Parallel Efficiency

Goal: linear speedup

$$T(p) = \frac{T(1)}{p}$$

Reality: performance is lost

$$T(p) = \frac{T(1)}{e(p)p}$$

Efficiency $e(p) < 1$ reduces effective number of CPN

Why?

No general answer, too many factors

Negative Factors

Simulation

- high number of sub-time-steps
- long boundaries
- slow data handling

Parallel Architecture

- load imbalances
- non-scalable control mechanisms

Communication

- slow encoding / decoding
- low bandwidth / high latency

Assumptions for Traffic Simulation

Simulation Data Topology

- homogeneous street network,
- only local data dependencies,

Parallelization Technique

- domain decomposition

Simulation Control

- time-step driven system
- ignore statistics
- ignore control messages

Boundary Communication

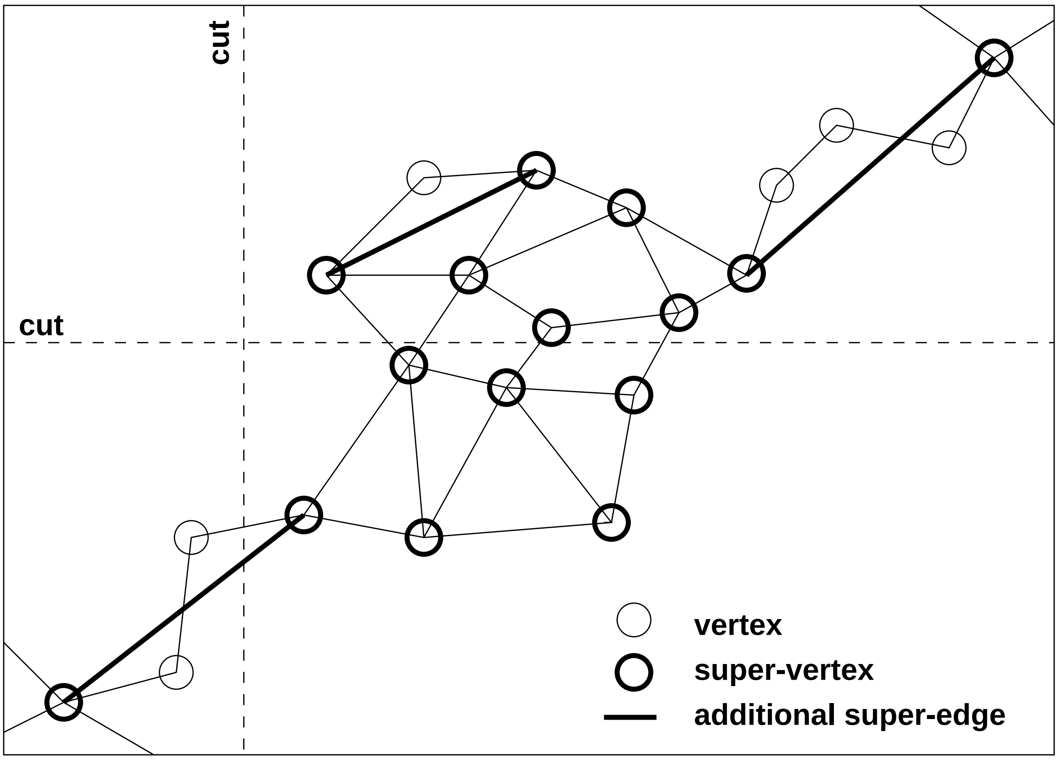
Most communication
is generated by
exchange of boundaries
on a
per sub-time-step
basis.

Two questions:

- How many boundaries are there?
- How long does it take to process the boundaries required in one time-step?

Estimating Parallel Efficiency

How many boundaries?

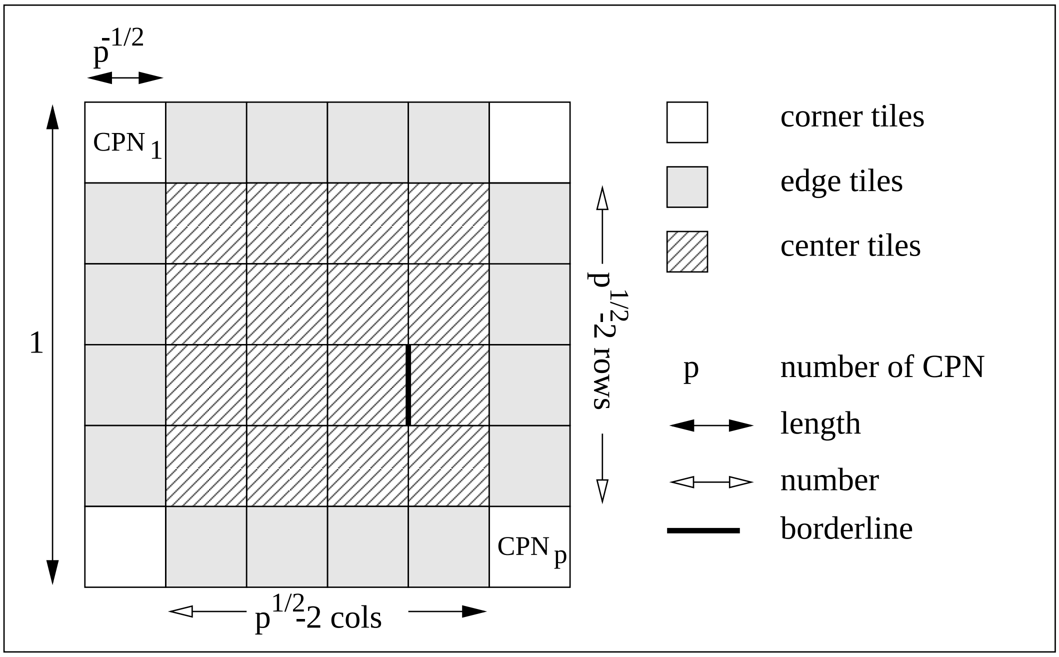


Assumption: All vertices with degree 2 have been replaced

$\Rightarrow N$ super-vertices and E super-edges

What is the probability of cutting a super-edge?

Map Decomposition I



Number of Boundaries $B(p)$

$$
\begin{aligned}
B(p) \quad = & \quad \varrho_E & \text{(edge density)} \\
\times & \quad N(p) & \text{(number of borders)} \\
\times & \quad l_b(p) & \text{(border length)}
\end{aligned}
$$

Map Decomposition II

Edge Density

$$\varrho_E(p) = \sqrt{N}\frac{deg(G)}{2} = \sqrt{N}\frac{2E}{2N} = \frac{E}{\sqrt{N}}$$

Number of Borders (Neighbours)

$$N(p) = 8 + 12(\sqrt{p} - 2) + 4(\sqrt{p} - 2)^2$$

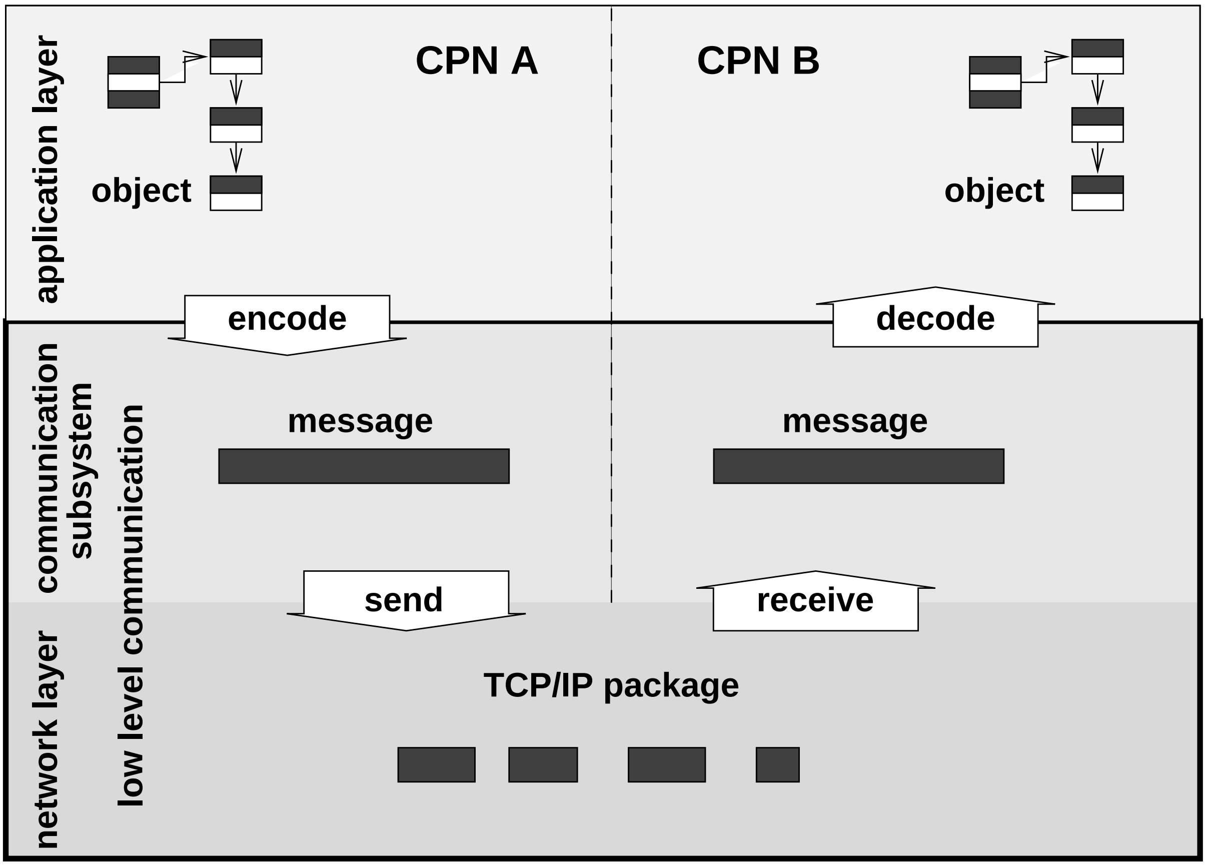
Border Length

$$l_b(p) = \frac{1}{\sqrt{p}}$$

$$B(p) = O(\sqrt{p})$$

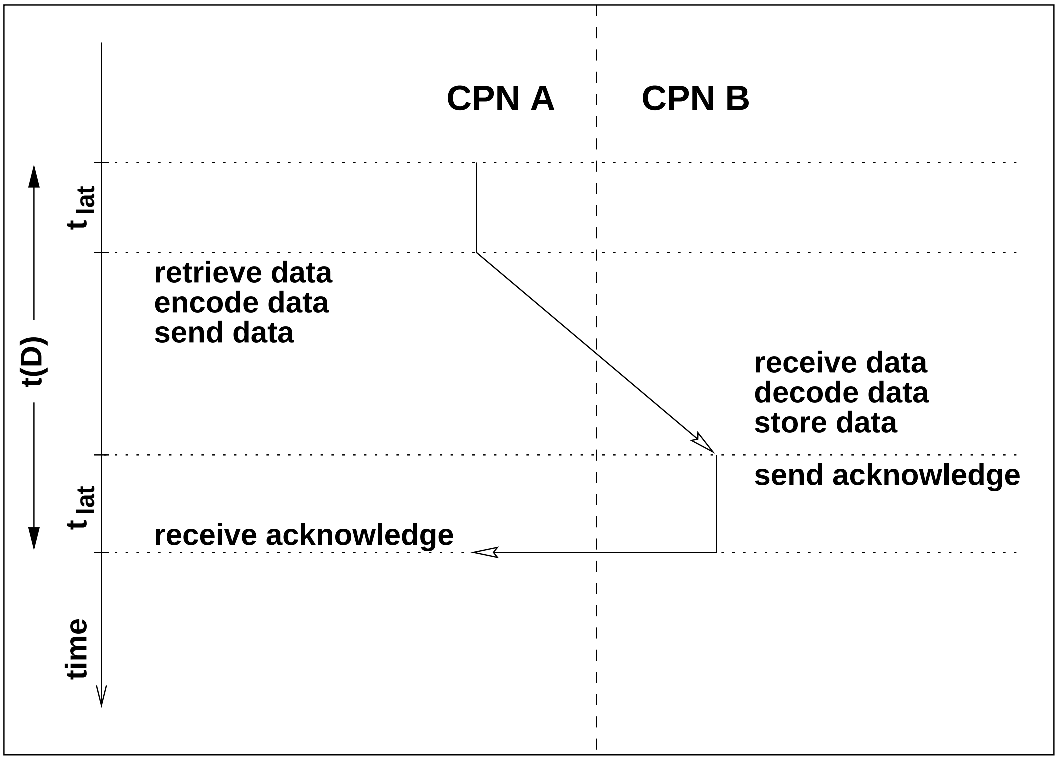
Number of boundaries does not scale!

Tracing a Boundary



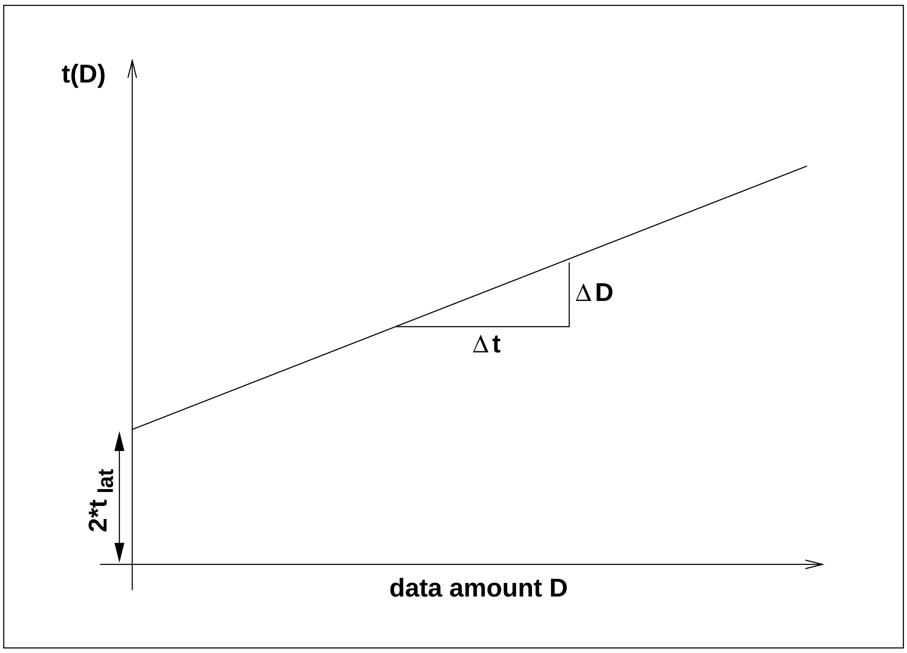
Measuring Application-level Latency and Bandwidth

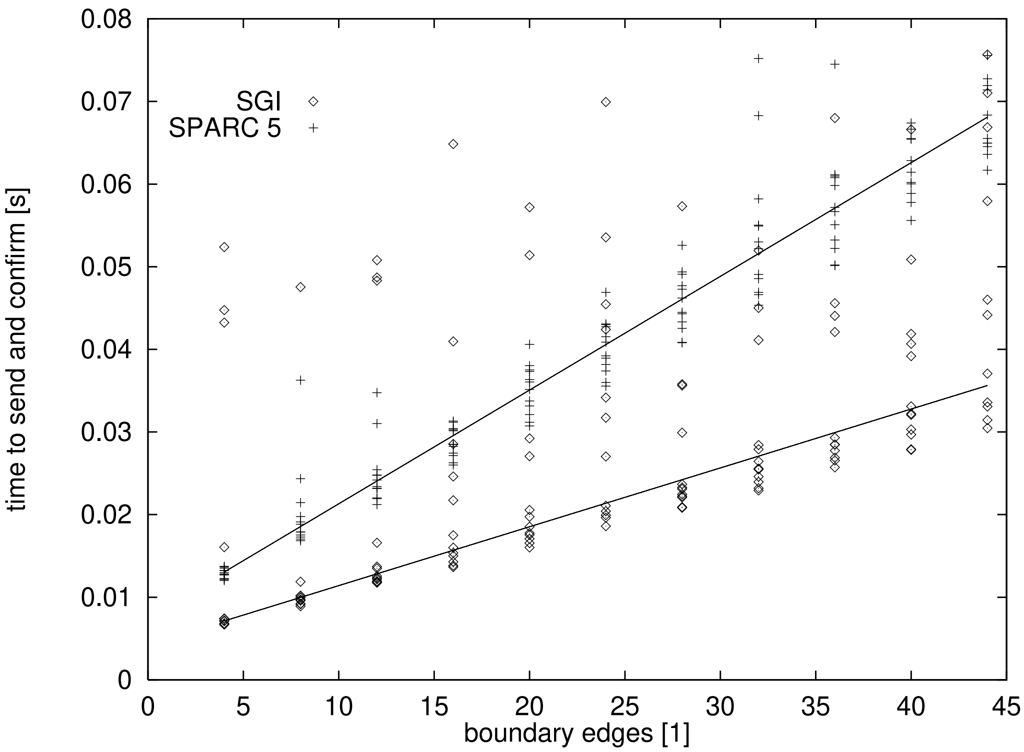
Measure the time $t(D)$ for retrieving, encoding, sending, receiving, decoding, and storing amount D data plus an additional acknowledge .



Estimating Parallel Efficiency

Measurements





Network Saturation I

Assumptions for application-level communication only hold, if communication network is not loaded otherwise!

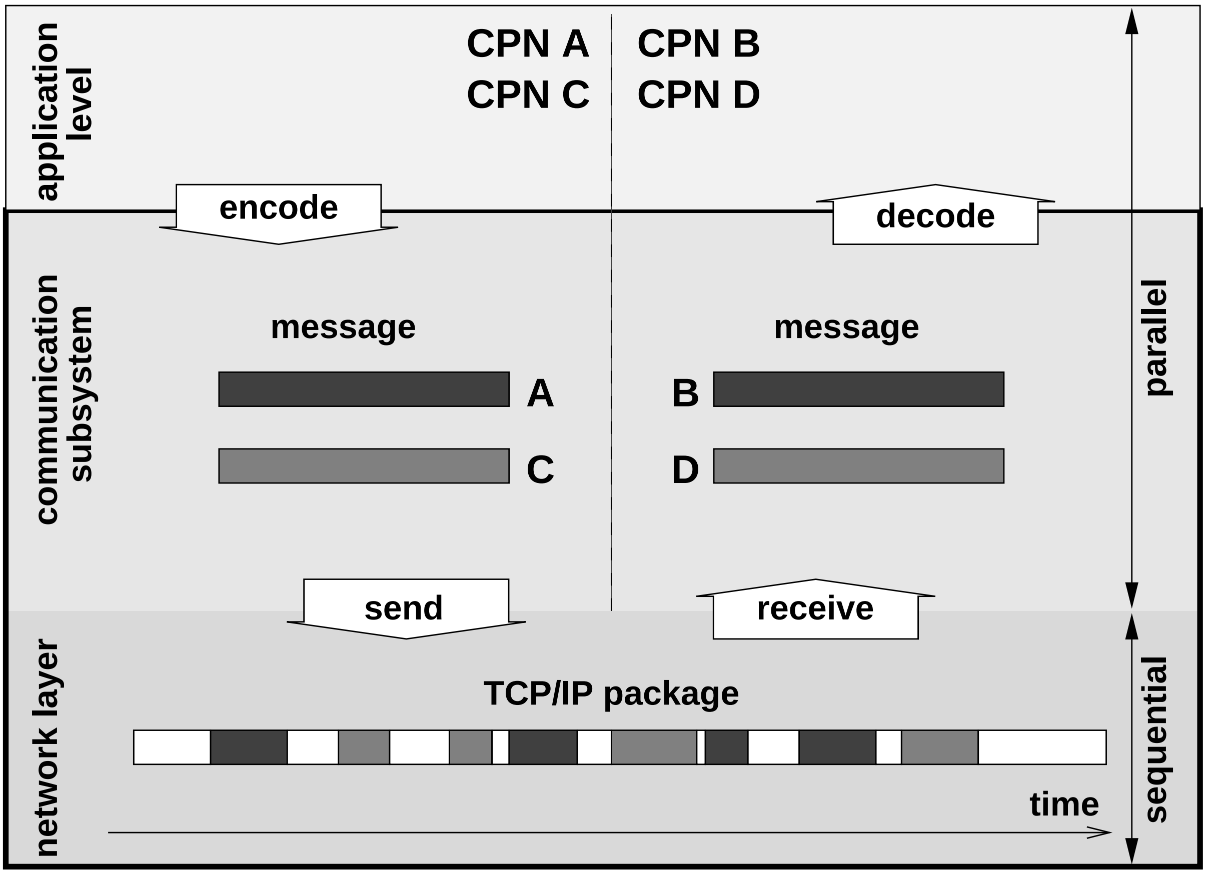
Multiple CPN have to share the same network bandwidth.

$\Rightarrow$

Above a certain number of CPN, communication performance may be dominated by low-level communication.

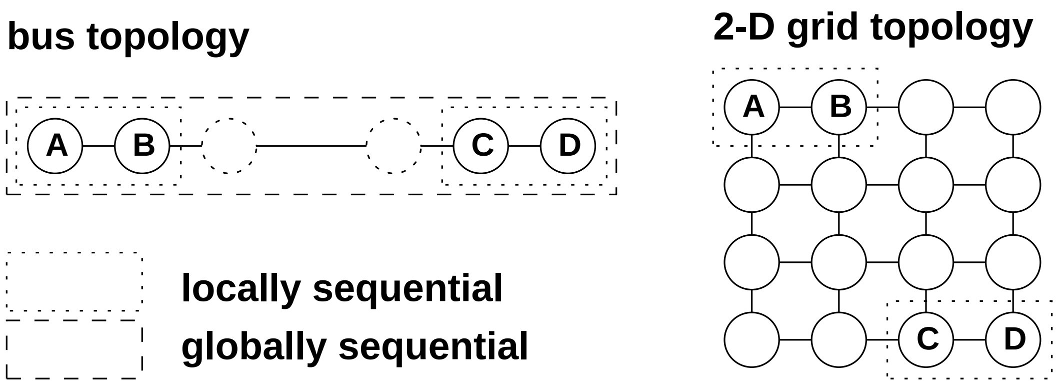
Depends on communication network topology!

Network Saturation II



Low-level communication scales
until
all gaps are filled!

Communication Network Topologies



Bus-Topology

- no difference between local and non-local communication
- all communication is sequential

2D-Grid-Topology

- local communication is sequential,
- but globally parallel

Traffic Simulation with 2D-Grid

Bus-Topology:

All boundaries between CPN have to be transferred sequentially

$\Rightarrow$

$$B_{seq}(p) = B(p) = O(\sqrt{p})$$

2D-Grid-Topology:

Only local boundaries have to be transferred sequentially

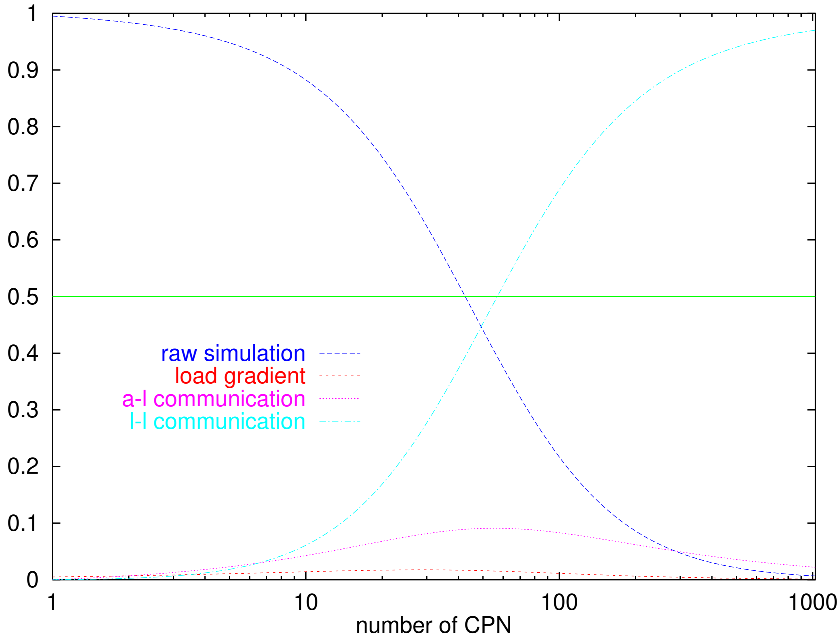
$\Rightarrow$

$$B_{seq}(p) \sim \frac{B(p)}{p} = O(1 + 1/\sqrt{p})$$

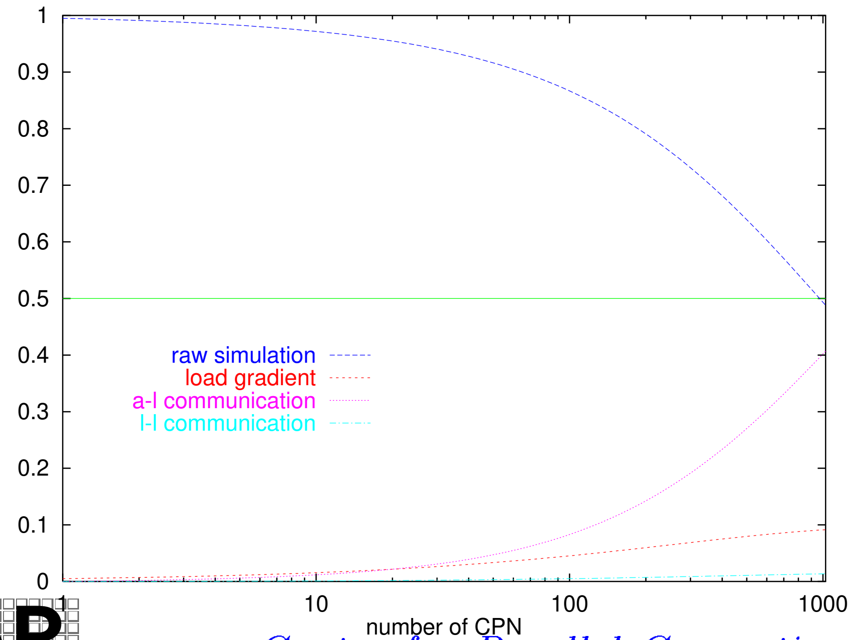
Boundary communication scales perfectly!

Efficiency

Sparc Cluster

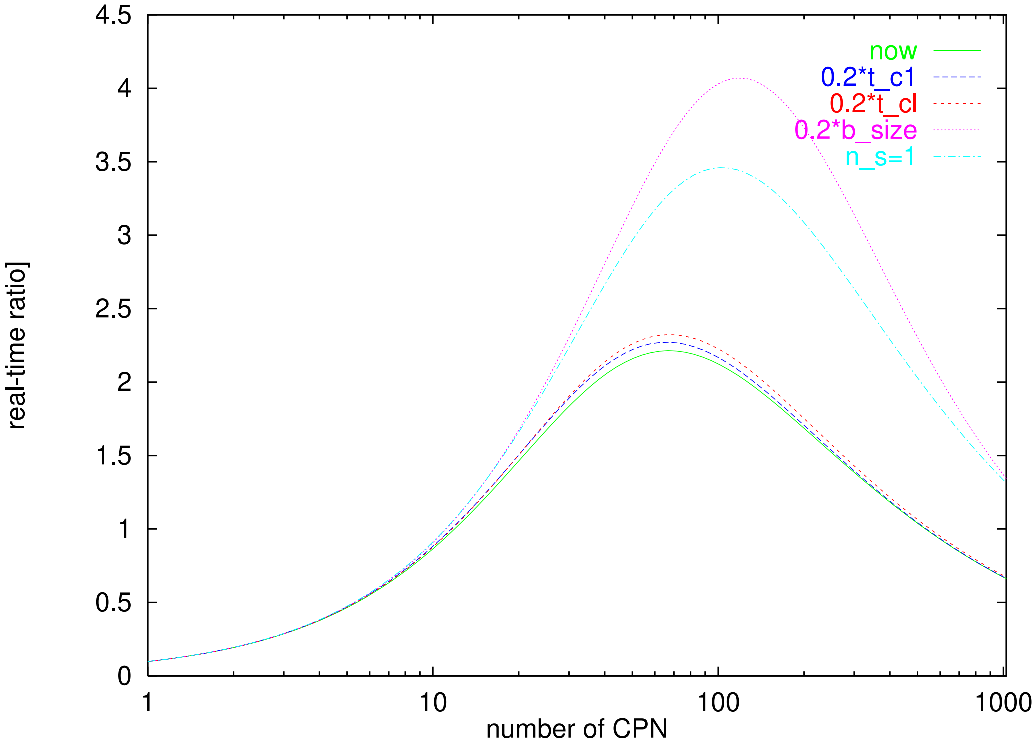


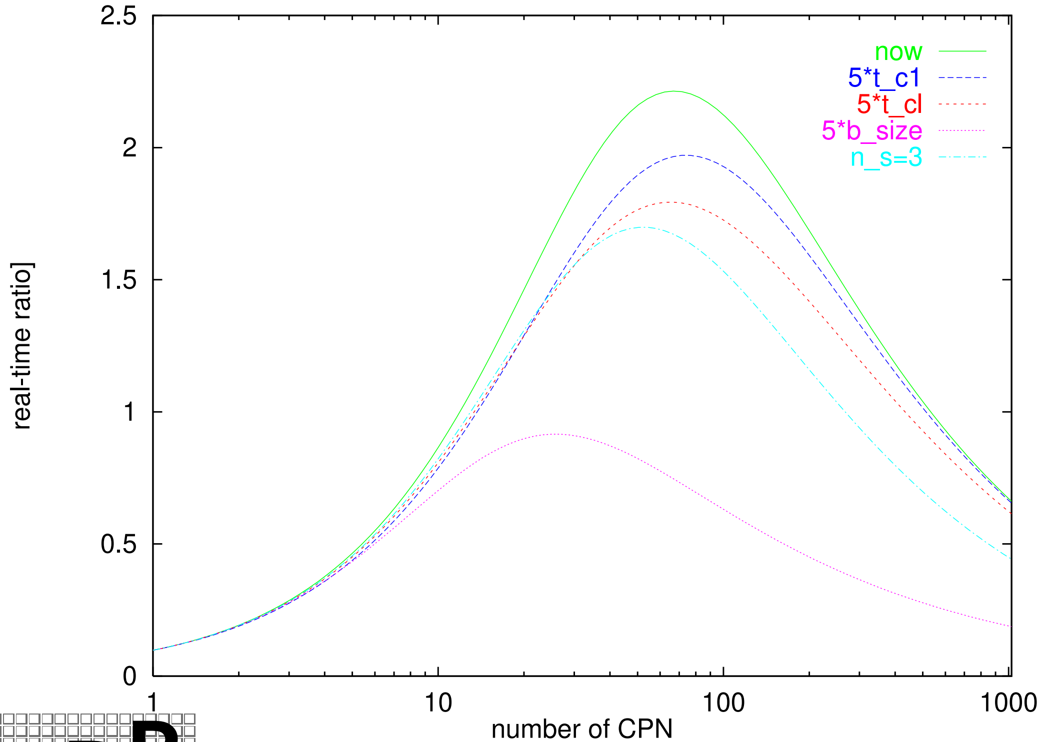
Paragon



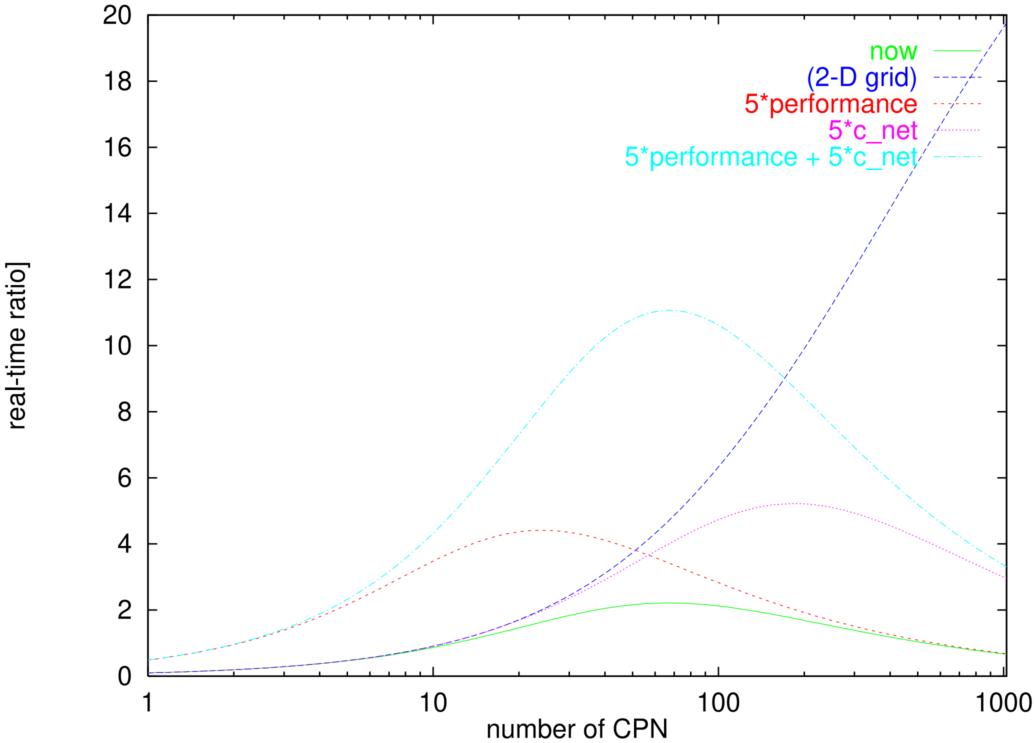
Estimating Parallel Efficiency

Software Modifications





Hardware Modifications



Guidelines I

- Keep boundaries as short as possible
  $\Rightarrow$ reduce low-level communication saturation

- Keep boundary handling simple
  - sort data items according to data types
  - use arrays instead of lists (at lowest level)
  - implement short "delivery paths" for messages

$\Rightarrow$ reduce application-level communication time

Guidelines II

- "anticipate" what boundary information is required instead of waiting for requests
  $\Rightarrow$ reduce number of sub-time-steps