ITMO UNIVERSITY

# Assessment of communication complexity

Mehti Musayev
M4551c academic group

# Performance effectiveness of Parallel Algorithms

- optimal selection of communication networks in parallel computers
- minimization of needed inter process communication and other accompanying overheads  (parallelization, control of PA, waiting times)

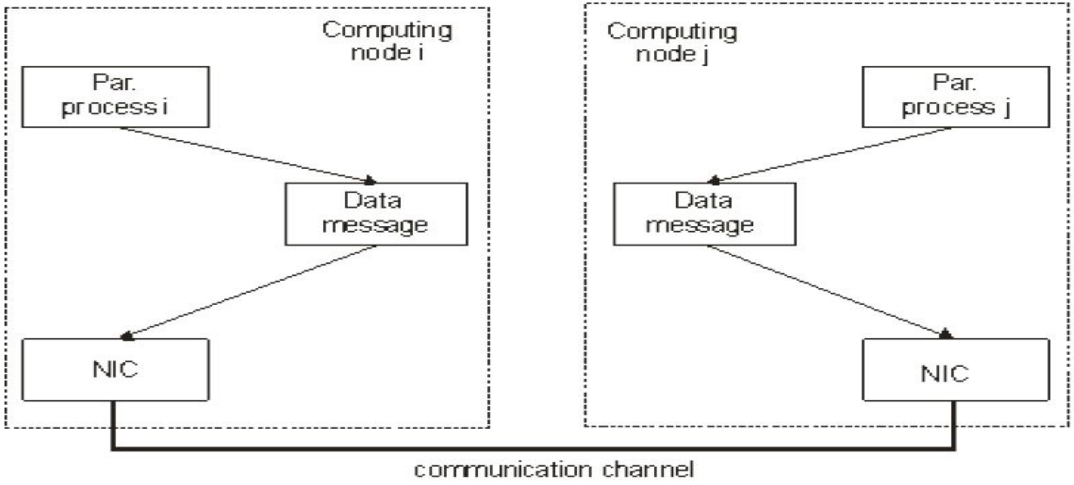


*Figure 2. Typical MPI network communication.*

# Model parameters in parallel computing models

- Semantic
  - communications network architecture (architecture, channels, control)
  - communication methods (communication protocols)
  - communication delay (latency)
- performance (complexity, efficiency)
  - working load s for given PA
  - size of the parallel system p (number of processors)
  - workload w - number of operations
  - sequential program execution time T(s, 1)
  - the computation execution time T_comp(s, p)
  - the whole execution time of a parallel algorithm T(s, p)
  - parallel speed up S(s, p)
  - efficiency E(s, p)
  - isoefficiency w(s)
  - average time of computation unit tc  (instruction, defined computing step etc.)

# Model parameters in parallel computing models

- communication technical parameters
  - average time to initialize communication (startup time) – $t_s$
  - average time to transmit data unit (data word) - $t_w$

# Modeling of Communication Complexity

- achieve high performance of all actual parallel computers (SMP, Now, Grid)
- to develop effective PA there is necessary to model and optimize inter process communications mainly for parallel algorithms with distributed memory
- Parallel computation time T_comm(s,p):

$$T(s,\mathrm{p})_{comp} = \frac{Z_{pp}.t_c}{p}$$

$$T(s,\mathrm{p})_{comp} = \lim_{p\to\infty} \frac{Z_{pp}.t_c}{p} = 0$$

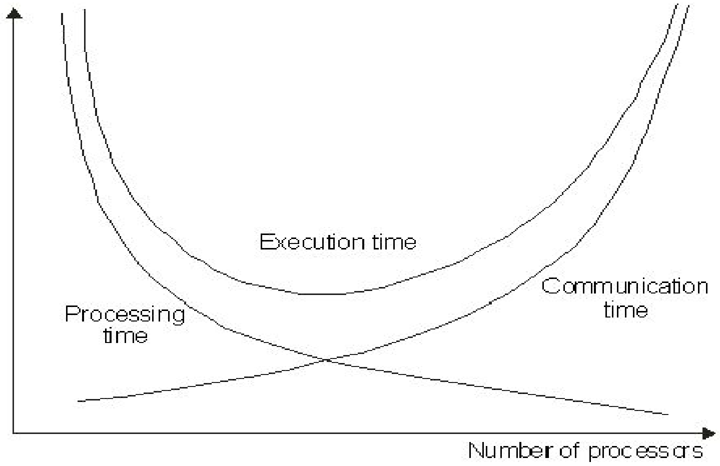


*Figure 9. Relations among parts of parallel execution time.*

# Communication Latencies

- two basic communication parameters:
  - communication parameter t_s  defined as parameter for initialization of communication step (startup time)
  - communication parameter t_w  as parameter for transmission latency of considered data unit (typically word)

$$T(s,p)_{comm} = Z(s,p)_{comm}(t_s + t_w)$$

# Communication Latencies

- function f1 (t_s ) which represents the whole number of communication initializations for given parallel process
- function f2 (t_w ) which correspondents to whole performed data unit transmission (usually time of word transmission for given parallel computer) in given parallel process

$$T(s,p)_{commNOW} = f_1(t_s) + f_2(t_w)$$

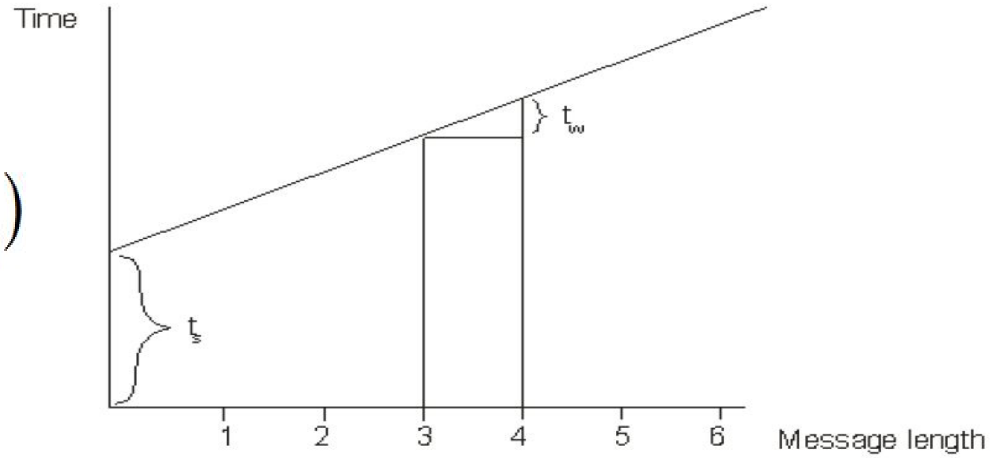


*Figure 10. Illustration of communication parameters.*

# Communication Latencies

- communication latency in Grid:

$$T(s,p)_{commGRID} = f_1(t_s) + f_2(t_w) + \sum_{i=1}^{u} f_3(t_s, t_w, l_h)$$

where:

- l_h is the number of network hops
- m is the number of transmitted data units (usually words)
- t_h is average communication time for one hop

# Communication Latencies of PA

- one might consider influences of overheads in parallel computing (communication, synchronization, parallelization, waiting etc.)
    - complexity in PA are to be reduced to only complexity analysis of own computations T_comp(s, p)
    - function of all existed control and communication overhead latencies h(s, p) were not a part of derived relations for whole parallel execution time T(s,p)

# Communication Latencies of PA

- the dominated function in the relation for used isoefficiency function w(s) of the parallel algorithms is complexity of performed massive computations T_comp(s, p)
- really true in using classical parallel computers (supercomputers, massive SMP, SIMD architecture etc.)

$$w(s) = \max \left[ T(s,p)_{comp}, h(s,p) \right] < T(s,p)_{comp}$$

# Communication Latencies of PA: most important overheads

- architecture of parallel computer T_arch(s, p)
- own computations T_comp(s, p)
- communication latency T_comm(s, p)
    - start - up time (t_s)
    - data unit transmission (t_w)
    - routing
- parallelization latency T_par(s, p)
- synchronization latency T_syn(s, p)
- waiting caused by limiting shared technical resources T_wait(s, p) (memory modules, communication channels etc.)

$$h(s,p)=\sum\left(T(s,p)_{arch},T(s,p)_{par},T(s,p)_{comm}T(s,p)_{syn}\right)$$

# References


- Presentation: https://github.com/b5y/parallel_programming
- Juraj Hanuliak. Modeling of Communication Complexity in Parallel Computing. American Journal of Networks and Communications. Special Issue: Parallel Computer and Parallel Algorithms. Vol. 3, No. 5-1, 2014, pp. 29-42. doi: 10.11648/j.ajnc.s.2014030501.13

# Thank you for attention!


- Email: mehty.musaev@gmail.com
- c.p.: +79110395135
- LinkedIn: https://linkedin.com/in/mehtimusayev