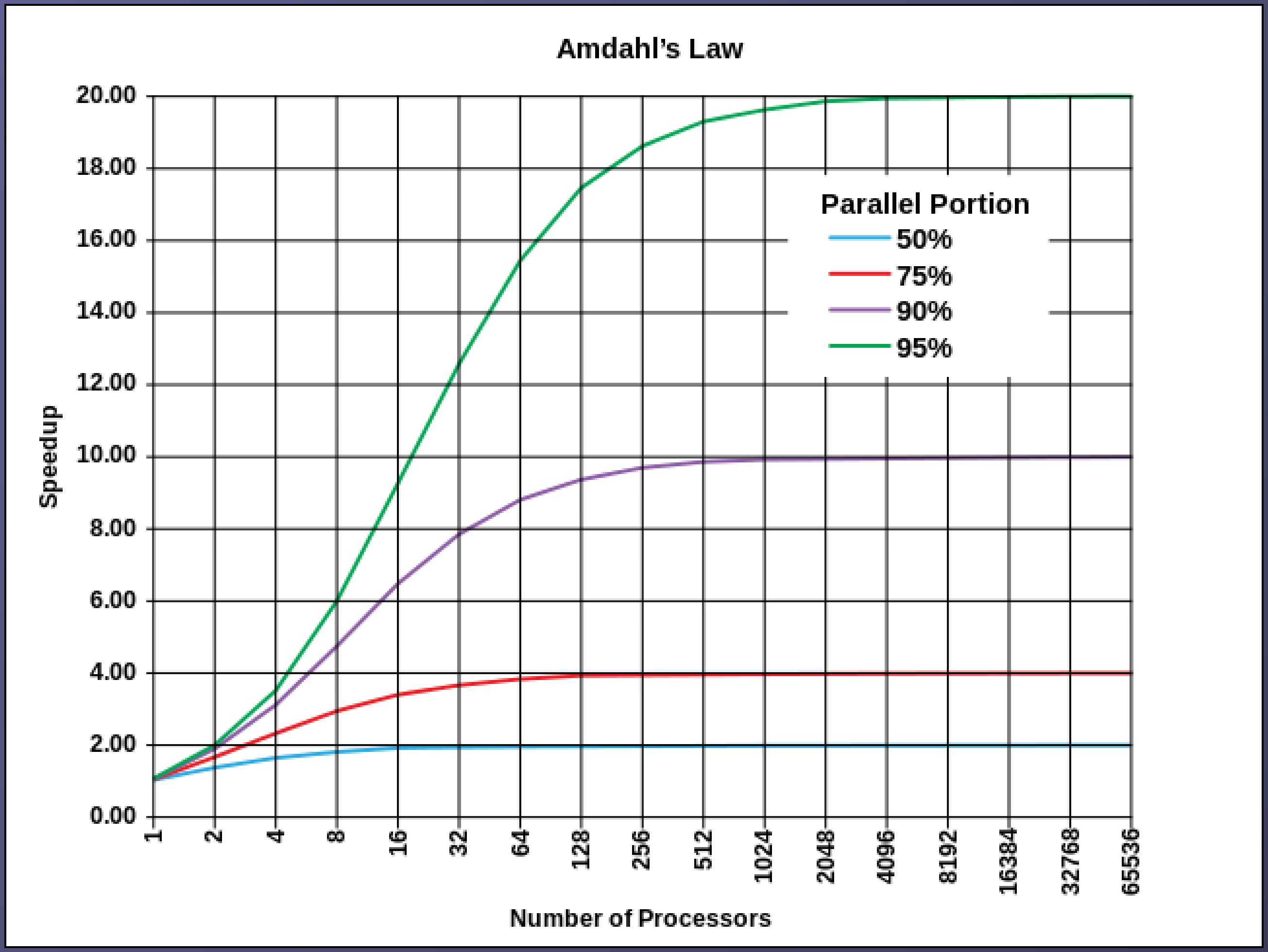
Parallel Computing

Some basic ideas

Amdahl's Law (Gene Amdahl 1967)



Evolution according to Amdahl's law of the theoretical speedup of the execution of a program in function of the number of processors executing it, for different values of p. The speedup is limited by the serial part of the program. For example, if 95% of the program can be parallelized, the theoretical maximum speedup using parallel computing would be 20 times.

Calculate Amdahl's Law:

Let X be the part of my program (in terms of computing time) which can be parallelised. The sequential computing time T_{seq} is normalized to unity (1), and can be expressed as:

$$T_{seq} = 1 = X + (1-X)$$

The parallel computing time Tpar under ideal conditions (ideal load balancing, ultrafast communication):

$$T_{par} = X/p + (1-X)$$

with processor number (core number) p ;

Then the speed-up of the program $S = T_{seq} / T_{par}$:

$$S = 1 / (1-X+X/p) \quad ;$$

Note: $T_{par}/T_{seq} = 1/S$ (sometimes also plotted)

Note the limit of S for p>>1 and X~1 is very large: S = 1/(1-X). S ~ p

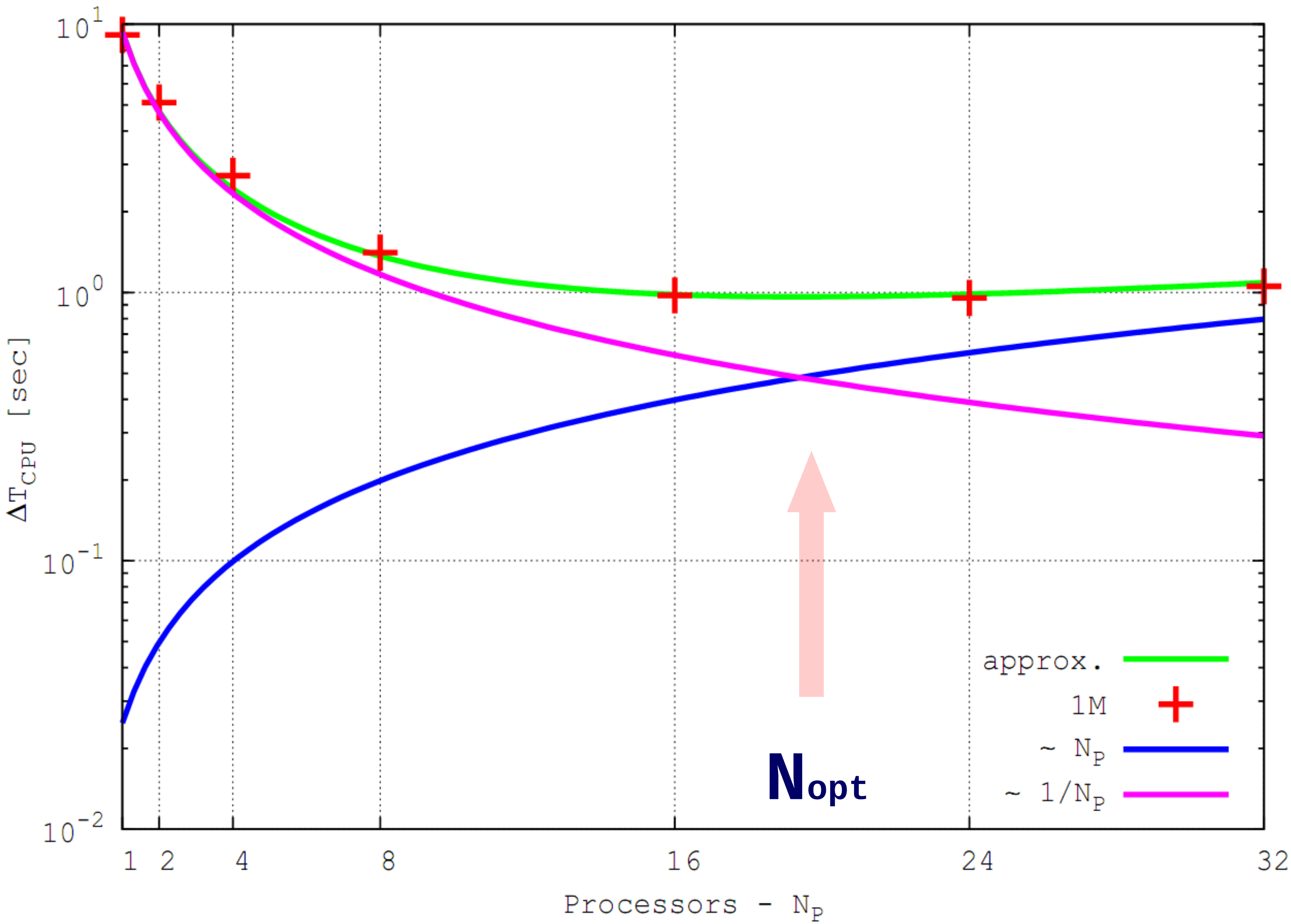
With communication overhead:

$$T_{par} = X/p + (1-X) + T_{comm} \quad \rightarrow \quad S = 1 / (1-X+X/p+T_{comm})$$

If T_{comm} independent of p we have for large p: $S = 1 / (1-X + T_{comm}) = \text{const.}$

If $T_{comm} = c\, p^k$ (k>0) we get: $S = 1 / (1-X + c\, p^k) \rightarrow 0$ for large p!!!

Parallel code on cluster



Strong and Soft Scaling

- Strong Scaling: Fixed Problem size, increase p
- Soft Scaling: Increase Problem size, increase p

(constant amount of work per processing element)

Ansatz for Soft Scaling (T_{comm} neglected here):

- $T_{seq} = p\,(X + (1-X))$
- $T_{par} = X + p\,(1-X)$

If $X \sim 1$: $p \gg 1$: → $T_{seq} \sim pX$; $T_{par} \sim X$

- $S = T_{seq}/T_{par} \sim p$

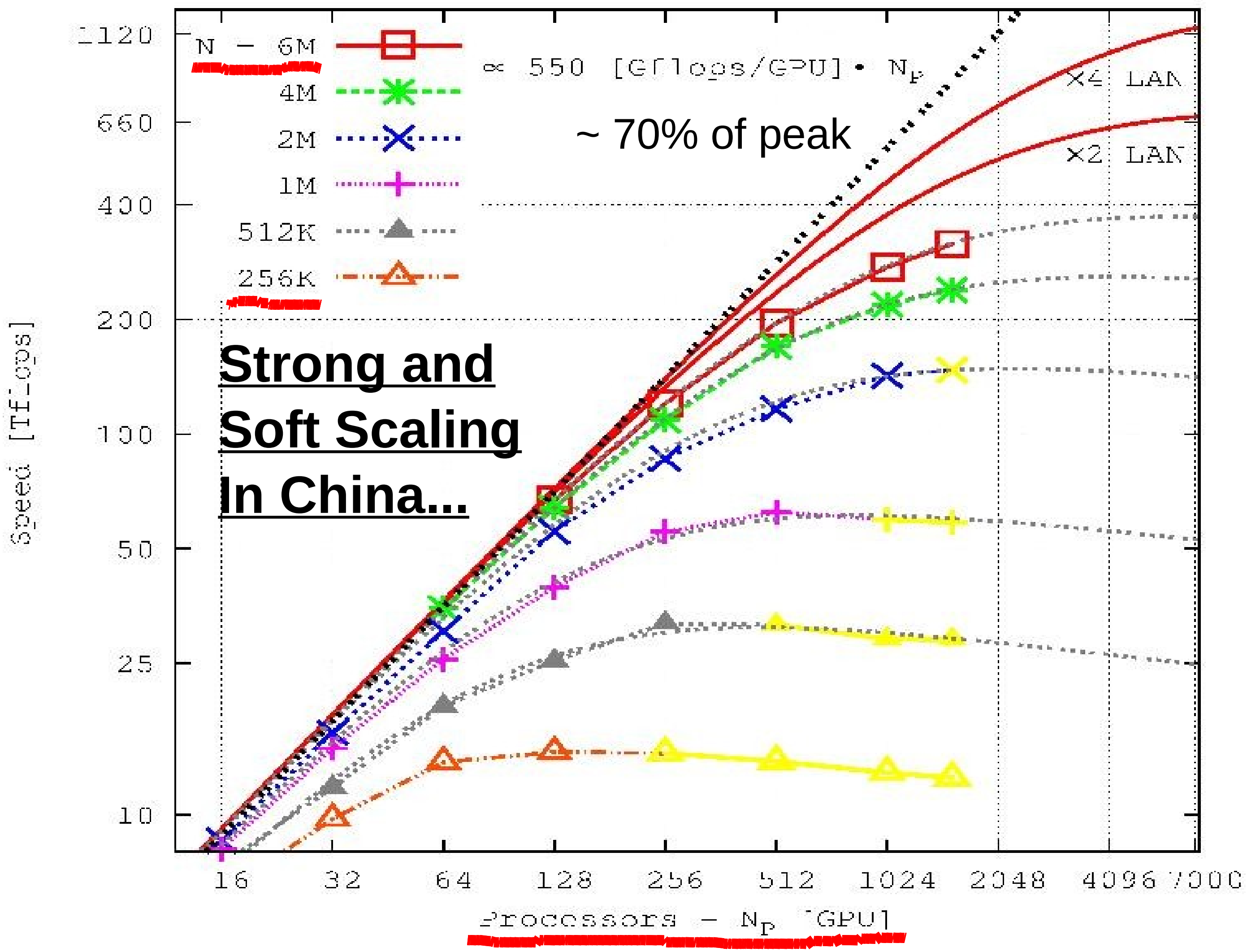
ΦGPU – NBODY Code

中国科学院国家天文台
National Astronomical Observatories, CAS

350 Teraflop/s
1600 GPUs .
440 cores
= 704.000
GPU-Cores

Using
Mole-8.5
of
IPE/CAS
Beijing

Berczik et al.
2013



中国科学院过程工程研究所
Institute Of Process Engineering, Chinese Academy Of Sciences

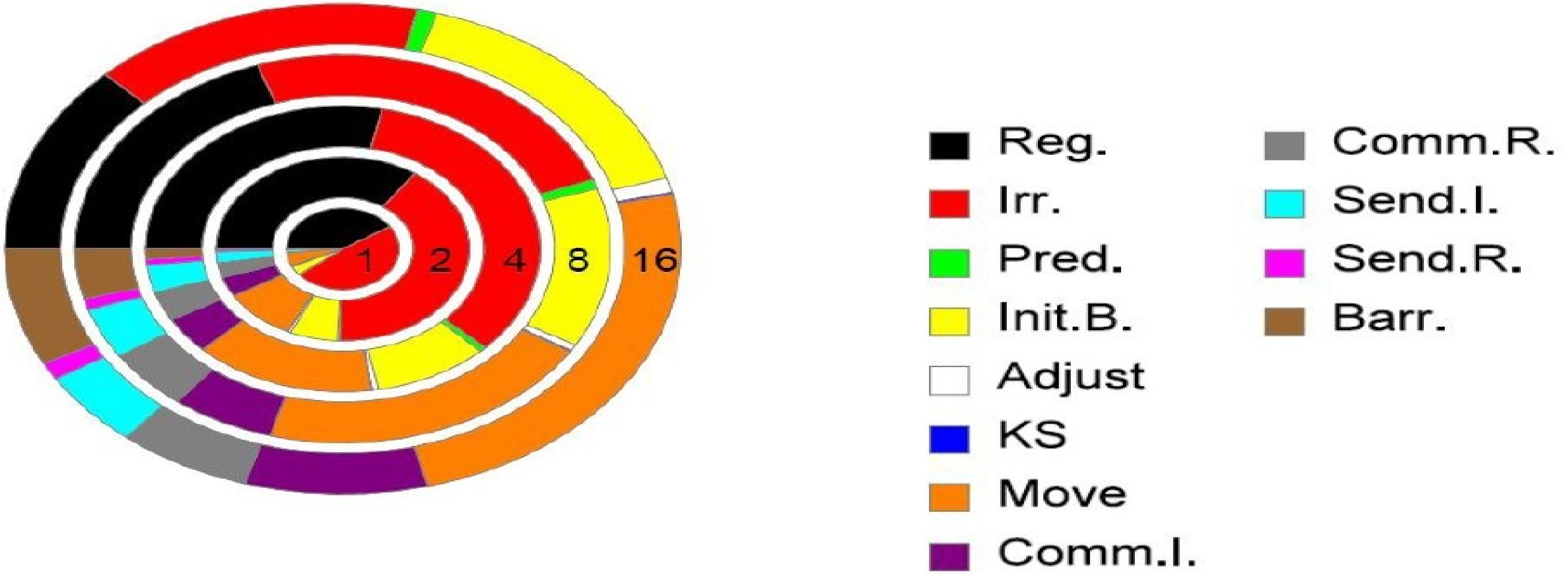
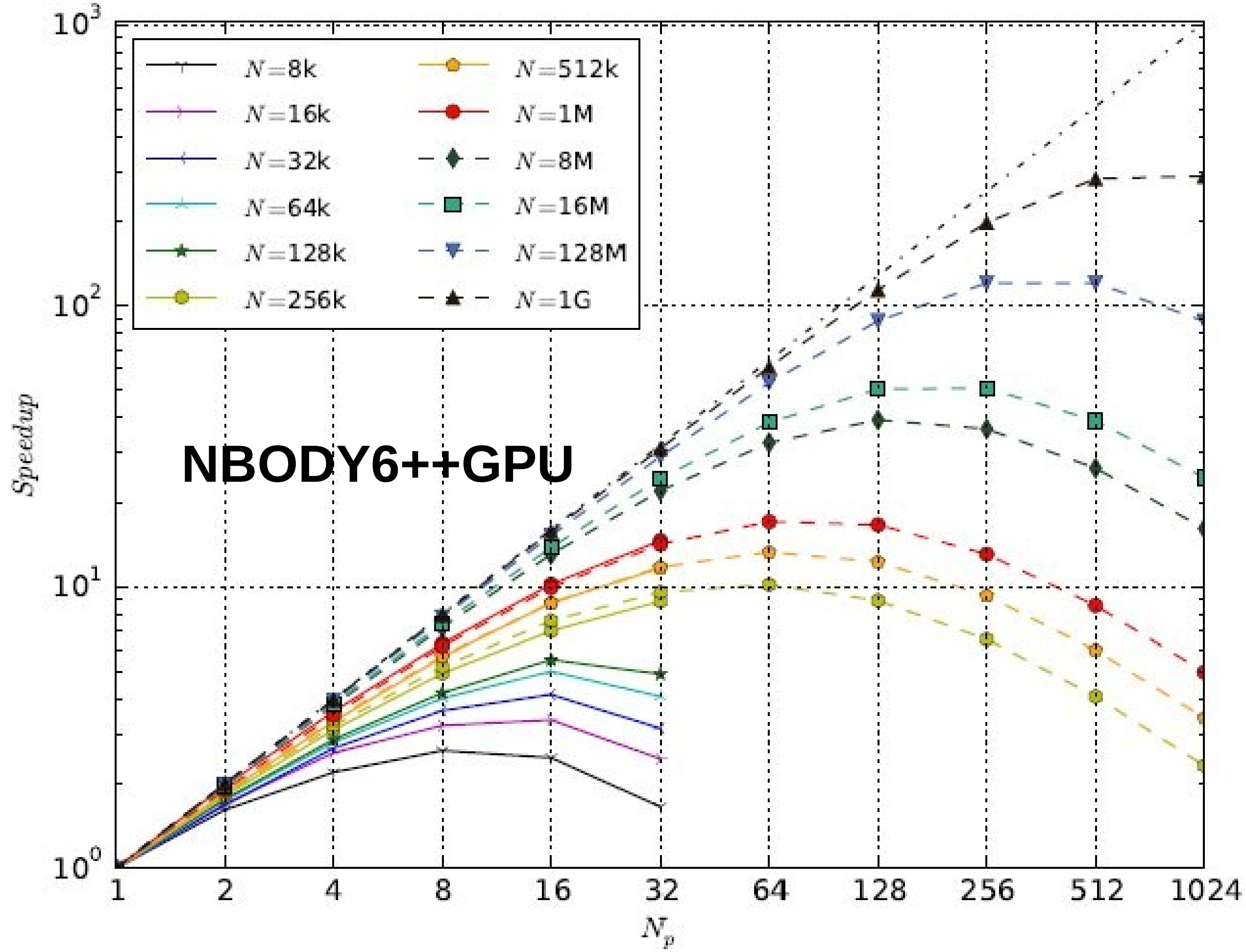


Table 1 Main components of NBODY6++

Description	Timing variable	Expected scaling N	Expected scaling N_p	Fitting value [sec]
Regular force computation	$T_{\rm reg}$	$\mathcal{O}(N_{\rm reg} \cdot N)$	$\mathcal{O}(N_p^{-1})$	$(2.2 \cdot 10^{-9} \cdot N^{2.11} + 10.43) \cdot N_p^{-1}$
Irregular force computation	$T_{\rm irr}$	$\mathcal{O}(N_{\rm irr} \cdot \langle N_{nb} \rangle)$	$\mathcal{O}(N_p^{-1})$	$(3.9 \cdot 10^{-7} \cdot N^{1.76} - 16.47) \cdot N_p^{-1}$
Prediction	$T_{\rm pre}$	$\mathcal{O}(N^{kn_p})$	$\mathcal{O}(N_p^{-kp_p})$	$(1.2 \cdot 10^{-6} \cdot N^{1.51} - 3.58) \cdot N_p^{-0.5}$
Data moving	$T_{\rm mov}$	$\mathcal{O}(N^{kn_{m1}})$	$\mathcal{O}(1)$	$2.5 \cdot 10^{-6} \cdot N^{1.29} - 0.28$
MPI communication (regular)	$T_{\rm mcr}$	$\mathcal{O}(N^{kn_{cr}})$	$\mathcal{O}(kp_{cr} \cdot \frac{N_p - 1}{N_p})$	$(3.3 \cdot 10^{-6} \cdot N^{1.18} + 0.12)(1.5 \cdot \frac{N_p - 1}{N_p})$
MPI communication (irregular)	$T_{\rm mci}$	$\mathcal{O}(N^{kn_{ci}})$	$\mathcal{O}(kp_{ci} \cdot \frac{N_p - 1}{N_p})$	$(3.6 \cdot 10^{-7} \cdot N^{1.40} + 0.56)(1.5 \cdot \frac{N_p - 1}{N_p})$
Synchronization	$T_{\rm syn}$	$\mathcal{O}(N^{kn_s})$	$\mathcal{O}(N_p^{kp_s})$	$(4.1 \cdot 10^{-8} \cdot N^{1.34} + 0.07) \cdot N_p$
Sequential parts on host	$T_{\rm host}$	$\mathcal{O}(N^{kn_h})$	$\mathcal{O}(1)$	$4.4 \cdot 10^{-7} \cdot N^{1.49} + 1.23$



Huang, Berczik, Spurzem, Res. Astron. Astroph. 2016, 16, 11.

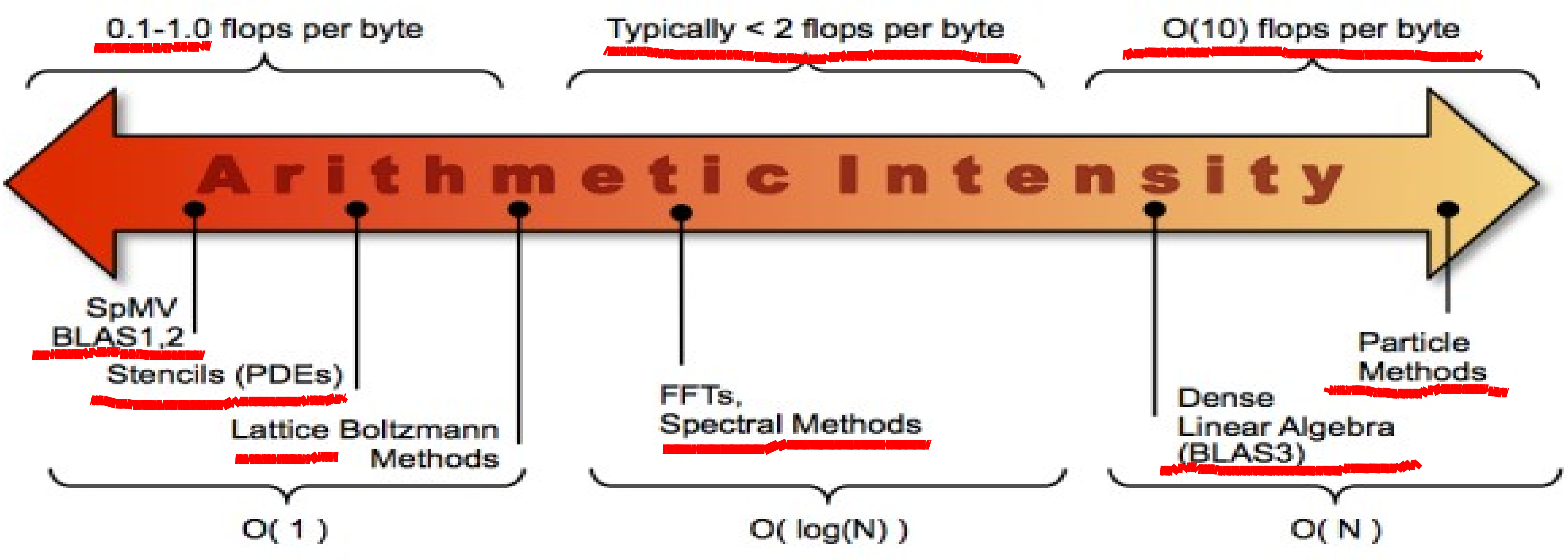
Fig. 2 The speed-up (S) of NBODY6++ as a function of particle number (N) and processor number (N_p). Solid points are the measured speed-up ratio between sequential and parallel wall-clock time, dash lines predict the performance of larger scale simulations further. The symbols used in figure have the magnitudes: $1k = 1,024$, $1M = 1k^2$ and $1G = 1k^3$.

Roofline Performance Model (LBL)

https://crd.lbl.gov/divisions/amcr/computer-science-amcr/par/research/roofline/

Arithmetic Intensity

The core parameter behind the Roofline model is Arithmetic Intensity. Arithmetic Intensity is the ratio of total floating-point operations to total data movement (bytes).



Roofline Performance Model (LBL): Lorena Barba Research Group at George Washington University: https://lorenabarba.com/

http://lorenabarba.com/wp-content/uploads/2012/01/roofline_slide.png