

# L5

---

- performance of many apps (29 as of 2006 spec) compressed into a single number
- spec uses geometric mean (GM)

$$\circ \sqrt[n]{sr_1 * sr_2 * ....sr_n},$$

where  $sr$  is exec time/reference time

reference time is the time taken to exec the program on the SPEC benchmark system

- Arithmetic mean is also popular but not good
- Weighted A.M. is better.

## Execution time after improvement

---

- Execution time unaffected (which is part of the program that cannot be parallelised) + {Exec time Affected/ Rate of improvement}
- If a certain part of a program eg. multiply has 80% of the exec time, we can't achieve - like a 5 fold enhancement, if 80% can't be parallelized

Amdahl's law says

Let,

$T_b$  -> execution before improvement

$f$  -> fraction that can be improved

$T_a$  -> After improvement

$s_f$  -> Speedup factor

$$T_b = (1 - f)T_b + fT_b$$

$$T_a = (1 - f)T_b + f\frac{T_b}{s_f}$$

$$1 = (1 - f)s_o + fs_o/s_f$$

$$\text{where } s_o = T_b/T_a$$

$$s_o = \frac{1}{((1-f)+f/s_f)}$$

max speedup will go to  $1/(1-f)$

$sf \propto 1/(\text{no. of cores})$  in theory, but actually its performance is lesser

the theoretical usage drops after some time and is shown by the amdahl's law graph

## Instructions

---

mIPS - million instructions per second

MIPS - Microcomputer without interlocked pipelined stages

- John L. Hennessy at Stanford University developed MIPS
  - inspired most architectures since the 80s
- Arithmetic
- Data transfer
- logical
- conditional branch
- conditional jump

32 registers -> \$0 to \$31

register no.	register name	functionality
\$0	\$zero	hardwired to 0
\$1	\$at	<ul style="list-style-type: none"><li>• reserved by the assembler</li><li>• used to handle large constants</li></ul>
\$2,\$3	\$v0, \$v1	function return args, values of temp results and expo values
\$4 - \$7	\$a0- \$a3	argument passing regs to the function
\$8 - \$15	\$t0 - \$t7	temporary regs
\$16 - \$23	\$s0 - \$s7	<ul style="list-style-type: none"><li>• saved registers, used for calling one fn to another,</li><li>• preserved across call</li></ul>
\$24-\$25	\$t8 - \$t9	temporary
\$26 - \$27	\$k0 - \$k1	reserved by OS for exception handling
\$28	\$gp	storing global variables
\$29	\$sp	stack pointer
\$30	\$fp	frame pointer
\$31	\$ra	return address