Efficiency Metrics in a POP performance audit

EU H2020 Center of Excellence (CoE)

1 October 2015 – 31 March 2018

Grant Agreement No 676553

Efficiencies

The following metrics are used in a POP performance audit.

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

Global Efficiency (GE)

- The **Global Efficiency** describes how well the parallelization of your application is working.

- The **Global Efficiency** can be split into Parallel Efficiency and Computation Efficiency.

GE = PE * CompE

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

Parallel Efficiency (PE)

- The **Parallel Efficiency** describes how well the execution of the code in parallel is working.

- The **Parallel Efficiency** can be split into Load Balance Efficiency and Communication Efficiency.

PE = LB * CommE

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

Load Balance Efficiency (LB)

- The **Load Balance Efficiency** reflects how well the distribution of work to processes of threads is done in the application.

- The **Load Balance Efficiency** is the ratio between the average time of a process spend in computation and the maximum time a process spends in computation.

$$\mathrm{LB} = \frac{avg(tcomp)}{\max(tcomp)}$$

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

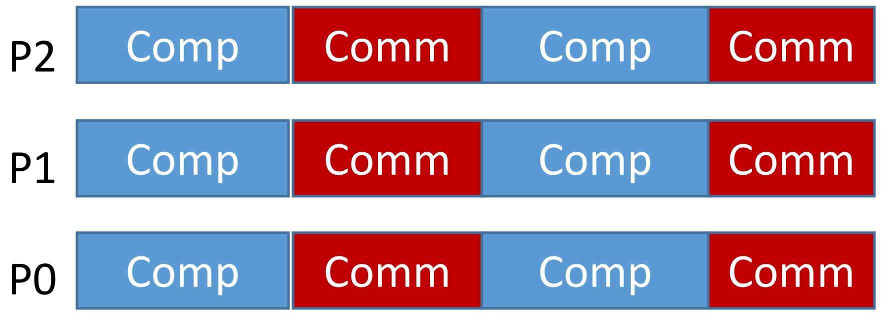
Load Balance Efficiency (LB)

- The **Load Balance Efficiency** reflects how well the distribution of work to processes of threads is done in the application.

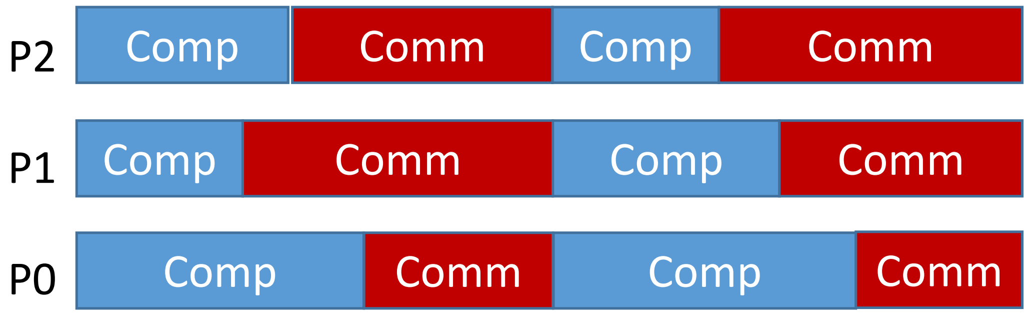
$$LB = \frac{avg(tcomp)}{\max(tcomp)}$$

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

Example 1: good load balance (LB = 100%)



Example 2: bad load balance (LB = 77%)



Communication Efficiency (CommE)

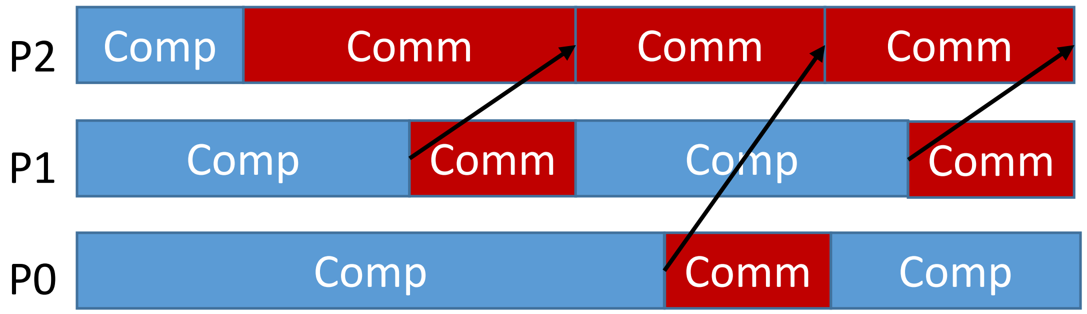
- The **Communication Efficiency** reflects the loss of efficiency by communication.

- The **Communication Efficiency** can be computed as

$$\max_{processes}\left(\frac{computation\ time}{total\ runtime}\right)$$

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

Example:



Compute	Communication	Efficiency
1 sec.	5 sec.	$^1/_6$
4 sec.	2 sec.	$^4/_6$
5 sec.	1 sec.	$^5/_6$

CommE = $^5/_6$ = 83%

Communication Efficiency (CommE)

- The **Communication Efficiency** reflects the loss of efficiency by communication.
- The **Communication Efficiency** can be split further into Serialization Efficiency and Transfer Efficiency.

CommE = SerE * TE

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

Serialization Efficiency (SerE)

- The **Serialization Efficiency** describes loss of efficiency due to dependencies between processes.
- Dependencies can be observed as waiting time in MPI calls where no data is transferred, because one required process did not arrive at the communication call yet.
- On an ideal network with instantaneous data transfer these inefficiencies are still present, as no real data transfer happens.

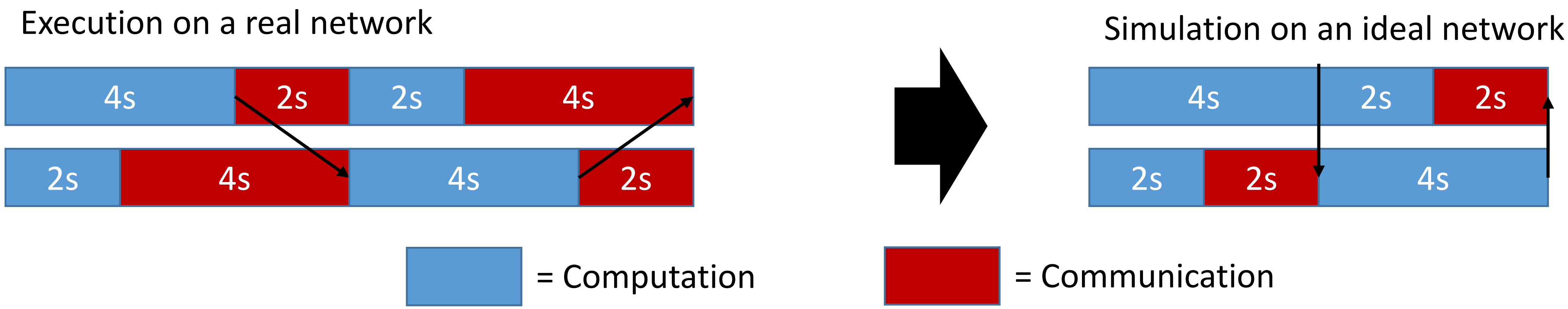
- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

Serialization Efficiency (SerE)

- On an ideal network with instantaneous data transfer these inefficiencies are still present, as no real data transfer happens.
- Serialization Efficiency is computed as

$$\max_{processes}\left(\frac{computation\ time\ on\ ideal\ network}{total\ runtime\ on\ ideal\ network}\right)$$

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling



SerE = $\frac{6}{8}$ = 75%

Transfer Efficiency (TE)

- The **Transfer Efficiency** describes loss of efficiency due to actual data transfer.

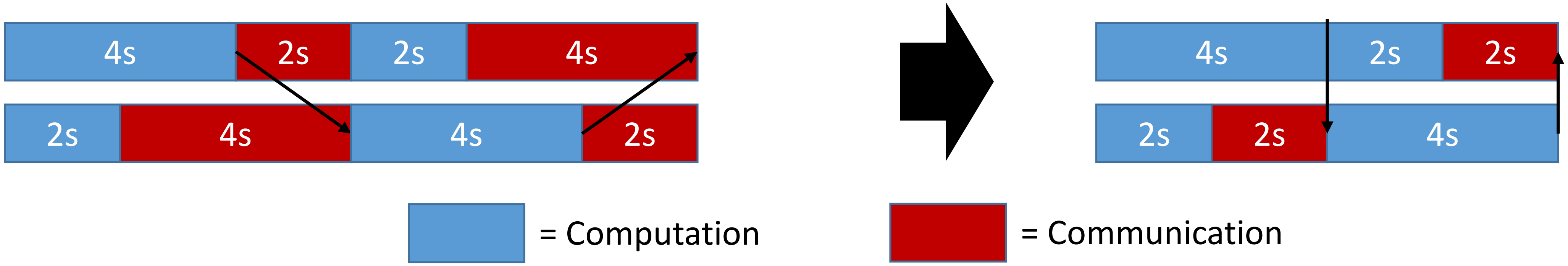
- The **Transfer Efficiency** can be computed as

$$\text{TE} = \frac{\mathit{total\ runtime\ on\ ideal\ network}}{\mathit{total\ measured\ runtime}}$$

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

Execution on a real network

Simulation on an ideal network



$$\text{TE} = \frac{8}{12} = 66.6\%$$

Computation Efficiency (CompE)

- The **Computation Efficiency** describes how well the computational load of an application scales with the number of processes.
- The **Computation Efficiency** is computed by comparing the total time spend in computation for a different number of threads/processes.
- For a linearly-scaling application the total time spend in computation is constant and thus the Computation efficiency is one.

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling

IPC Scaling / Instruction Scaling

- A low computation efficiency can have two reasons:
  1. With more processes more instructions are executed, e.g. some extra computation for the domain decomposition is needed.

     Instruction Scaling compares the total number of instructions executed for a different number of threads/processes.
  2. The same number of instructions is computed but the computation takes more time, this can happen e.g. due to shared recourses like memory channels.

     IPC Scaling compares how many instructions per cycle are executed for a different number of threads/processes.

- Global Efficiency (GE)
  - Parallel Efficiency (PE)
    - Load Balance Efficiency (LB)
    - Communication Efficiency (CommE)
      - Serialization Efficiency (SerE)
      - Transfer Efficiency (TE)
  - Computation Efficiency (CompE)
    - IPC Scaling
    - Instruction Scaling