N-teles

UROŠ LOTRIČ

Uvod

Fizika, kemija

Opis problema

- Gravitacija
- Sila telesa j na telo i
  - $\vec{r}_{ij} = (x_{ij}, y_{ij}, z_{ij}) = (x_j - x_i, y_j - y_i, z_j - z_i)$
  - $r_{ij}^2 = x_{ij}^2 + y_{ij}^2 + z_{ij}^2$
  - $\overrightarrow{F_{ij}} = G\frac{m_i m_j}{r_{ij}^2}\frac{\vec{r}_{ij}}{r_{ij}} = (F_{ijx}, F_{ijy}, F_{ijz}) = (G\frac{m_i m_j}{r_{ij}^2}\frac{x_{ij}}{r_{ij}}, G\frac{m_i m_j}{r_{ij}^2}\frac{y_{ij}}{r_{ij}} G\frac{m_i m_j}{r_{ij}^2}\frac{z_{ij}}{r_{ij}})$
- Sila vseh teles na telo i
  - $F_{ix} = Gm_i\sum_{j\neq i}\frac{m_j}{r_{ij}^2}\frac{x_{ij}}{r_{ij}}$, $F_{iy} = Gm_i\sum_{j\neq i}\frac{m_j}{r_{ij}^2}\frac{y_{ij}}{r_{ij}}$, $F_{iz} = Gm_i\sum_{j\neq i}\frac{m_j}{r_{ij}^2}\frac{z_{ij}}{r_{ij}}$
  - $F_{ix} = Gm_i\sum_{j}\frac{m_j}{R_{ij}^2}\frac{x_{ij}}{R_{ij}}$, $F_{iy} = Gm_i\sum_{j}\frac{m_j}{R_{ij}^2}\frac{y_{ij}}{R_{ij}}$, $F_{iz} = Gm_i\sum_{j}\frac{m_j}{R_{ij}^2}\frac{z_{ij}}{R_{ij}}$
  - Z glajenjem $R_{ij}^2 = r_{ij}^2 + \varepsilon^2$ se izognemo deljenju z 0 pri $i = j$

Uvod

Opis problema

- Posodobitev položaja in hitrosti telesa i
  - $a_{ix} = \frac{F_{ix}}{m_i}$, $a_{iy} = \frac{F_{iy}}{m_i}$, $a_{iz} = \frac{F_{iz}}{m_i}$
  - $a_{ix} = G\sum_j \frac{m_j}{R_{ij}^2}\frac{x_{ij}}{R_{ij}}$, $a_{iy} = G\sum_j \frac{m_j}{R_{ij}^2}\frac{y_{ij}}{R_{ij}}$, $a_{iz} = G\sum_j \frac{m_j}{R_{ij}^2}\frac{z_{ij}}{R_{ij}}$
  - $s_{ix} = s_{ix} + v_{ix}\Delta t + \frac{1}{2}a_{ix}\Delta t^2$, $s_{iy} = s_{iy} + v_{iy}\Delta t + \frac{1}{2}a_{iy}\Delta t^2$, $s_{iz} = s_{iz} + v_{iz}\Delta t + \frac{1}{2}a_{iz}\Delta t^2$
  - $v_{ix} = v_{ix} + a_{ix}\Delta t$, $v_{iy} = v_{iy} + a_{iy}\Delta t$, $v_{iz} = v_{iz} + a_{iz}\Delta t$

Vzporejanje

Delitev teles

- N teles x N izračunov sil: $t_s(N) = \chi N^2 = O(N^2)$
- Vzporejanje: $t_p(N, P) = O\left(\frac{N^2}{P}\right) + \kappa(N, P)$ pri P nalogah

Delitev računanja sil

- Prepolovimo število izračunov, saj velja $F_{ij} = -F_{ji}$
- Bolj zapletena delitev dela

Poskus

Uporabljene tehnologije

- OpenMP
- OpenCL
- OpenMPI

Merjenje časa

- Na časovni korak
- Prenašanje podatkov na začetku in na koncu

Konfiguracije

- 8, 16, 32, …, 65536 teles
- 5 meritev pri enaki konfiguraciji

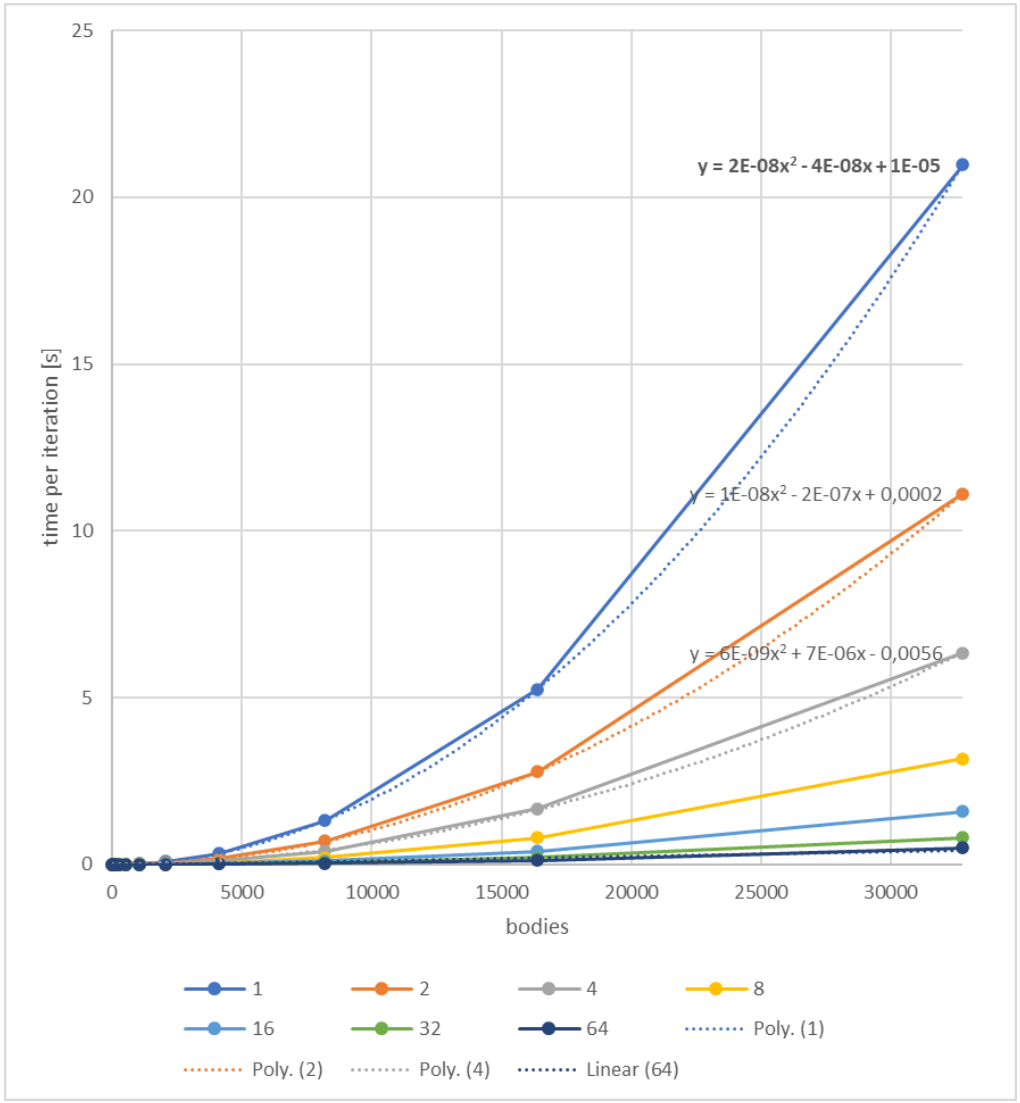
OpenMP

Koda

- nbody-mp.c

Jedra

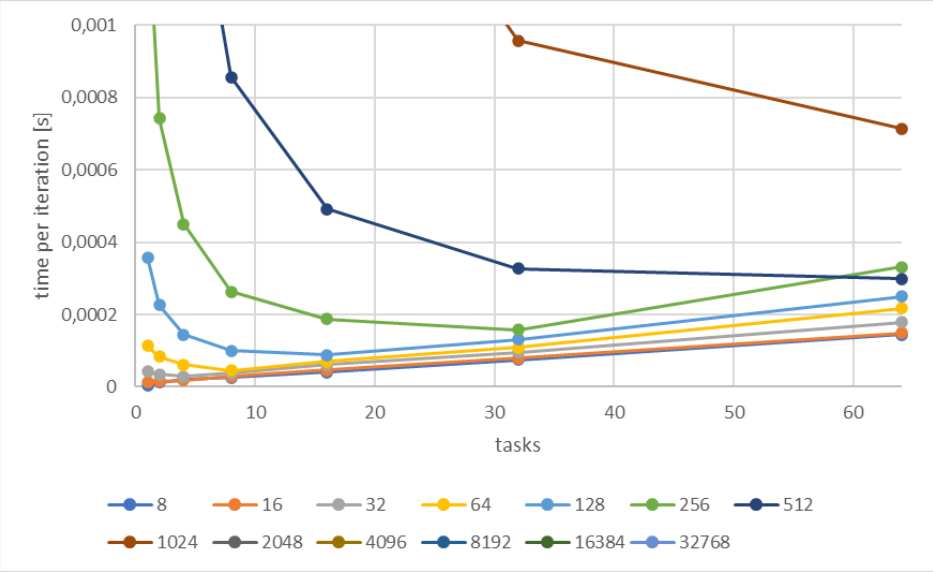
- 1, 2, 4, 8, … 64
- Pričakovana kvadratična odvisnost
  - računanje sile kvadratična odvisnost
  - posodobitev položajev in hitrosti linearna odvisnost
- Čas za korak
  - 1 nit: ni stroškov upravljanja z nitmi
  - $\chi_{MP} = 2 \cdot 10^{-8} s$

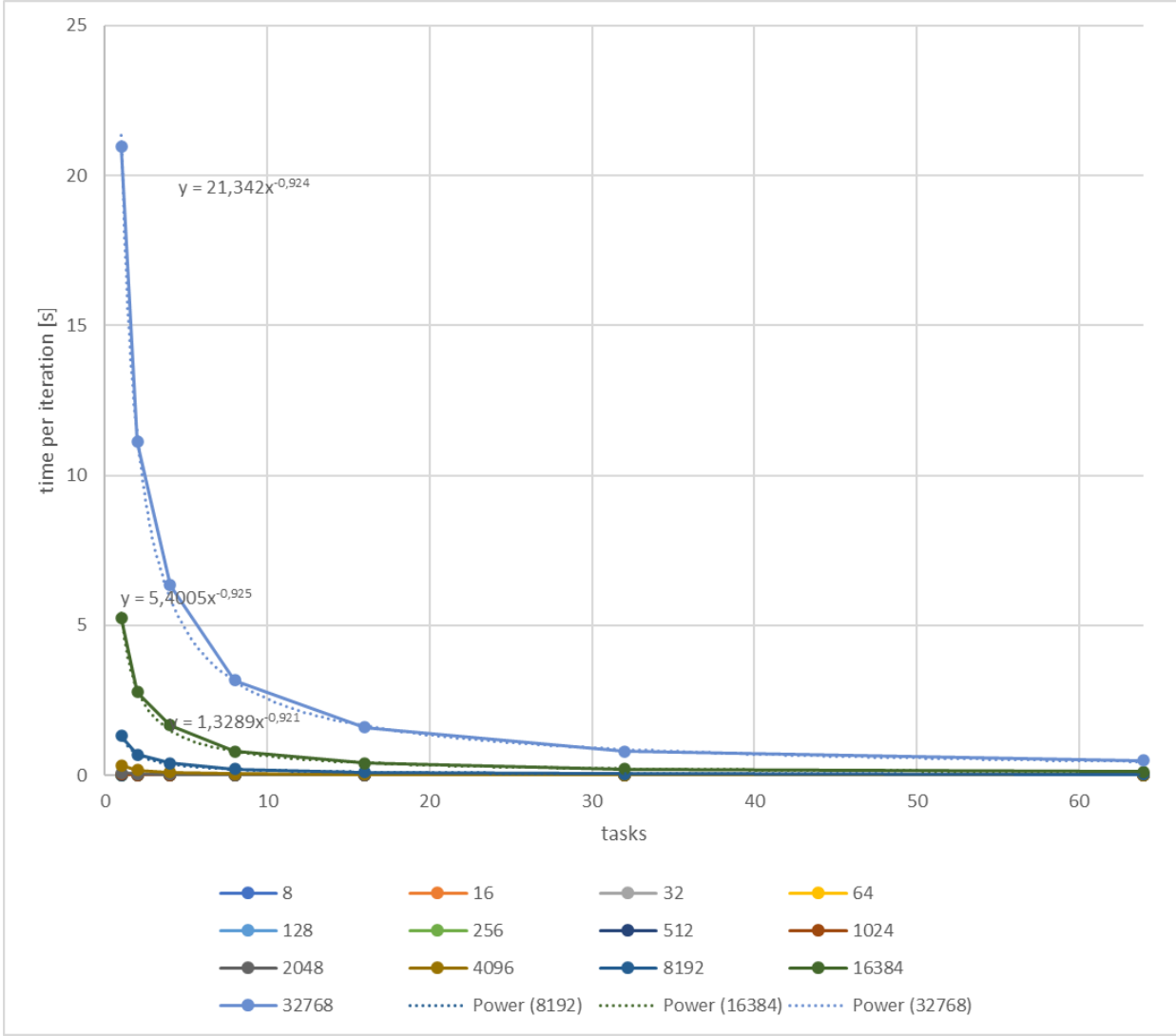


OpenMP

Jedra

- Čas z večanjem števila jeder pada
  - Teorija: -1, poskus $-0{,}925$
  - Pri teoretični analizi nismo upoštevali dela z nitmi (ustvarjanje, pregrade, …)
  - Pri majhnem številu teles nam veliko niti/jeder ne koristi

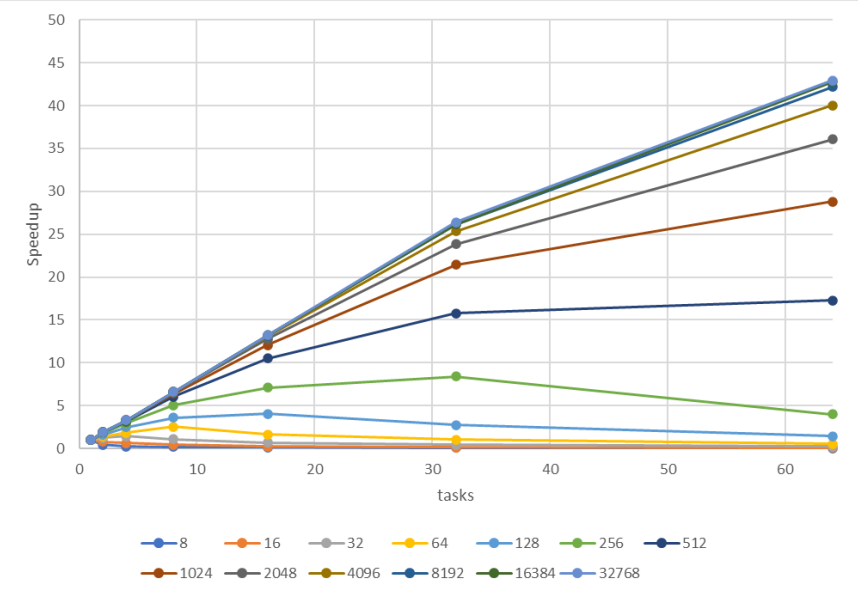


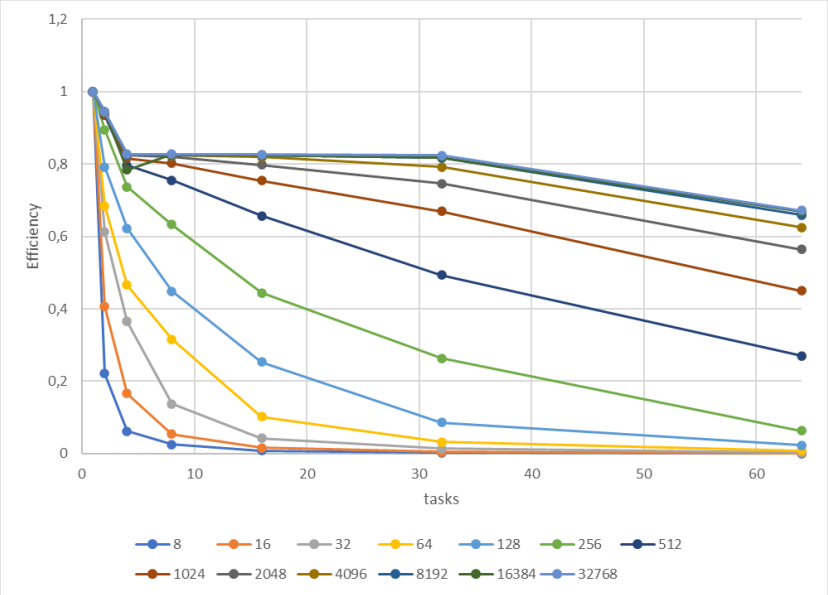


OpenMP

Pohitritev in učinkovitost

- $S(N,P) = \frac{t_s(N)}{t_p(N,P)}$, $E(N,P) = \frac{S(N,P)}{P}$
- Pohitritev narašča s številom niti
- Učinkovitost je višja pri več telesih
  - Celo 32768 teles ni dovolj za učinkovito računanje na 64 jedrih





OpenCL

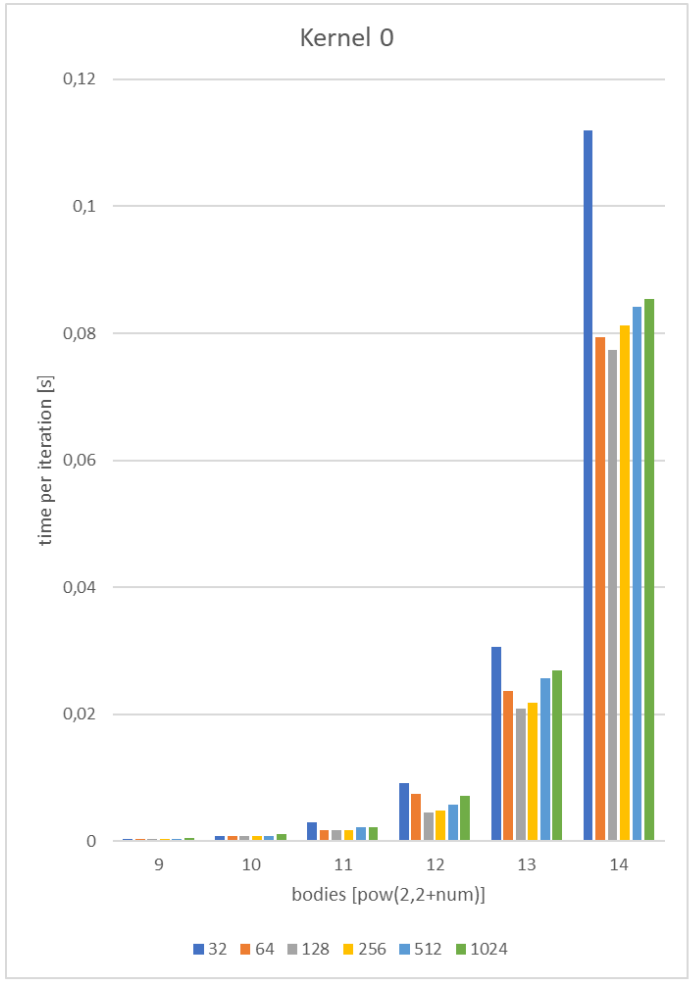
Koda

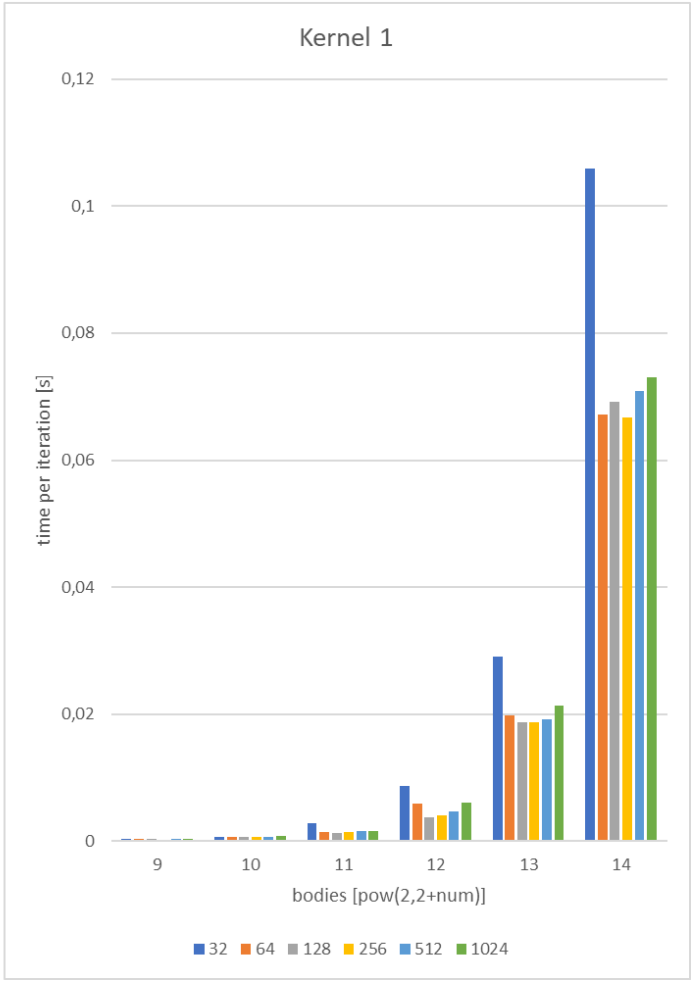
- Gostitelj
  - nbody-cl.c
- Dva ščepca
  - nbody-cl-0.cl
    - Dela z globalnim pomnilnikom
  - nbody-cl-1.cl
    - Položaje teles kopira v lokalni pomnilnik v paketih
    - Sile na telesa posodablja paket za paketom

OpenCL

Velikost delovne skupine

- Enojna natančnost
- Uporabljene velikosti
  - 32, 64, 128, 256, 512, 1024
- Komentar
  - Majhne delovne skupine: veliko pregrad
  - Velike delovne skupine: ni dovolj lokalnih virov
    - K40m: 2880 PE → 192 PE x 15 RE



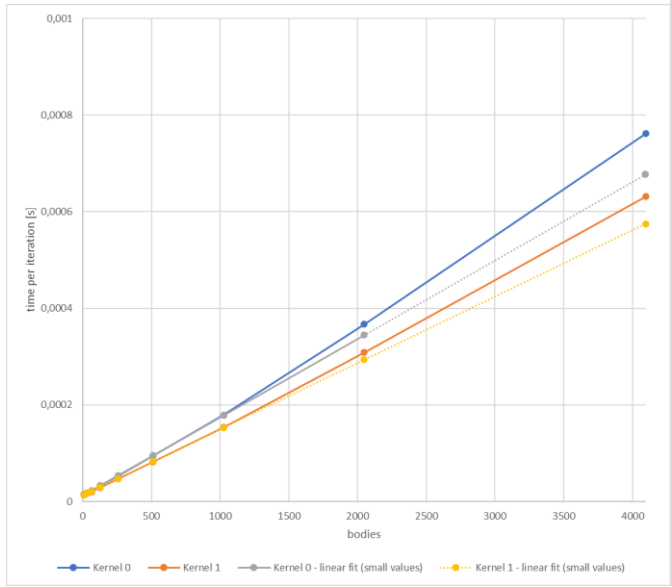


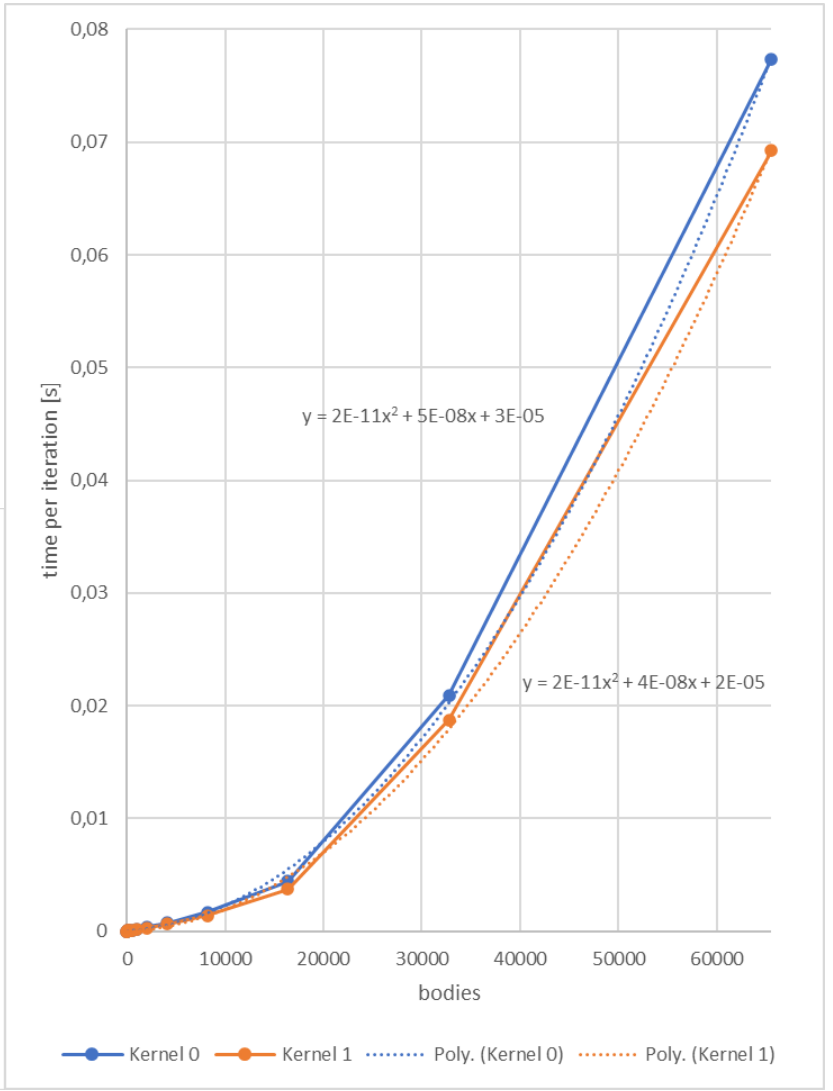
OpenCL

Čas za korak

- Kvadratična odvisnost od števila teles
- Čas enega izračuna χ
  - ščepec 0: $\chi = 1{,}72 \cdot 10^{-11} s$, ščepec 1: $\chi = 1{,}56 \cdot 10^{-11} s$
  - Ščepec 1 je hitrejši
  - Računanje na GPE je približno 1000-krat hitrejše kot na CPE

- Linearna odvisnost pri majhnem številu teles
  - GPE ni popolnoma izkoriščena
  - Za polno izkoriščenost vsaj $15 \cdot 128 = 1920$ niti

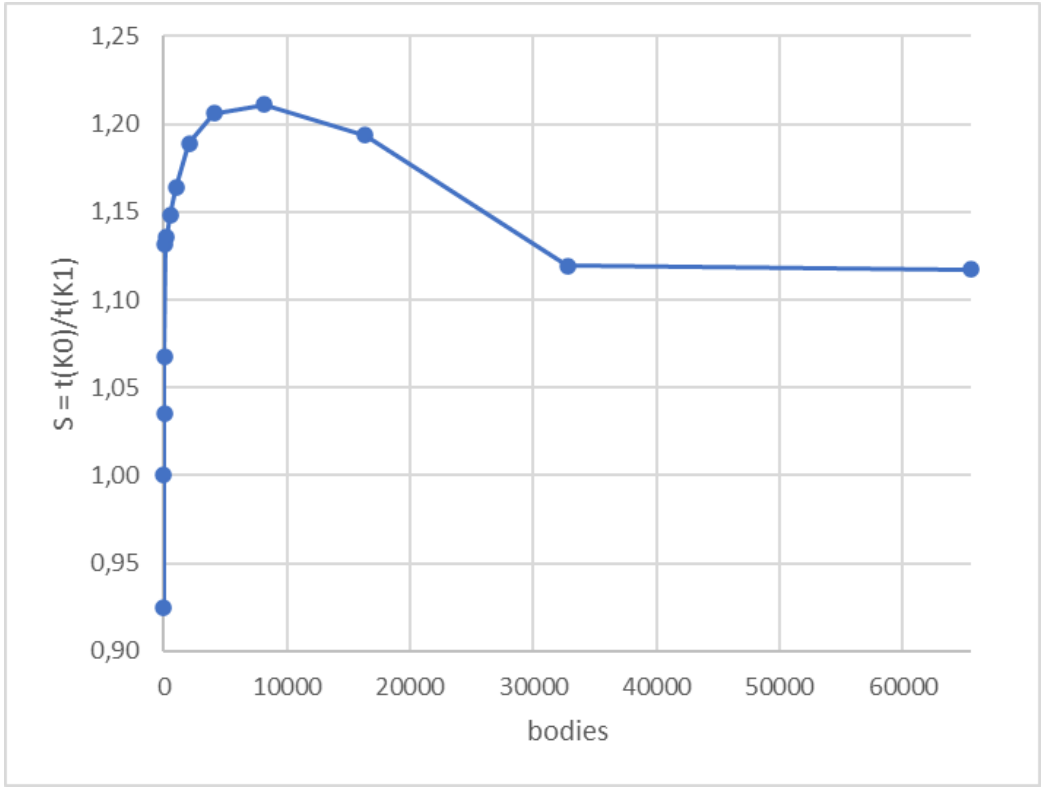




OpenCL

Čas za korak

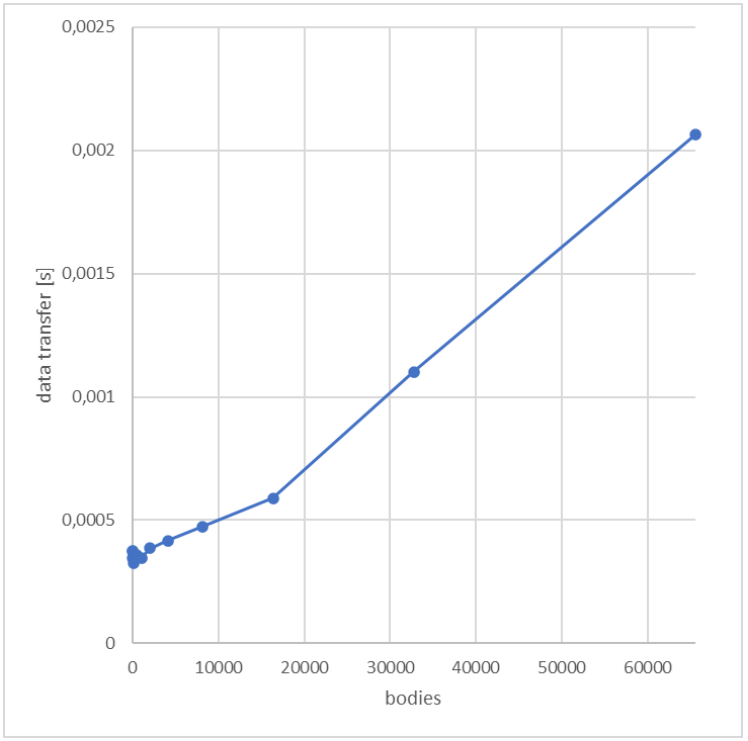
- Speedup: $S(N) = \frac{t_{\text{ščepec 0}}(N)}{t_{\text{ščepec 1}}(N)}$
- Analiza
  - ščepec 0 naključno dostopa do globalnega pomnilnika GPE
  - Ščepec 1 ima dve pregradi
  - Za zelo malo teles (< 128)
    - poravnan dostop do pomnilnika ne igra večje vloge
    - zaradi pregrad je ščepec 1 počasnejši
  - Za veliko teles
    - Mnogo niti, boljše zakrivanje latence
    - Od 256 do16384 teles GPE lahko hrani podatke v predpomnilniku
    - Ko ne gredo vsi podatki v predpomnilnik, je pohitritev 1,1-kratna

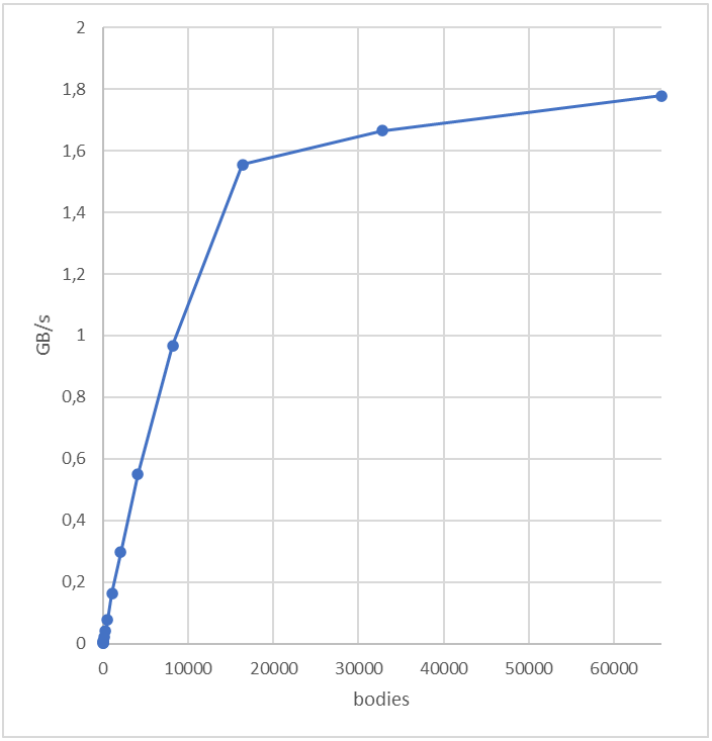


OpenCL

Prenos podatkov

- Iz gostitelja na GPE in nazaj
  - 2 x 7 x N x W: (na napravo in nazaj) x [1 masa + 3 položaji + 3 hitrosti) x število teles x 4B za enojno nat.
- Prenosi 1,8 GB/s
  - PCI-E 3.0 omogoča prenose do 10 GB/s
  - Problem ni dovolj velik, da bi dosegli maksimalno hitrost

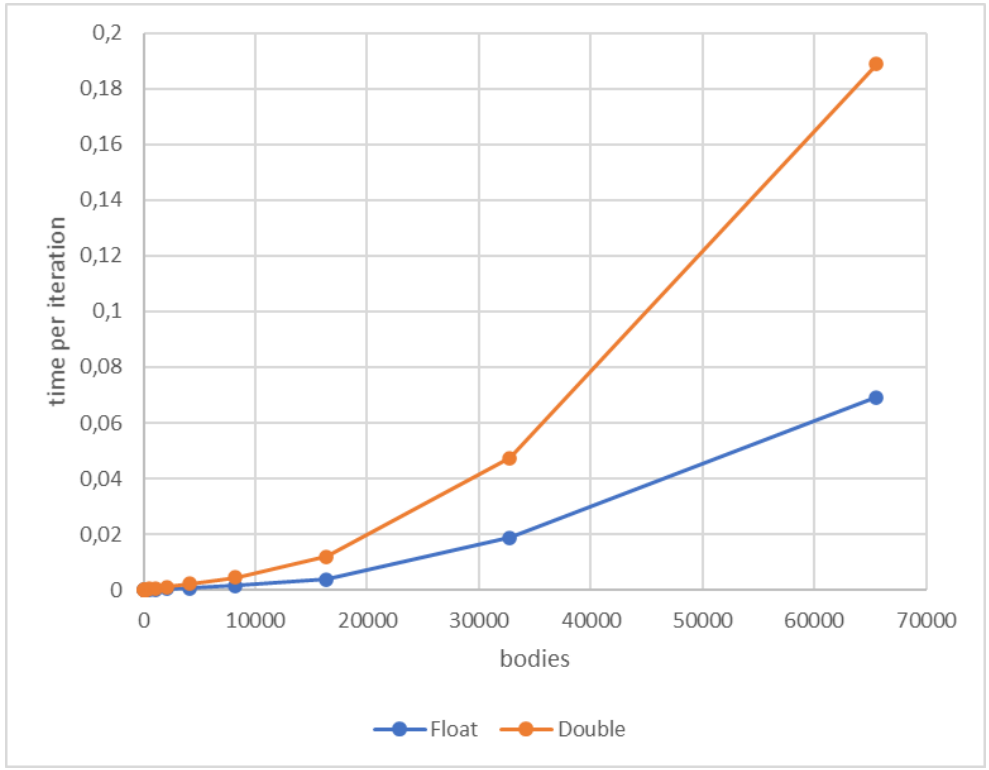


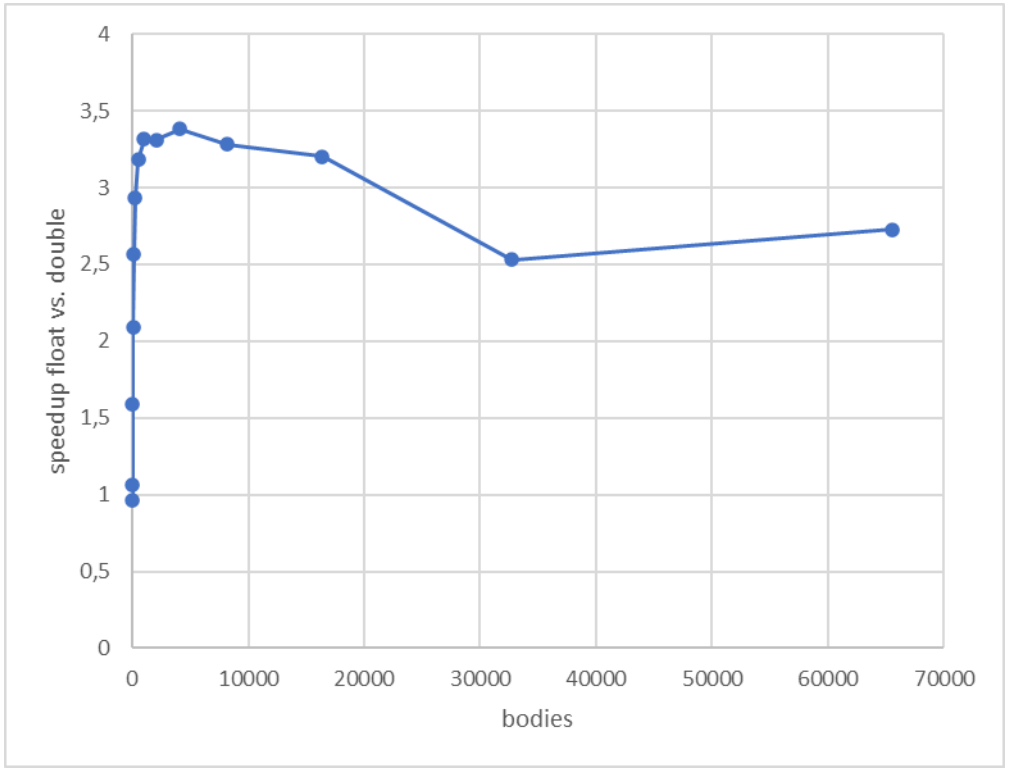


OpenCL

Enojna in dvojna natančnost

- 128 niti v delovni skupini
- Ščepec 1





OpenMPI

Koda

- nbody-mpi.c
- Skupinska komunikacija

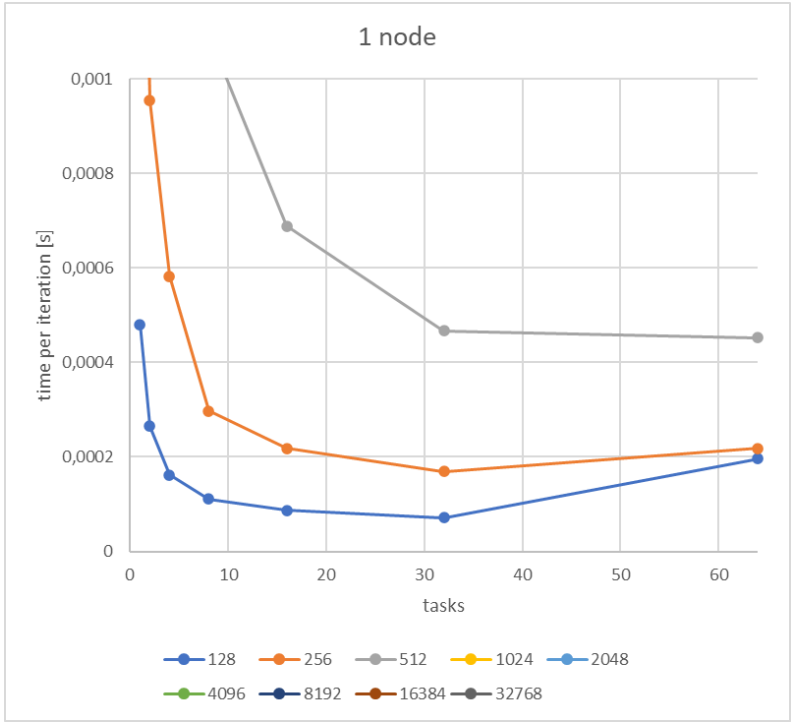
Poskus

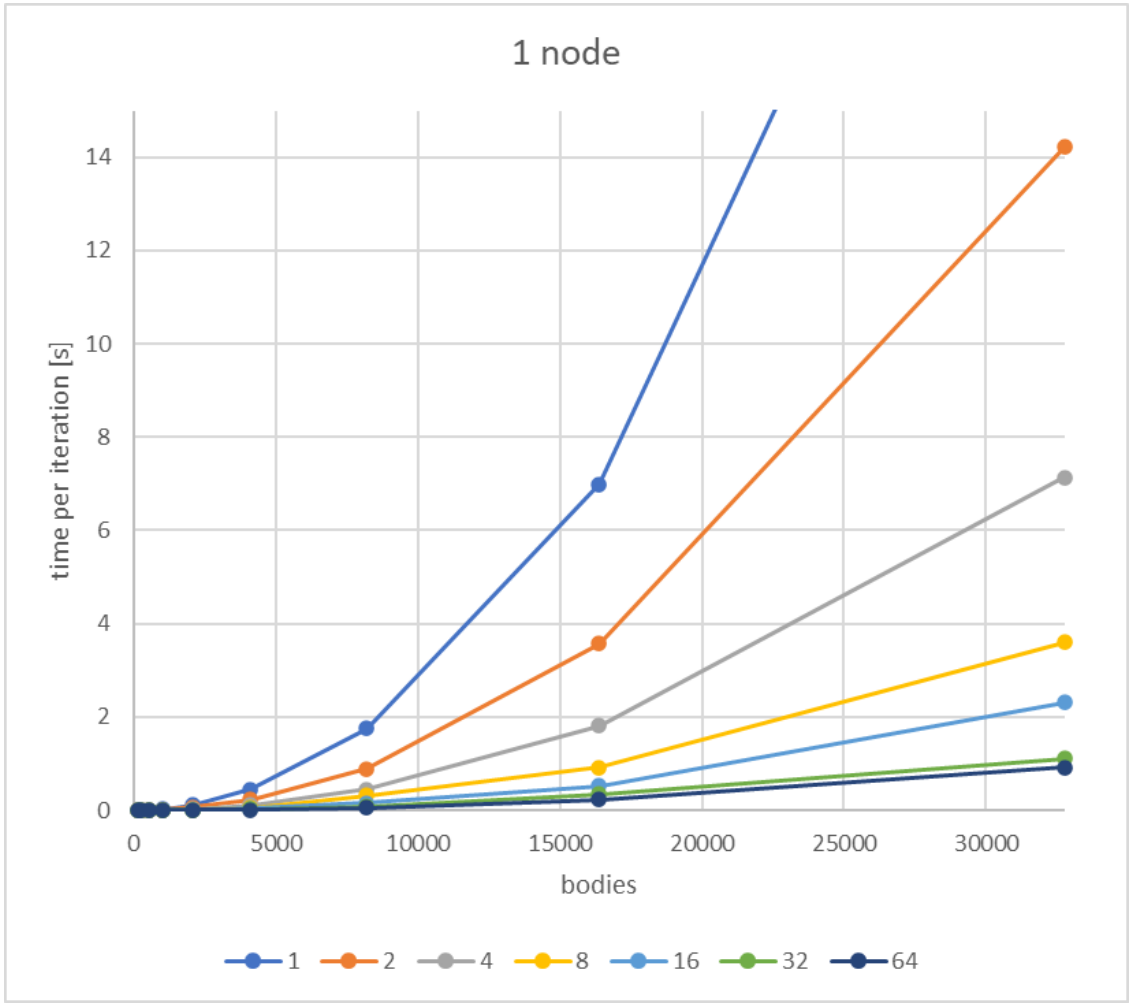
- 1, 2, 4, 8, 16, 32, 64, 128 procesov
- 1 ali 2 vozlišči – obe vozlišči AMD!
- Procesi so enakomerno porazdeljeni med vozlišča in procesorje

OpenMPI

Čas za korak, 1 vozlišče

- Računanje + Allgather
  - Za 64 procesov/jeder najmanj 512 teles

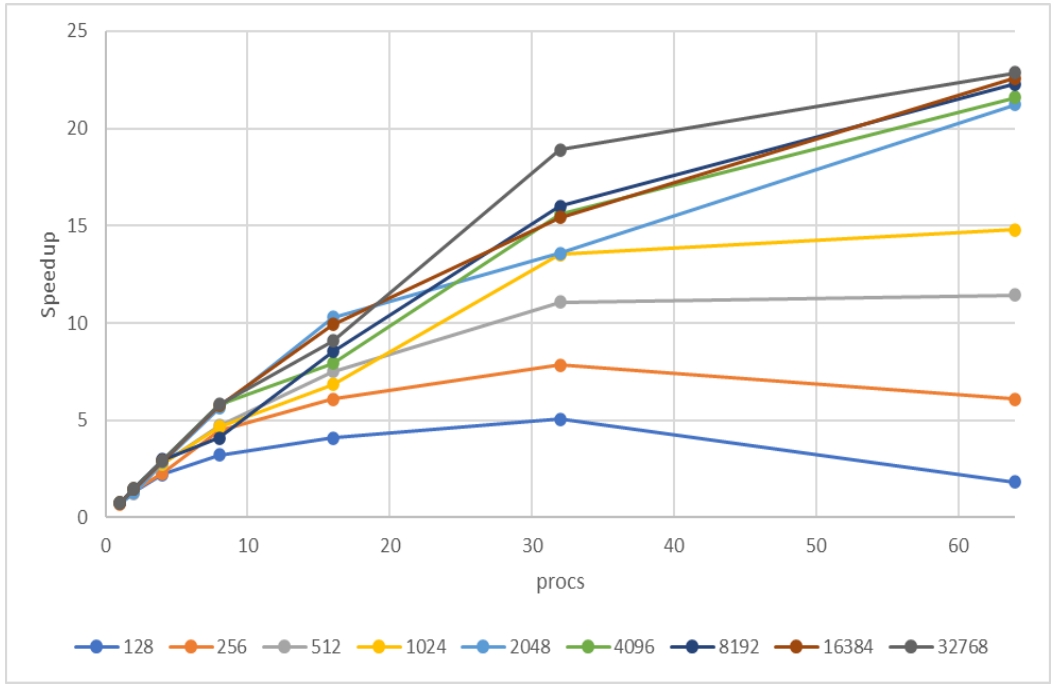


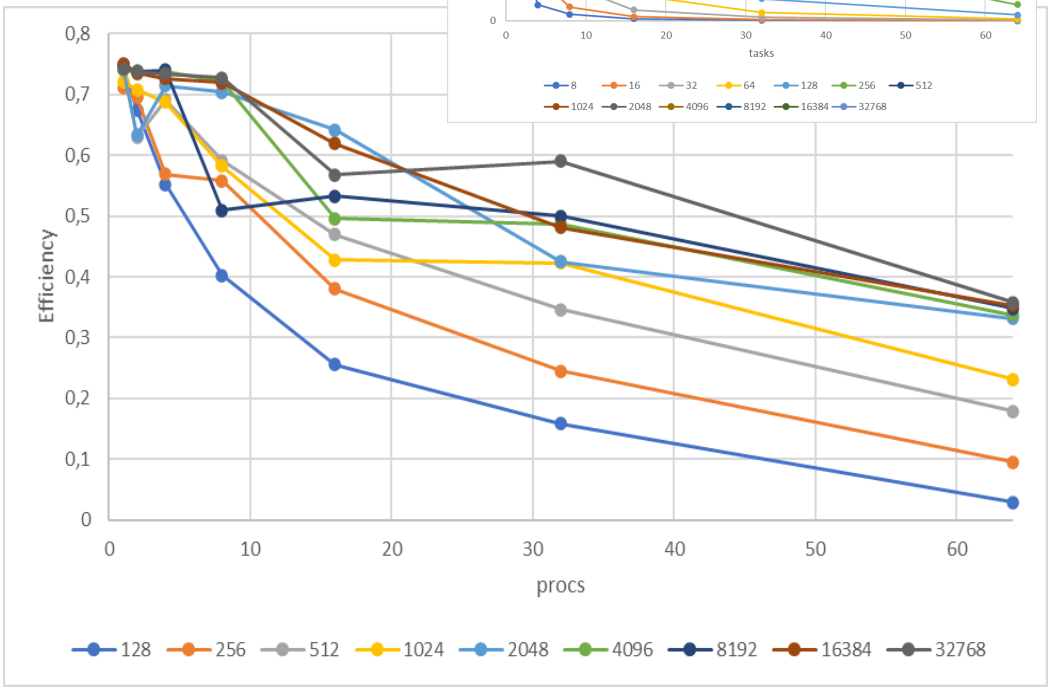


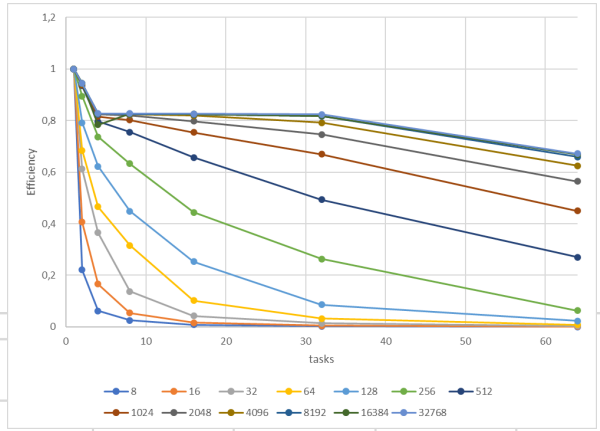
OpenMPI

Čas za korak, 1 vozlišče

- Pohitritev in učinkovitost
- OpenMPI je manj učinkovit kot OpenMP (75 %)
  - OpenMPI potrebuje večje probleme za boljšo učinkovitost
  - Hibridna rešitev: OpenMP + OpenMPI?





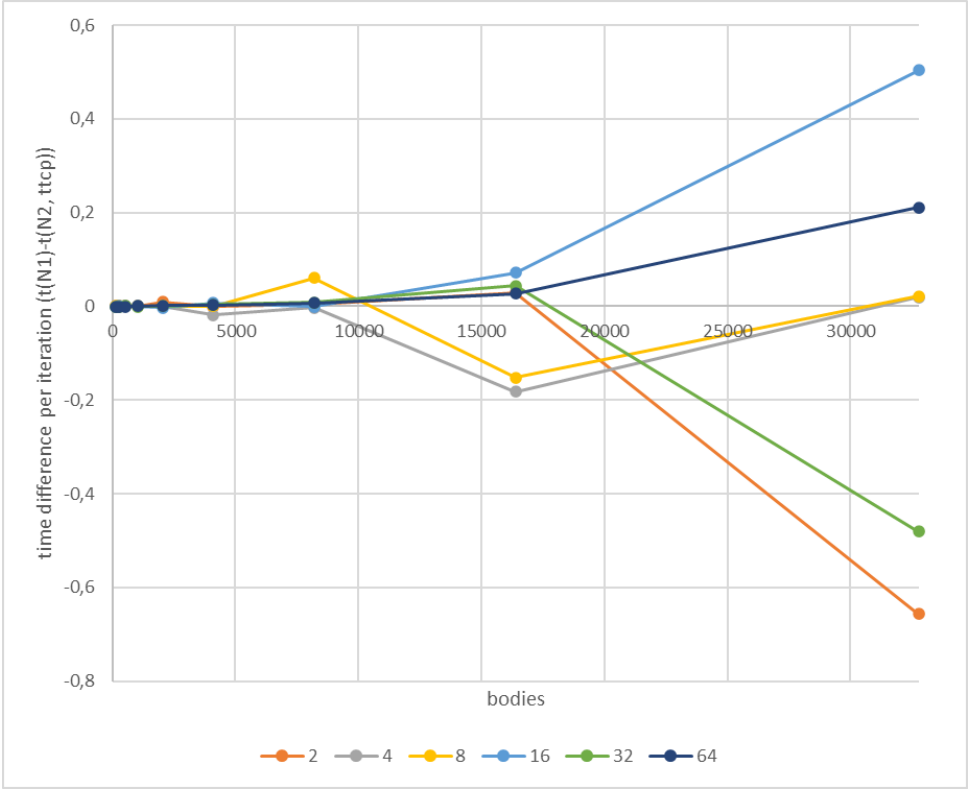


OpenMPI

Čas za korak, 2 vozlišči

- Gledamo razliko v časih: $t_{1\ vozlišče} - t_{2\ vozlišči}$
- Malce bolje na 2 vozliščih
  - Komunikacija samo med dvema vozliščema
  - Komunikacijski protokol med procesi na istem vozlišču?
  - Pri 2 vozliščih
    - manj težav s predpomnilnikom
    - Vsak proces ima na voljo več procesorskih virov (AMD, 2 jedri si delita enoto za računanje v plavajoči vejici)

bodies	1	2	4	8	16	32	64	128
128		-0,0000534	0,000028	2,88E-05	-0,00027	-0,00035	-0,00042	
256		-4,66E-05	9,64E-05	3,18E-05	-0,00014	-0,0003	-0,00131	
512		0,000558	3,08E-05	-6,8E-06	-8,9E-05	-0,00019	-0,00045	
1024		-3,86E-05	9,34E-05	0,000568	0,000741	-0,00039	-0,0002	
2048		0,0090488	0,000676	0,000314	-0,00274	0,000367	0,000976	
4096		-0,0007266	-0,01835	0,000502	0,007516	0,003891	0,00309	
8192		0,0051994	-0,00328	0,060123	-0,00023	0,007614	0,006723	
16384		0,0290196	-0,18183	-0,1526	0,071758	0,04376	0,027287	
32768		-0,6567244	0,019261	0,021433	0,504166	-0,48074	0,211331	
						Green: better time on 2 nodes		



OpenMPI

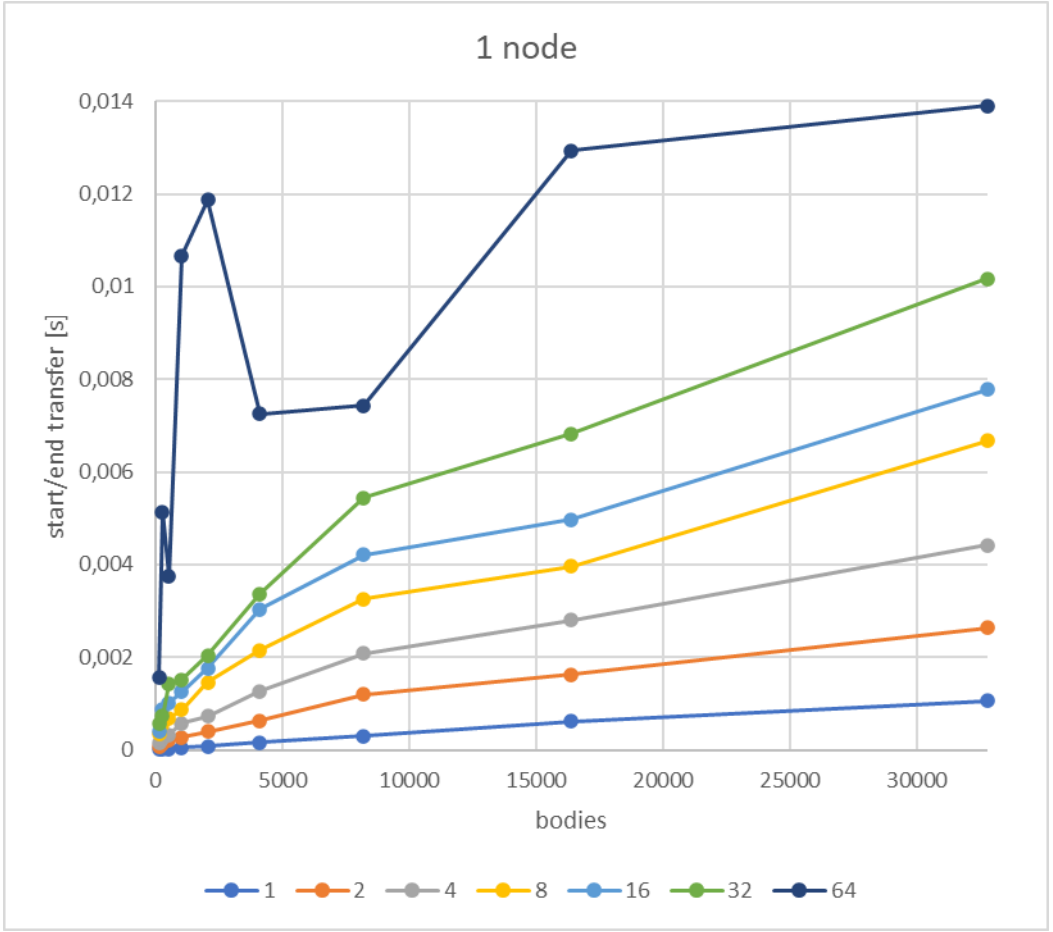
Ocena stroškov komunikacije

- Bcast: $\log_2 P \left(\lambda + \frac{N}{\beta}\right)$
- Scatter/gather/allgather: $\sum_{i=1}^{\log_2 P} (\lambda + \frac{N}{2^i \beta}) = \lambda \log_2 P + \frac{N}{\beta} \cdot \frac{P-1}{P}$

OpenMPI

Prenos podatkov, 1 vozlišče

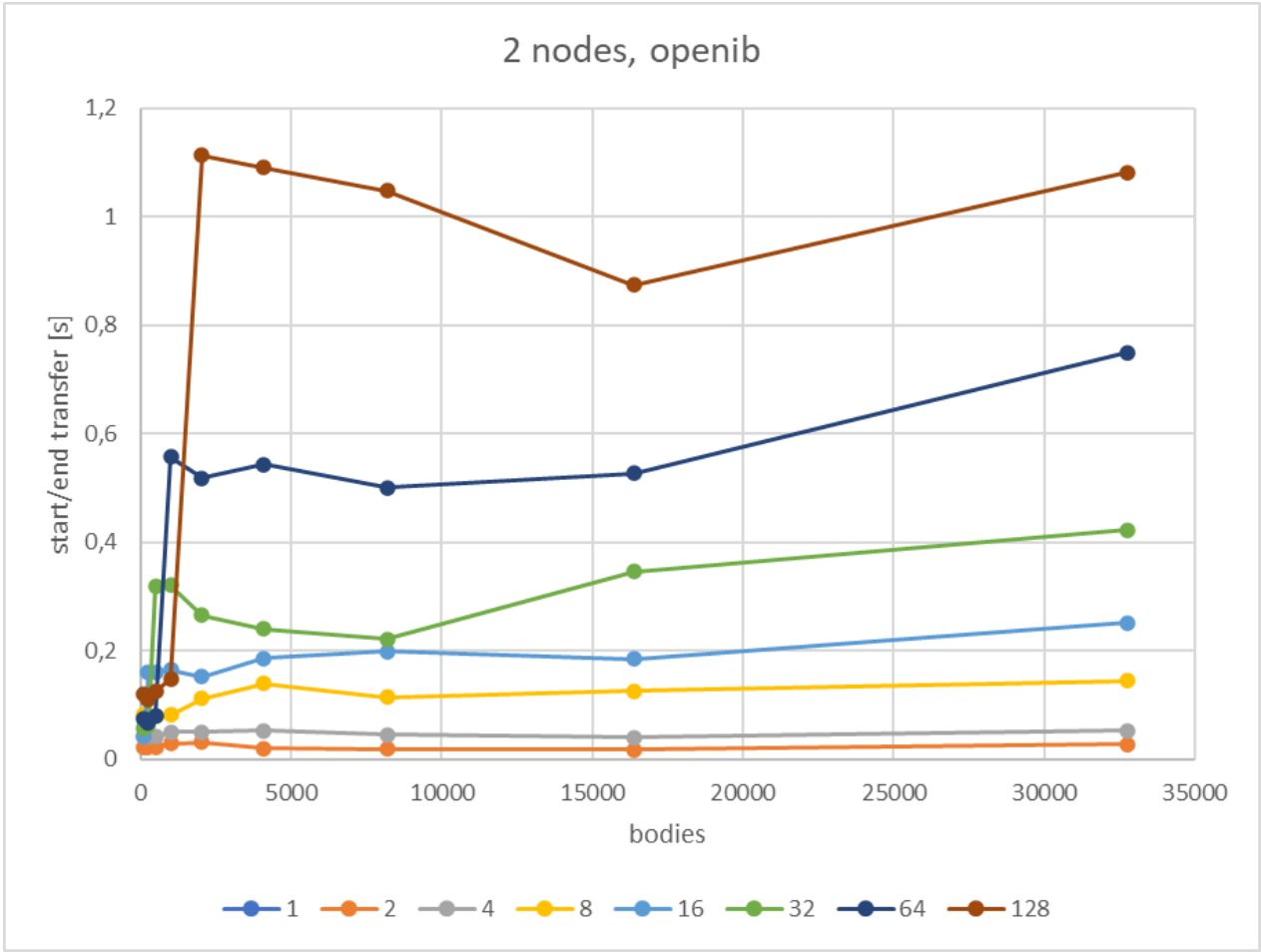
- 1 x bcast + 1 x scatter + 1 x gather
- Linearna odvisnost
  - Latenca $\lambda = 6{,}7 \cdot 10^{-5} s$
  - Pasovna širina $\beta = 450\ MB/s$



OpenMPI

Prenos podatkov, 2 vozlišči

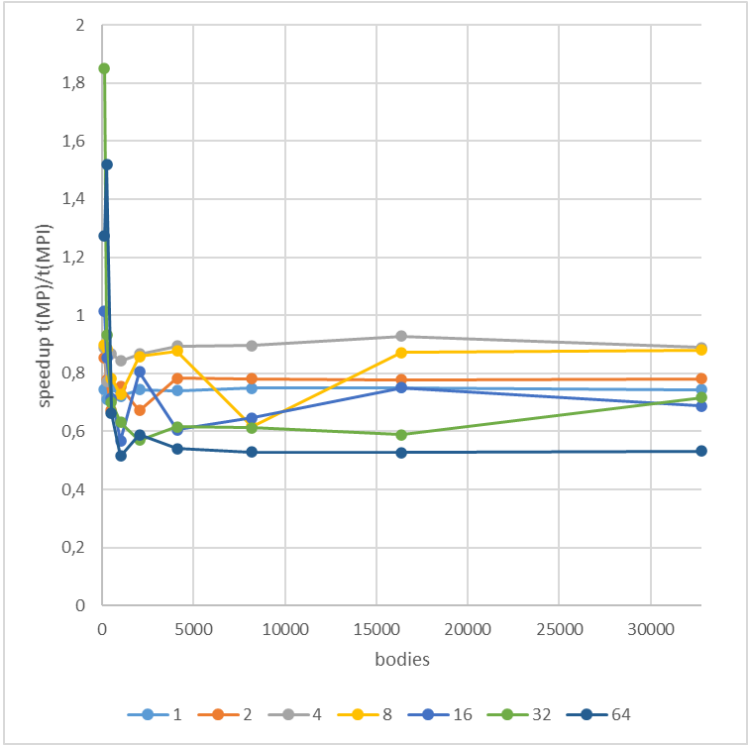
- Infiniband
  - $\lambda \approx 1{,}8 \cdot 10^{-2}\ s, \beta \approx 11\ MB/s$
- Prevladuje latenca, ni dovolj podatkov za prenos?
- Boljša ocena
  - Prenos večjih količin podatkov, več zagonov za boljšo statistiko
  - Testiranje komunikacije ločeno od računanja

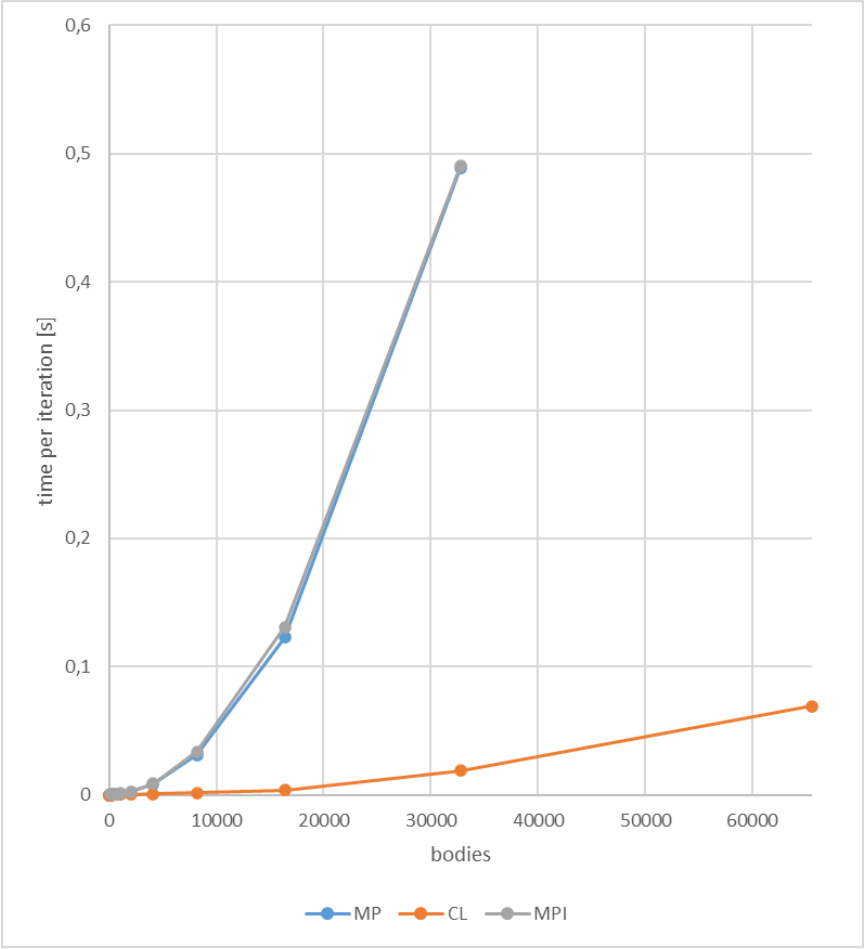


Primerjava vseh treh pristopov

Čas za korak

- Najboljši rezultat za posamezno tehnologijo
- Problem je primeren za OpenCL
- OpenMP and OpenMPI imata podobne rezultate
  - Pohitritve OpenMPI so 75 % pohitritev OpenMP
  - Hibridna rešitev
    - OpenMPI med vozlišči
    - OpenMP na vozlišču

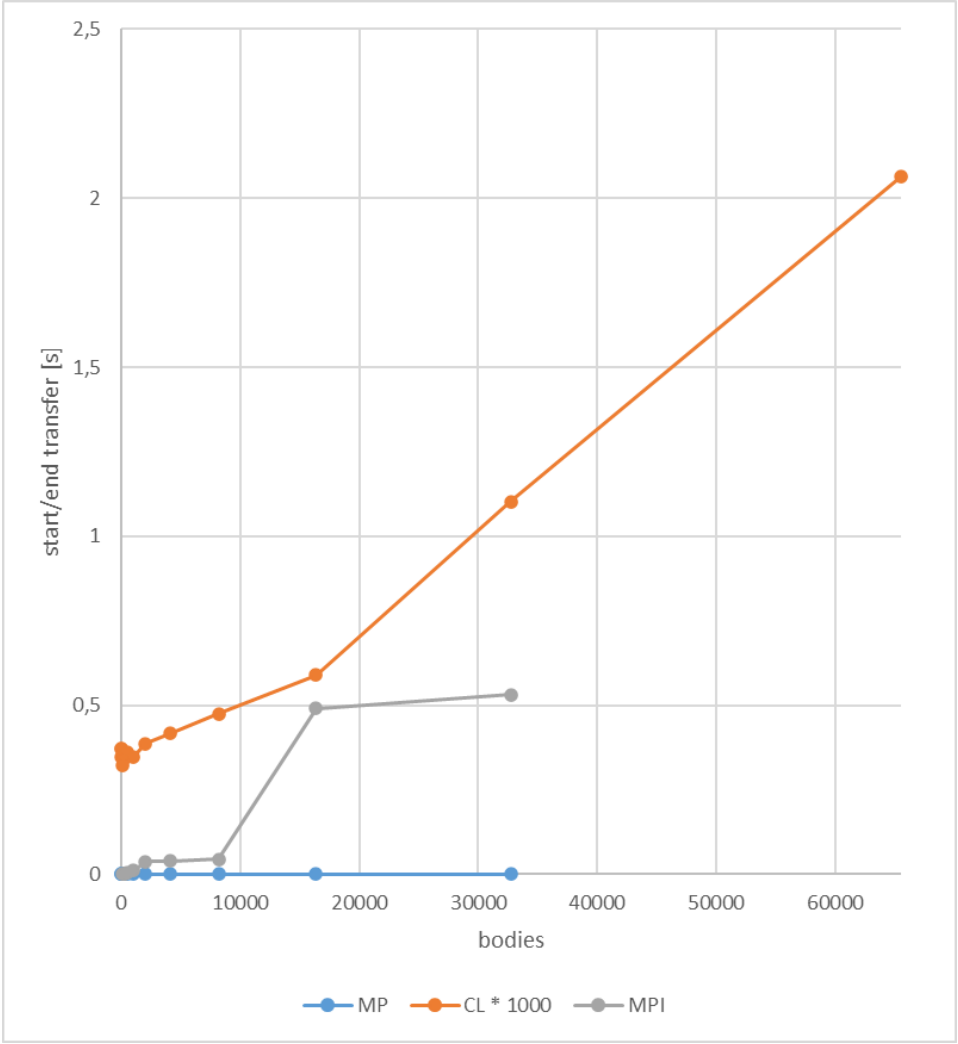




Primerjava vseh treh pristopov

Prenos podatkov

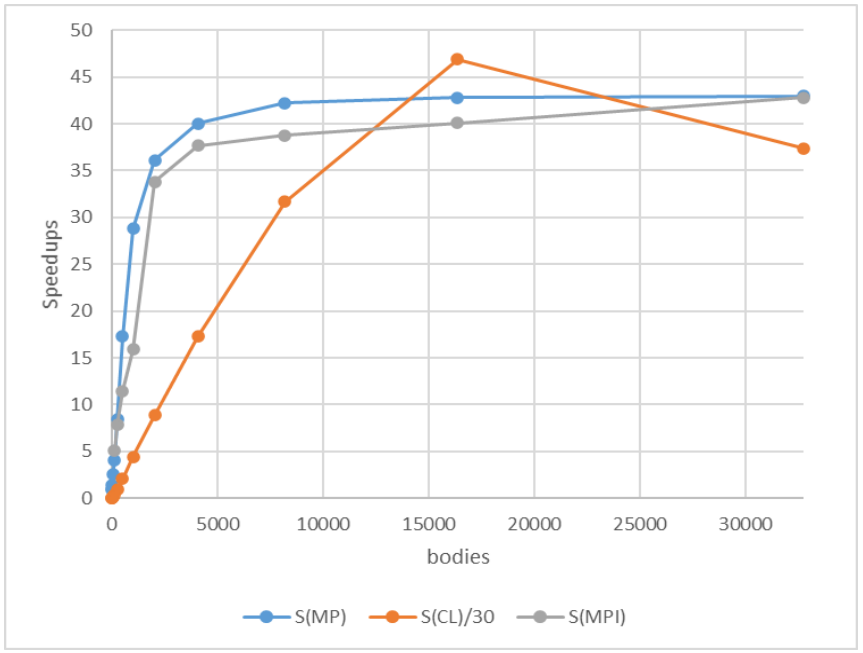
- OpenCL je 1000-krat hitrejši
- OpenMP nima eksplicitnega prenosa podatkov
- OpenMPI prenaša podatke tudi med procesi na istem vozlišču

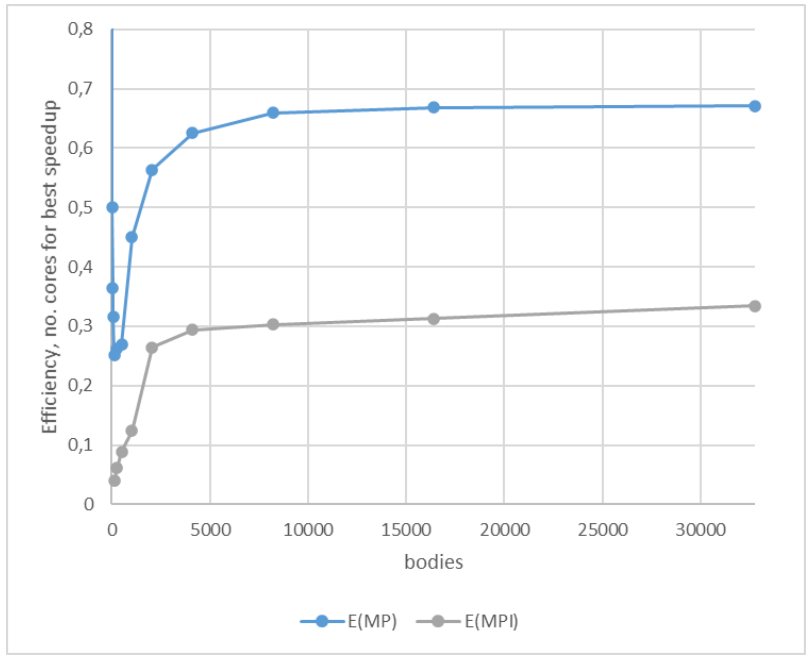


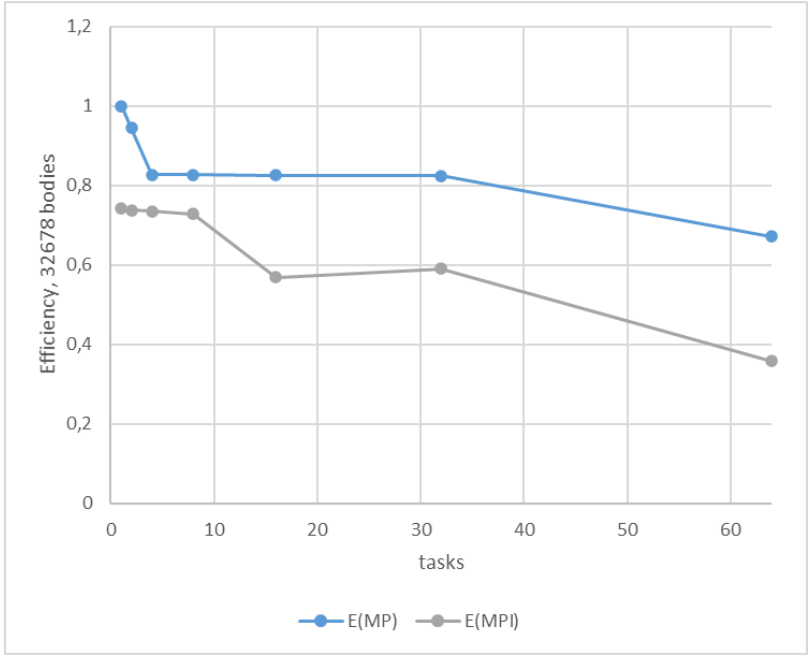
Primerjava vseh treh pristopov

Pohitritev in učinkovitost

- Glede na čase izvajanja na enem jedru
- OpenCL je 30-krat hitrejši od OpenMP in OpenMPI
- Učinkovitost pada s številom niti/procesov
- OpenMP je bolj učinkovit od OpenMPI







Možne izboljšave

Boljše meritve

- Večje število korakov
- Večje število meritev pri istih parametrih

Premalo podatkov za dobro oceno stroškov komunikacije

- Ločen prenos podatkov za večje probleme
- Ločeno merjenje časov za bcast in scatter/gather/allgather
- morda bolje gledati komunikacijo med dvema procesoma, ki tečeta na istem ali na različnih vozliščih

Meritev časa za odseke programa

- Lažje določanje parametrov teoretičnega modela

Možne izboljšave

Hibridne rešitve

- OpenCL + OpenMP
  - Ne bomo veliko pridobili
- OpenCL na dveh GPE
  - Pridobimo precej, nekaj bo stroškov prenosa podatkov med GPE
- OpenCL + OpenMPI
  - Eno jedro na GPE
    - MPI skrbi za prenos podatkov med jedri
    - Jedra skrbijo za izračune na GPE
  - Eno jedro na vozlišče
    - Jedro izkoristi vse GPE
    - Impliciten prenos podatkov med GPE