Magasabb rendű függvények
 - map, foldr, foldl, filter, stb.
- Különböző adatszerkezetek
 - Lista
 - Tömb
- Párhuzamosítás
 - Brute force (naiv)
 - Process pool
 - Folyamatok száma
 - Üzenetküldés költsége
 - Magok száma

MÉRT ALGORITMUSOK – N KIRÁLYNŐ

➤ 5 féle megoldás

➤ Lista

- Magasabb rendű függvények megléte vagy eliminálása
- Naiv párhuzamos verzió

➤ Tömb

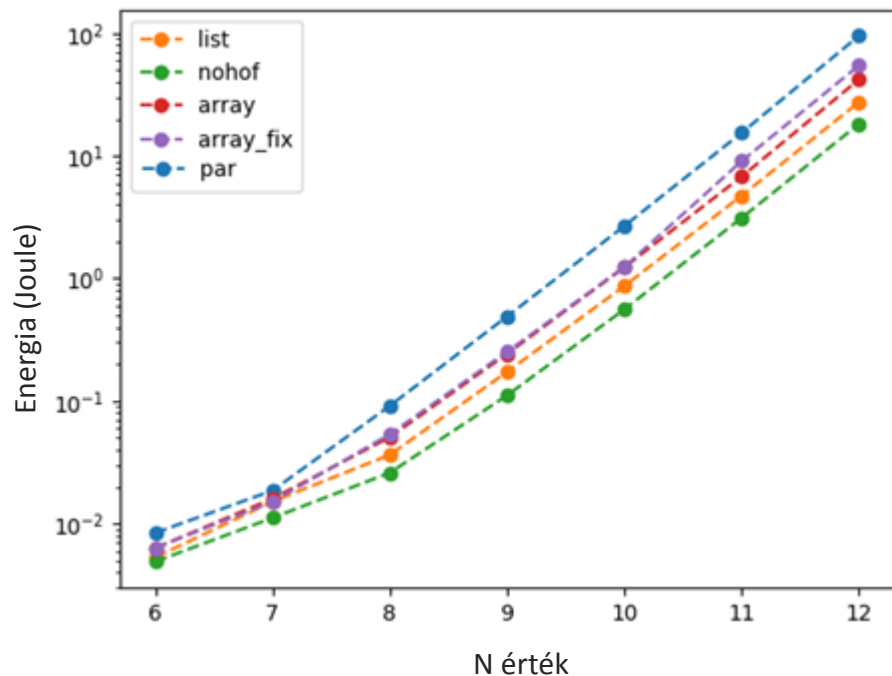
- Fix vagy növelhető méret

```
par_map(F, Xs) ->  
  Me = self(),  
  [spawn(fun() -> Me ! F(X) end) || X<-Xs],  
  [receive Res -> Res end || _ <- Xs].
```

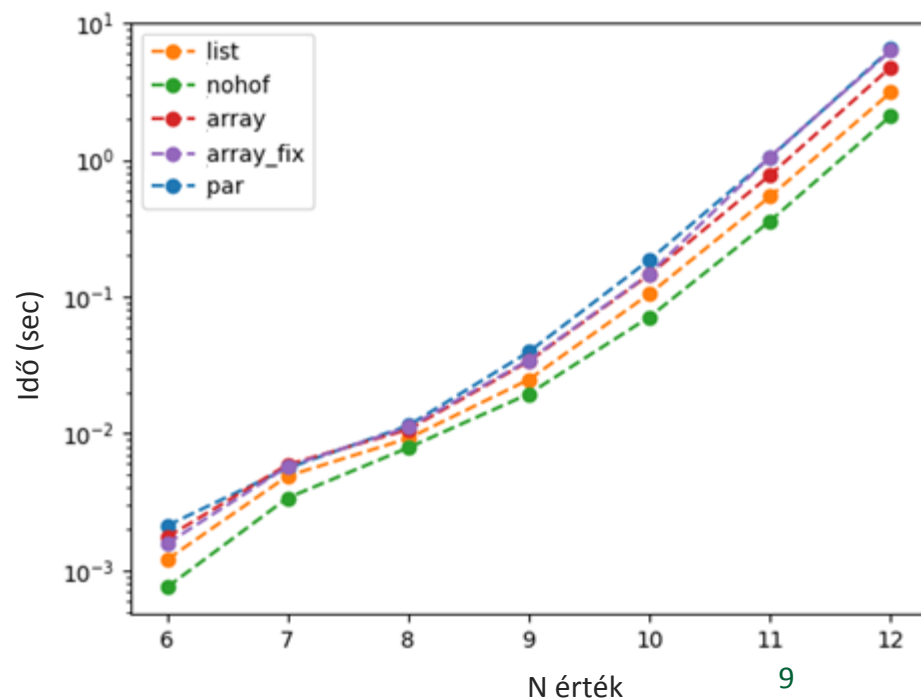
```
lists:flatmap(  
  fun(Qs) -> solve_list(N, Row+1, Qs) end,  
  [[{Col, Row} | Queens] || Col <- lists:seq(1, N), legal_list({Col, Row}, Queens)]  
).
```

```
flatmap_nohof(Queens, N, Row) -> flatmap_nohof(Queens, [], N, Row).  
flatmap_nohof([], R, _, _) -> R;  
flatmap_nohof([H|T], R, N, Row) ->  
  flatmap_nohof(T, solve_nohof(N, Row+1, H) ++ R, N, Row).
```


EREDMÉNYEK – N KIRÁLYNŐ



- Magasabb rendű függvény nélkül hatékonyabb
- Tömbök kevésbé hatékonyak
- Naiv párhuzamosítás rossz



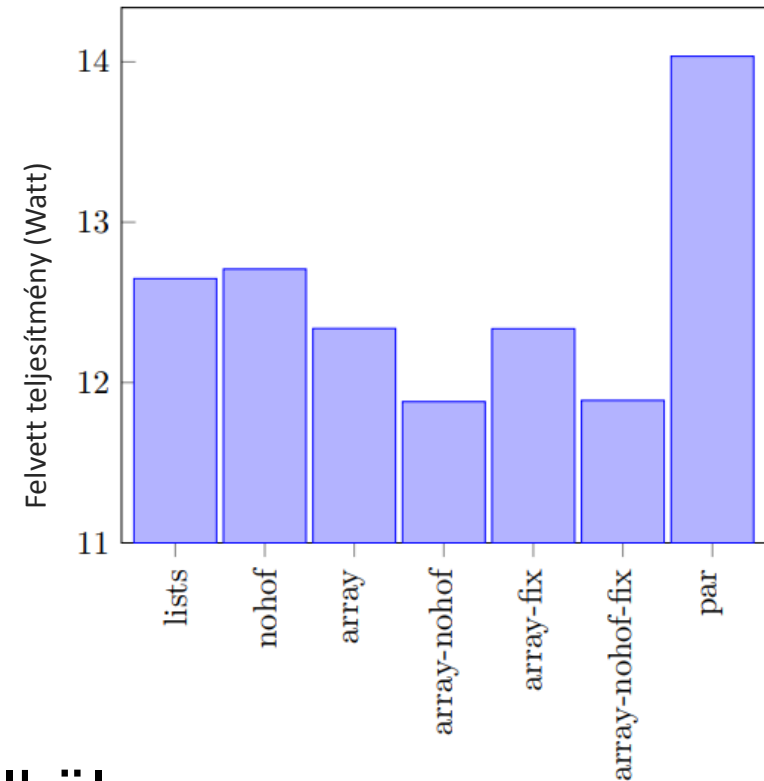
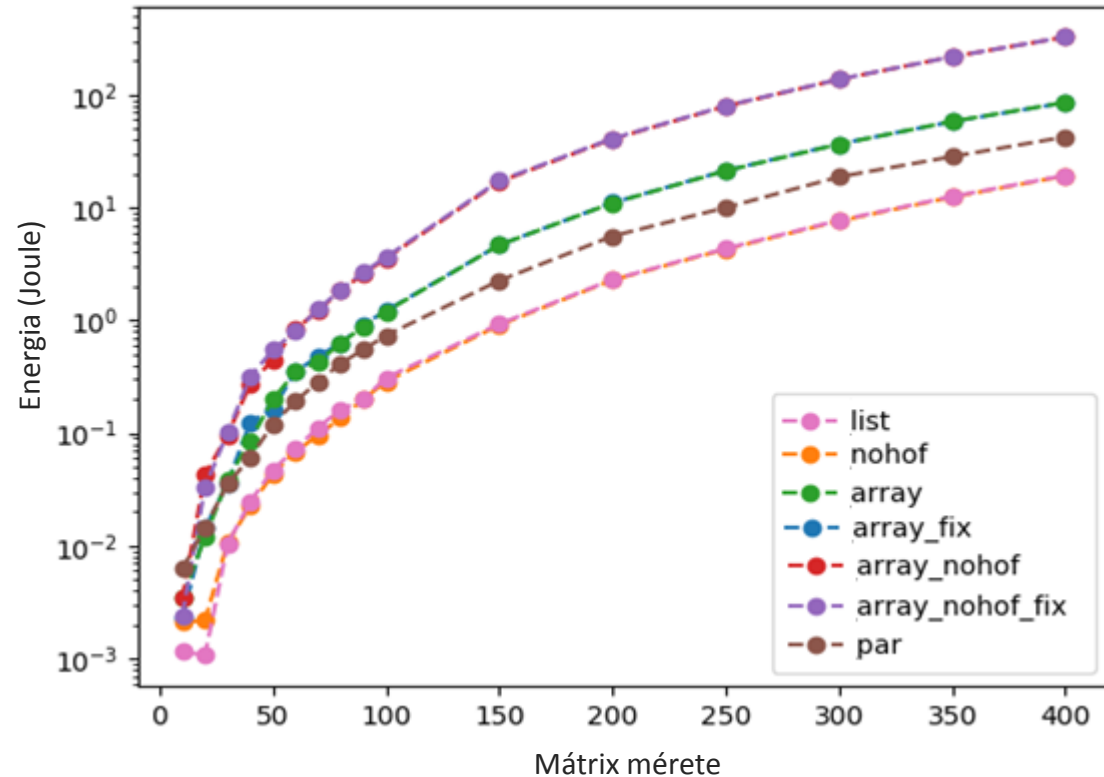
MÉRT ALGORITMUSOK – RITKA MÁTRIX SZORZÁS

- Többféle megoldás, hasonlóan az N királynőhöz
- Magasabb rendű függvények használata tömböknél is
- Különbözik a reprezentáció tömbök és listák esetén

```
mxm_array(Rows, Cols) ->  
  array:sparse_map(fun(_, Col) ->  
    if Col == undefined -> undefined;  
    true -> mxv_array(Rows, Col) end end, Cols) .
```

```
vxv_array_map(Index, Size, _, Row) when Index == Size -> Row;  
vxv_array_map(Index, Size, Col, Row) ->  
  ElemR = array:get(Index, Row),  
  ElemC = array:get(Index, Col),  
  if ElemR == undefined -> vxv_array_map(Index+1, Size, Col, Row);  
  ElemC == undefined -> vxv_array_map(Index+1, Size, Col, array:set(Index,  
    undefined, Row));  
  true -> vxv_array_map(Index+1, Size, Col, array:set(Index, ElemC*ElemR, Row)  
  )  
end.
```

EREDMÉNYEK – RITKA MÁTRIX SZORZÁS

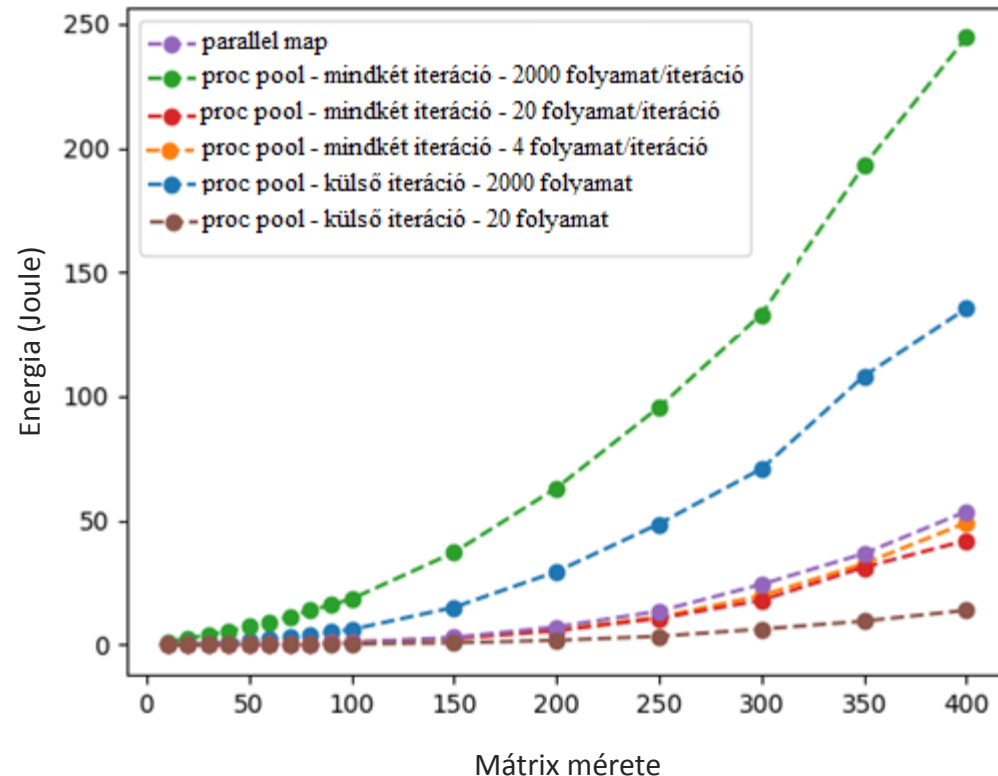


- Magasabb rendű függvények nélkül
 - Listánál javulás
 - Tömbnél romlás
- Nem érdemes tömböt használni, ha sok az ismeretlen elem

PÁRHUZAMOSÍTÁSI MÓDSZEREK

- Ritka mátrix szorzás párhuzamosítása
- Cél: hogyan hat a folyamatok száma az energiafogyasztásra?
- Brute force (naiv) módszer
 - Ahol map van, ott párhuzamossá tesszük
- Process poolok
 - dispatcher, worker, collector
- Csak a program külső iterációinak párhuzamosítása
 - Kevesebb folyamat jön létre
 - Egy folyamat több számítást végez

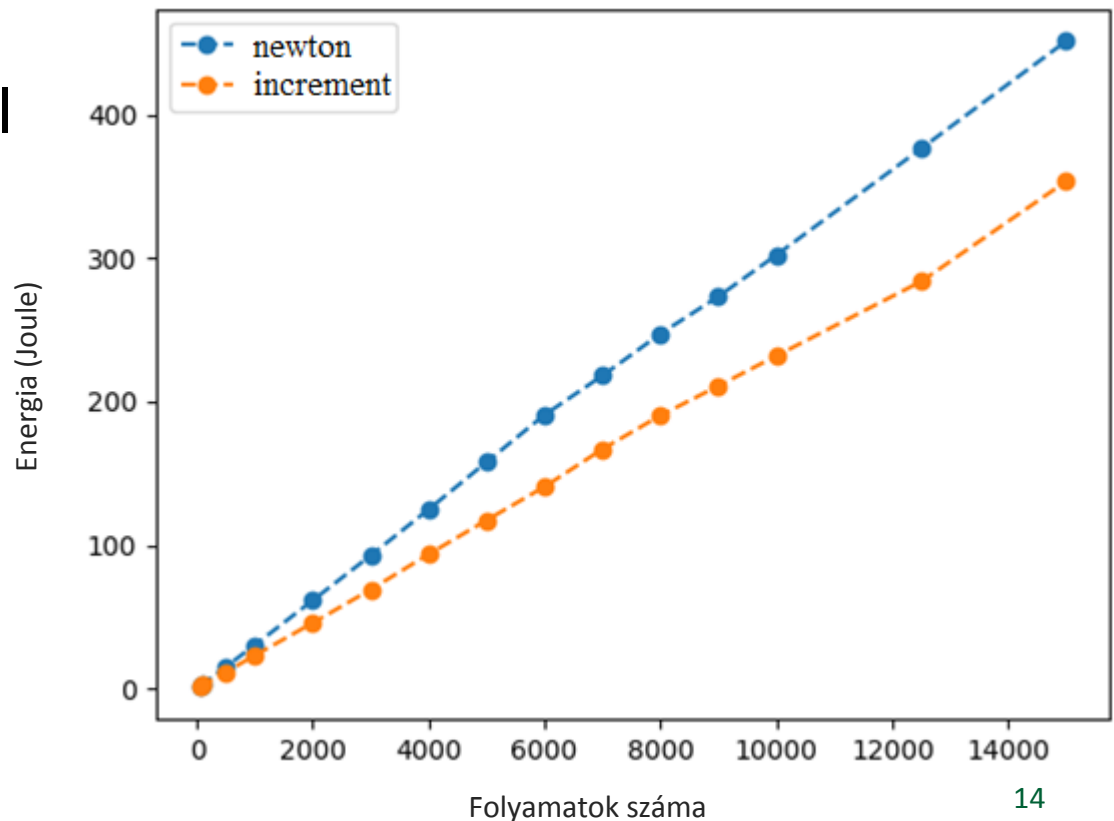
PÁRHUZAMOSÍTÁSI MÓDSZEREK



- Ha sok a folyamat sok energiát fogyaszt
- Legjobb módszer:
 - Kevés folyamat él egyszerre
 - Kevés létrehozás
 - A folyamatok száma és a magok száma közel van egymáshoz

TOKEN RING

- Cél: mi az üzenetküldés költsége?
- Körbe küldjük a folyamatokon az üzenetet
- Minimális a többi számítás
 - Inkrementálás
 - Newton módszerrel gyökvonás



ÖSSZEFOGLALÁS

- Erlang programok energiafogyasztásának mérése
- Komplex algoritmusok
 - N királynő
 - Ritka mátrix szorzás
- Magasabb rendű függvények vizsgálata
- Különböző adatszerkezetek használata
 - Lista vagy tömb
- Párhuzamosítási módszerek elemzése
 - Folyamatok számát hatása
 - Üzenetküldések vizsgálata

KONKLÚZIÓ

- Eszköz Erlang programok energiafogyasztásának mérésére
- Magasabb rendű függvények eliminálása csökkenti az energiafogyasztást
- Lista vagy tömb
 - A vizsgált problémákra a listák hatékonyabbnak bizonyultak
 - A sok ismeretlen elem ront a tömbökön
- Párhuzamosítás
 - Folyamatok számát érdemes limitálni
 - Logikai magok számához közeli limit
 - Extra üzeneteket küldeni olcsóbb, mint a sok folyamat miatti kontextusváltások költsége

EREDMÉNYEK

- Megjelent folyóiratcikk
 - Mészáros, Á. A. & Nagy, G. (2018). Towards Green Computing in Erlang. *Studia Universitatis Babeş-Bolyai Informatica*, 63(1), pp. 64-79.
- Két konferencián előadás
 - 12th Joint Conference on Mathematics and Computer Science, 2018
 - Conference on Software Technology and Cyber Security, 2019
- Egy cikk elbírálás alatt
 - *Acta Cybernetica*

TOVÁBBI EREDMÉNYEK

- Alapvető, elemi nyelvi konstrukciók vizsgálata
 - Lista, map, dictionary adatszerkezetek vizsgálata
- Magasabb rendű függvények hatásának megerősítése
- Különféle párhuzamosítási tervezési minták vizsgálata
 - Pl.: taskfarm
- Energiafogyasztást csökkentő refaktorálások implementálása

KÖSZÖNJÜK A FIGYELMET!

- Megjelent folyóiratcikk
 - Mészáros, Á. A. & Nagy, G. (2018). Towards Green Computing in Erlang. *Studia Universitatis Babeş-Bolyai Informatica*, 63(1), pp. 64-79.
- Két konferencián előadás
 - 12th Joint Conference on Mathematics and Computer Science, 2018
 - Conference on Software Technology and Cyber Security, 2019
- Egy cikk elbírálás alatt
 - *Acta Cybernetica*

KÉRDÉSEK

Lehetőség van Inteltől eltérő architektúrán mérni az energiafogyasztást?

- Legújabb rapl-read (2018 szeptember) támogatja az újabb AMD Ryzen processzorok mérését is
- Tervezzük a mérések megismétlését más architektúrákon is

KÉRDÉSEK

Milyen hatása van az energiafogyasztásra más fordítóprogramok vagy optimalizálást segítő kapcsolók használatának? Léteznek ilyenek?

- Használt fordító: BEAM
 - bájt kód
- Alapvető optimalizálások vannak, de kapcsoló hozzá nincs
- HiPe fordító
 - natív kódot fordít
 - Hatása:
 - A relatív viszonyok megmaradnak
 - A különbségek mértéke változik

A refaktorálások hogyan befolyásolják a kód méretét, karbantarthatóságát?

- Magasabb rendű függvények eliminációja
 - Sok extra kód, átláthatatlan
 - Csak a végleges fordítás előtt lenne érdemes
 - A kódbázisban marad az olvasható, karbantartható kód
- Adatszerkezet váltás
 - Pl.: proplist → map
 - Megmarad a karbantartható kód
 - Csak hívások lesznek lecserélve

KÉRDÉSEK

Mekkora a mérések átlaga, szórása, terjedelme?

Az N királynő probléma egy megoldásának statisztikai jellemzői

Bemenet	Átlag	Szórás	Medián	Min	Max	Terjedelem
6	0.0041	0.00072	0.0039	0.0034	0.0052	0.0019
7	0.012	0.0016	0.011	0.0088	0.014	0.0049
8	0.031	0.0023	0.031	0.028	0.034	0.0062
9	0.16	0.0059	0.16	0.15	0.17	0.019
10	0.80	0.0099	0.80	0.79	0.82	0.035
11	4.42	0.016	4.43	4.40	4.44	0.047
12	25.50	0.16	25.46	25.28	25.73	0.45

A táblázatban szereplő adatok Jouleban vannak megadva

KÉRDÉSEK

A funkcionális programozás világában vannak-e próbálkozások a computation offloading kérdéskörrel kapcsolatban?

- CloudI /klaʊdi/

- open source, cloud computing platform, Erlangban írva

- Energiafogyasztás szempontjából

- Költségmodell

- Lehet hogy a szerveren jobban lehet párhuzamosítani, így megéri üzenetet küldeni

KÉRDÉSEK

Lehet-e bármilyen formálisabb becslést adni annál, hogy kimérjük az algoritmust?

- Az energiafogyasztás nagyban függhet a vizsgált architektúrától
- Formális becsléshez pontosan ismerni kell az architektúrát
- Inkább érdemes empirikus módszerekkel költségmodellt felállítani

KÖSZÖNJÜK A FIGYELMET!

- Megjelent folyóiratcikk
 - Mészáros, Á. A. & Nagy, G. (2018). Towards Green Computing in Erlang. *Studia Universitatis Babeş-Bolyai Informatica*, 63(1), pp. 64-79.
- Két konferencián előadás
 - 12th Joint Conference on Mathematics and Computer Science, 2018
 - Conference on Software Technology and Cyber Security, 2019
- Egy cikk elbírálás alatt
 - *Acta Cybernetica*