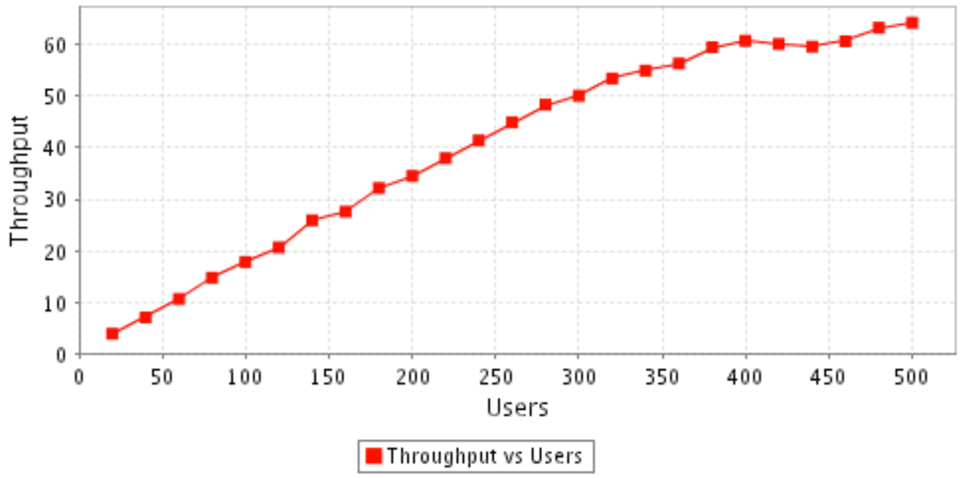
4. Performance Analysis of Parallel Programs

4.1 Performance Evaluation of Computer

User criteria:
- Small **response times**

Computing center criteria:
- High **throughputs**



4.1.1 Evaluation of CPU Performance

4.1.1 Evaluation of CPU Performance

The response time of a program A can be split into:

Freie Universität Berlin

4.1.1 Evaluation of CPU Performance

The response time of a program A can be split into:

User CPU time of A

4.1.1 Evaluation of CPU Performance

The response time of a program A can be split into:

User CPU time of A

System CPU time of A

4.1.1 Evaluation of CPU Performance

The response time of a program A can be split into:

User CPU time of A

System CPU time of A

Waiting time of A

4.1.1 Evaluation of CPU Performance

The response time of a program A can be split into:

User CPU time of A

System CPU time of A

Waiting time of A

4.1.1 Evaluation of CPU Performance

User CPU time of A

4.1.1 Evaluation of CPU Performance

User CPU time of A

$$T_{\text{U\_CPU}}(A) = n_{\text{cycle}}(A) \cdot t_{\text{cycle}}$$

t_{cycle} -> reciprocal to clock rate: T=1/f -> 2GHz = 1/(2*10^9)s = **0.5ns** (cycle time)

n_{cycle} (A)-> total number of CPU cycles needed for all instructions of A

4.1.1 Evaluation of CPU Performance

CPI (**C**lock cycles **P**er **I**nstruction)

4.1.1 Evaluation of CPU Performance

CPI (**C**lock cycles **P**er **I**nstruction)

$$T_{\mathrm{U\_CPU}}(A) = n_{\mathrm{instr}}(A) \cdot CPI(A) \cdot t_{\mathrm{cycle}}$$

Freie Universität Berlin

4.1.1 Evaluation of CPU Performance

CPI (**C**lock cycles **P**er **I**nstruction)

$$T_{\text{U\_CPU}}(A) = n_{\text{instr}}(A) \cdot CPI(A) \cdot t_{\text{cycle}}$$

n_{instr}(A) -> total number of instructions executed for A

4.1.1 Evaluation of CPU Performance

CPI (**C**lock cycles **P**er **I**nstruction)

$$n_{\text{cycle}}(A) = \sum_{i=1}^{n} n_i(A) \cdot CPI_i$$

n_i(A) -> is the number of instructions of type I_i executed for the program A
CPI_i -> number of CPU cycles needed for instructions of type I_i

4.1.1 Evaluation of CPU Performance

CPI (**C**lock cycles **P**er **I**nstruction)

Example: We consider a processor with three instruction classes I_1, I_2, I_3
containing instructions which require 1, 2, or 3 cycles for their execution.
We assume that there are two different possibilities for the translation of a
Programmi structions.

	Instruction classes				
Translation	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	Sum of the instructions	n_{cycle}
1	2	1	2	5	10
2	4	1	1	6	9

$$CPI_i = n_{\text{cycle}}(A) \Big/ \sum_{i=1}^{n} n_i(A)$$

$CPI_1 = 10/5 = 2$

$CPI_2 = 9/6 = 1{,}5$

Freie Universität Berlin

4.1.2 MIPS and MFLOPS

4.1.2 MIPS and MFLOPS

MIPS (**M**illion **I**nstructions **P**er **S**econd)

$$MIPS(A) = \frac{n_{\mathrm{instr}}(A)}{T_{\mathrm{U\_CPU}}(A) \cdot 10^6}$$

4.1.2 MIPS and MFLOPS

MIPS (**M**illion **I**nstructions **P**er **S**econd)

$$MIPS(A) = \frac{n_{\text{instr}}(A)}{T_{\text{U\_CPU}}(A) \cdot 10^6}$$

Drawbacks/limitations:
- Only considers the number of instructions.
- MIPS rate does not necessarily correspond to the execution time.

4.1.2 MIPS and MFLOPS

MFLOPS (**M**illion **F**loating-point **O**perations **P**er **S**econd)

4.1.2 MIPS and MFLOPS

MFLOPS (**M**illion **F**loating-point **O**perations **P**er **S**econd)

$$MFLOPS(A) = \frac{n_{\text{flp\_op}}(A)}{T_{\text{U\_CPU}}(A) \cdot 10^6} \ [1/\text{s}]$$

4.1.2 MIPS and MFLOPS

MFLOPS (**M**illion **F**loating-point **O**perations **P**er **S**econd)

$$MFLOPS(A) = \frac{n_{\text{flp\_op}}(A)}{T_{\text{U\_CPU}}(A) \cdot 10^6} \; [1/\mathrm{s}]$$

Drawbacks/limitations:
- Doesn’t difference between types of floating-points operations performed.

4.1.3 Performance of Processors with a Memory

$$T_{\mathrm{U\_CPU}}(A) = n_{\mathrm{cycle}}(A) \cdot t_{\mathrm{cycle}}$$

4.1.3 Performance of Processors with a Memory

$$T_{\mathrm{U\_CPU}}(A) = n_{\mathrm{cycle}}(A) \cdot t_{\mathrm{cycle}}$$

$$\Downarrow$$

$$T_{\mathrm{U\_CPU}}(A) = (n_{\mathrm{cycles}}(A) + n_{\mathrm{mm\_cycles}}(A)) \cdot t_{\mathrm{cycle}}$$

Freie Universität Berlin

4.1.3 Performance of Processors with a Memory

$$T_{\text{U\_CPU}}(A) = n_{\text{cycle}}(A) \cdot t_{\text{cycle}}$$

$$\Downarrow$$

$$T_{\text{U\_CPU}}(A) = (n_{\text{cycles}}(A) + n_{\text{mm\_cycles}}(A)) \cdot t_{\text{cycle}}$$

$\mathbf{n_{mm\_cycles}(A)}$ -> number of additional machine cycles caused by memory
accesses of A .