Measurement issues

From gathering numbers to gathering knowledge

Dr. A.P. Kostelijk (Ton), PDSL

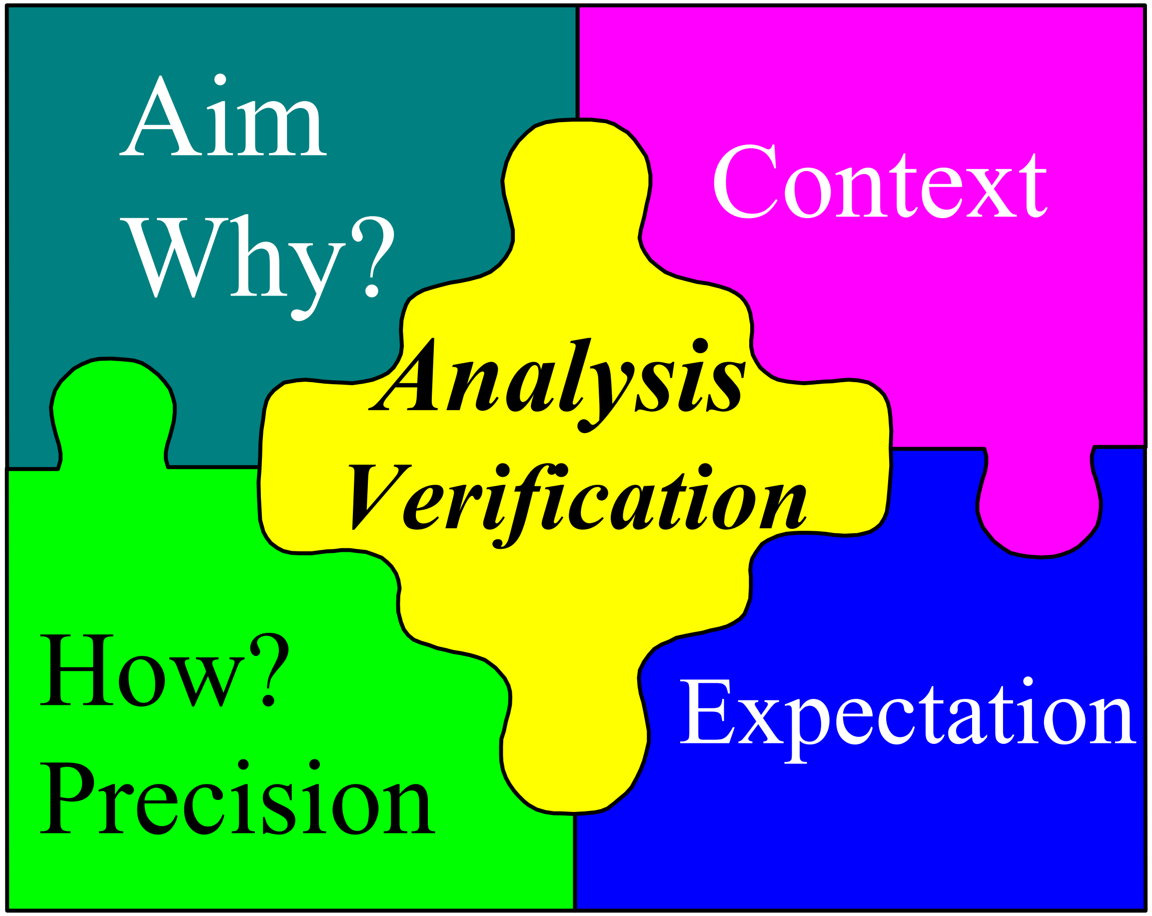
Contents

- Discussion
- Pieces of the puzzle
- Multi-level performance modelling

Example. Discussion.

```
int main() {
   int t1,t2, n=10;
   t1 = OS_time();
   for (i=0; i<n; i++)
      f(0);
   t2 = OS_time();
   return (t2 - t1)/n;
}
```

Pieces of the puzzle



Aim

- Why measure something ?
- What is the underlying model?
  - Without a model one is collecting meaningless figures.
- Guestimate, estimate, measure, …?
- Min / max / typical?
- Feasibility: min / max reasoning (close in)

Determine context / state (static)

Version of

- source-code
- operating system
- compiler(s)
- hardware
- dataset operated on
- compiler settings
- linker settings
- hardware settings (Board-support package)
- other HW bus traffic
- memory configuration

Determine context / state (dynamic)

- Cache — Looping optimism.
- Network — Silence or busy.
- Disks — Jitter due to move & rotate.
- Higher level other processing — Other threads, or interrupts (Profilers)

Expectation

- Experience with and understanding of the old system is key
- Use a model
- Balance effort by granularity, guided by analysis and relative importance.
- Simplify.

Pitfall
Measuring

- Measuring overhead ignored
- OS-timer ticks: unsave
- Forget asynchronous elaboration

Overhead

- Bus protocol
- Bus congestion
- Array indexing overhead
  (usually solved by compiler)
- Loop overhead
- Function-call
- Thread / interrupt switch
- Measurement

Measurement precision

- Use HW timer register (Limit range!)
- IO-pin toggling with Logic Analyser
- OS-timer
- Profilers

- Precision: 1 instruction, or cache-line updates
- Idem, plus Logic An. Precision
- 1 OS-tick ≈ 10 ms
- Diverse

Measurement precision

When subtracting measurement results...

- $X = 10.0 \pm 5\%$, $Y = 8.0 \pm 5\%$
- $X - Y = 2.0 \pm 45\ \%$.

Statistics out of scope.

Last but not least: m/µs, byte or bit, clock cycles of what? Be explicit on units.

Example

Idle process measurements:

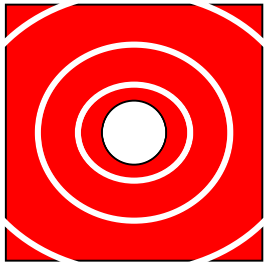
- OS-based profiler that counts the time each task has run?

Missing (part of) task-switch overhead, interrupts (5 - 25 % load).

Instruction i++ Counter in lowest priority task might be a better idea.

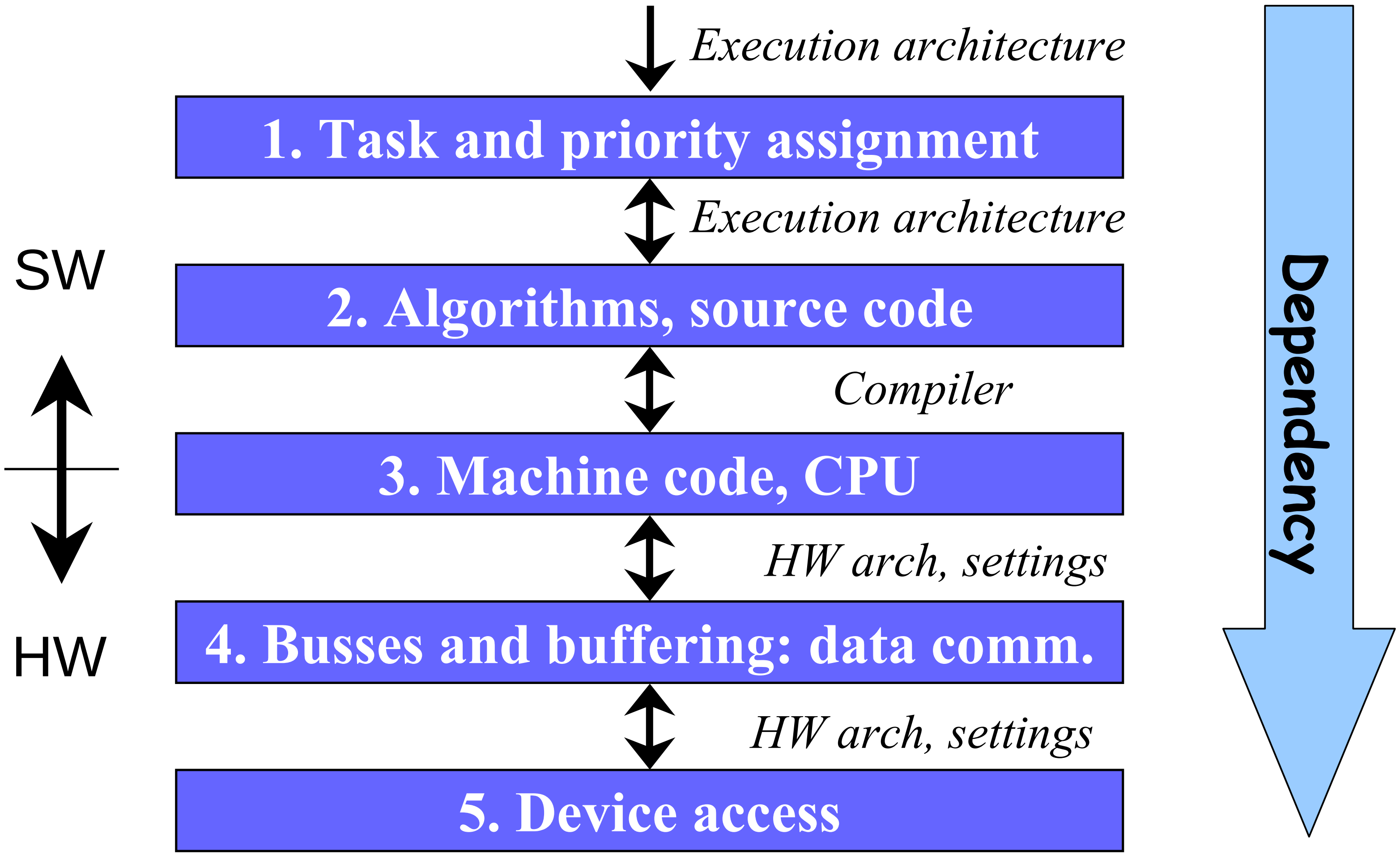
Analysis & Verification

Modelling is the heart of the matter

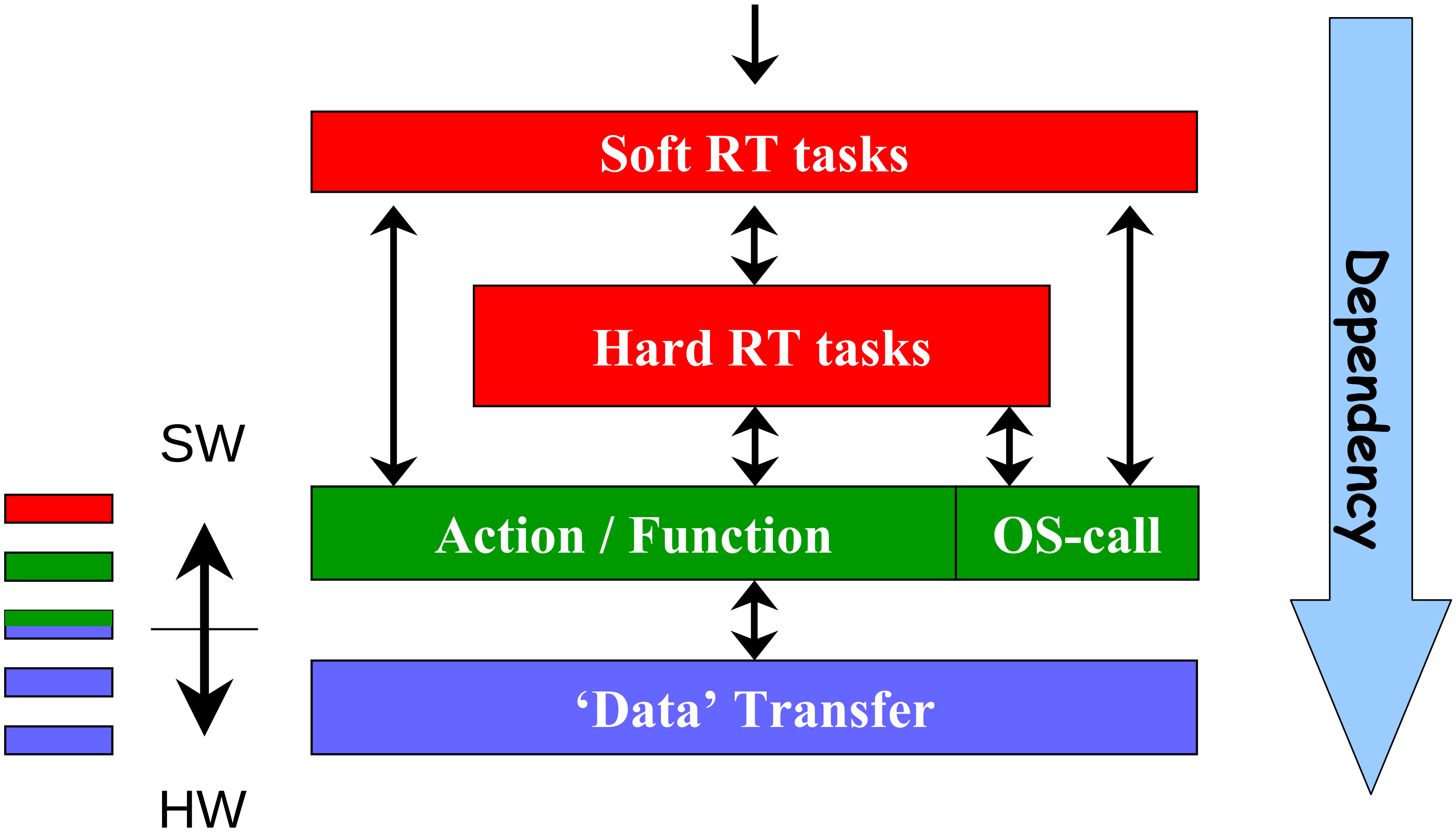


and therefore subject of the rest of this presentation.

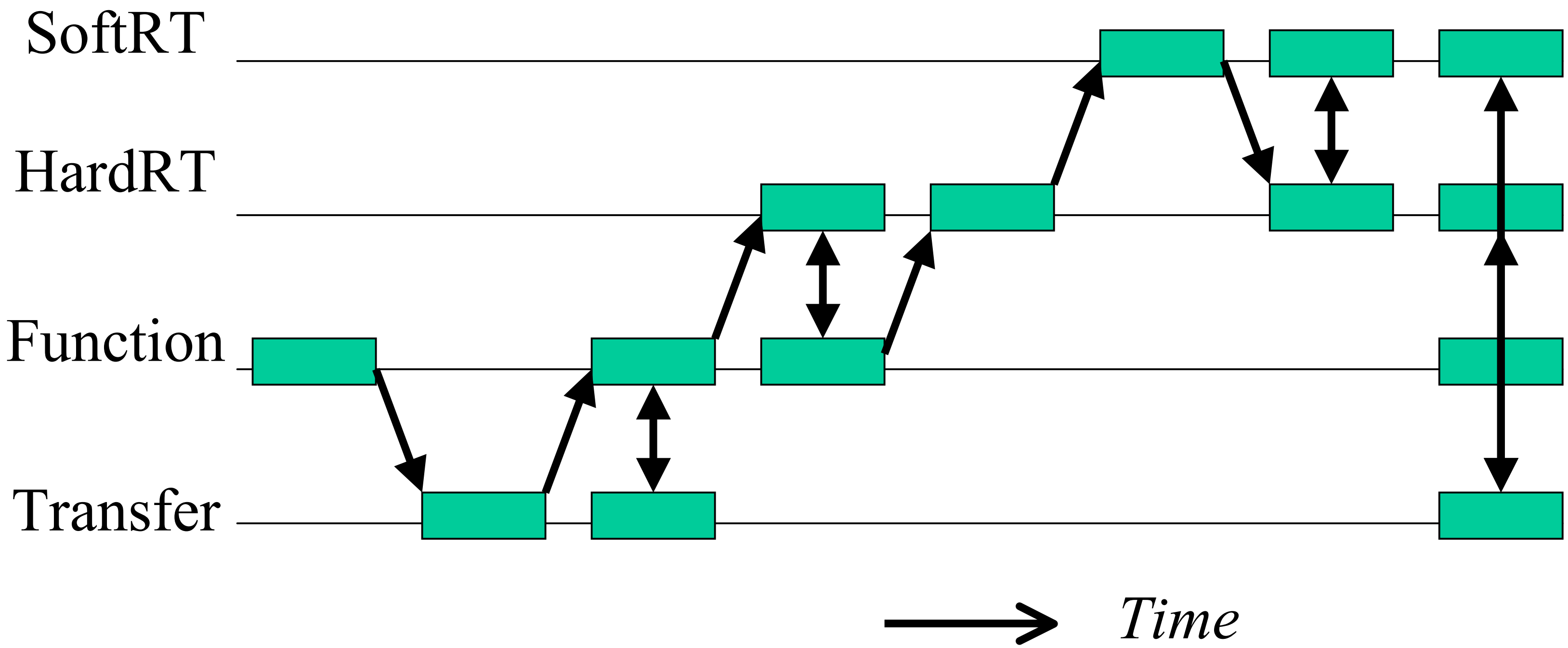
System levels of execution



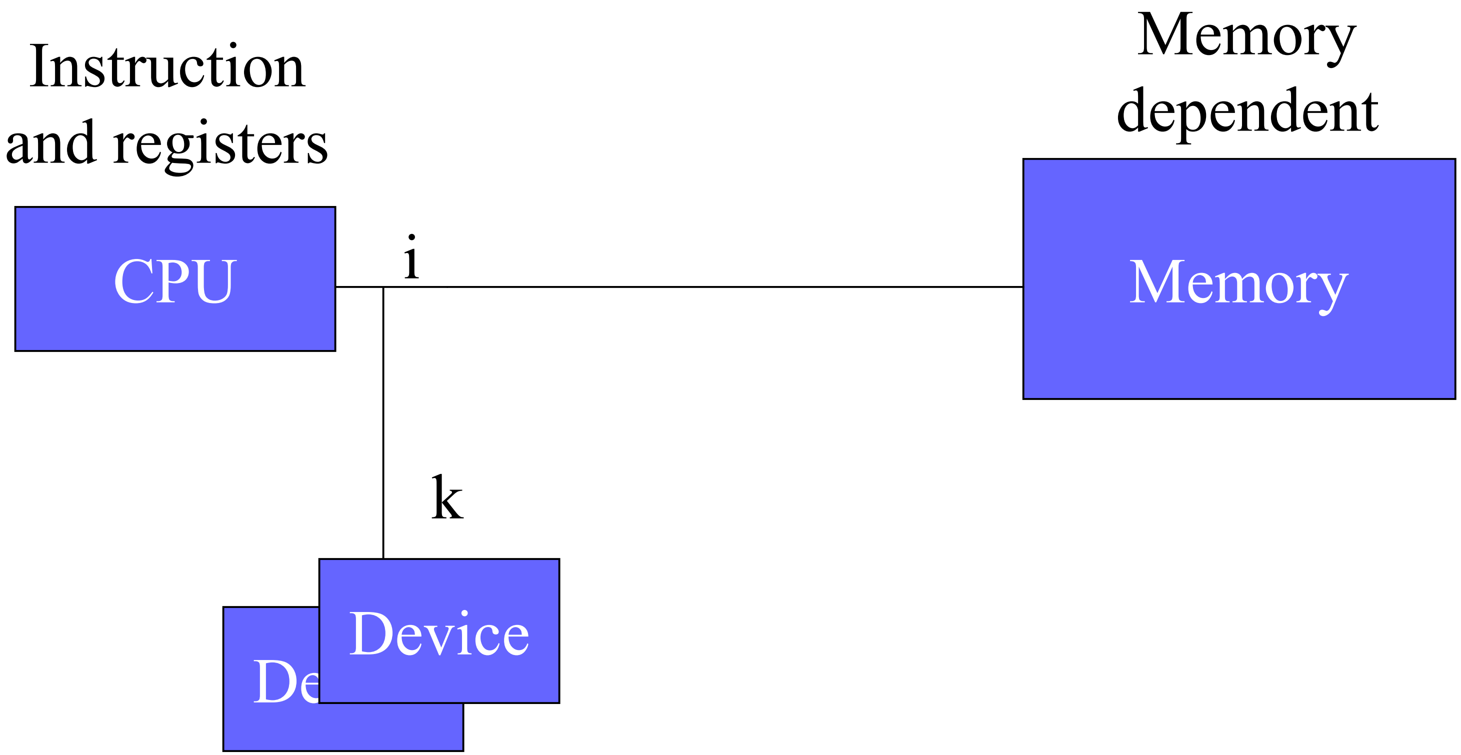
Software Latency models



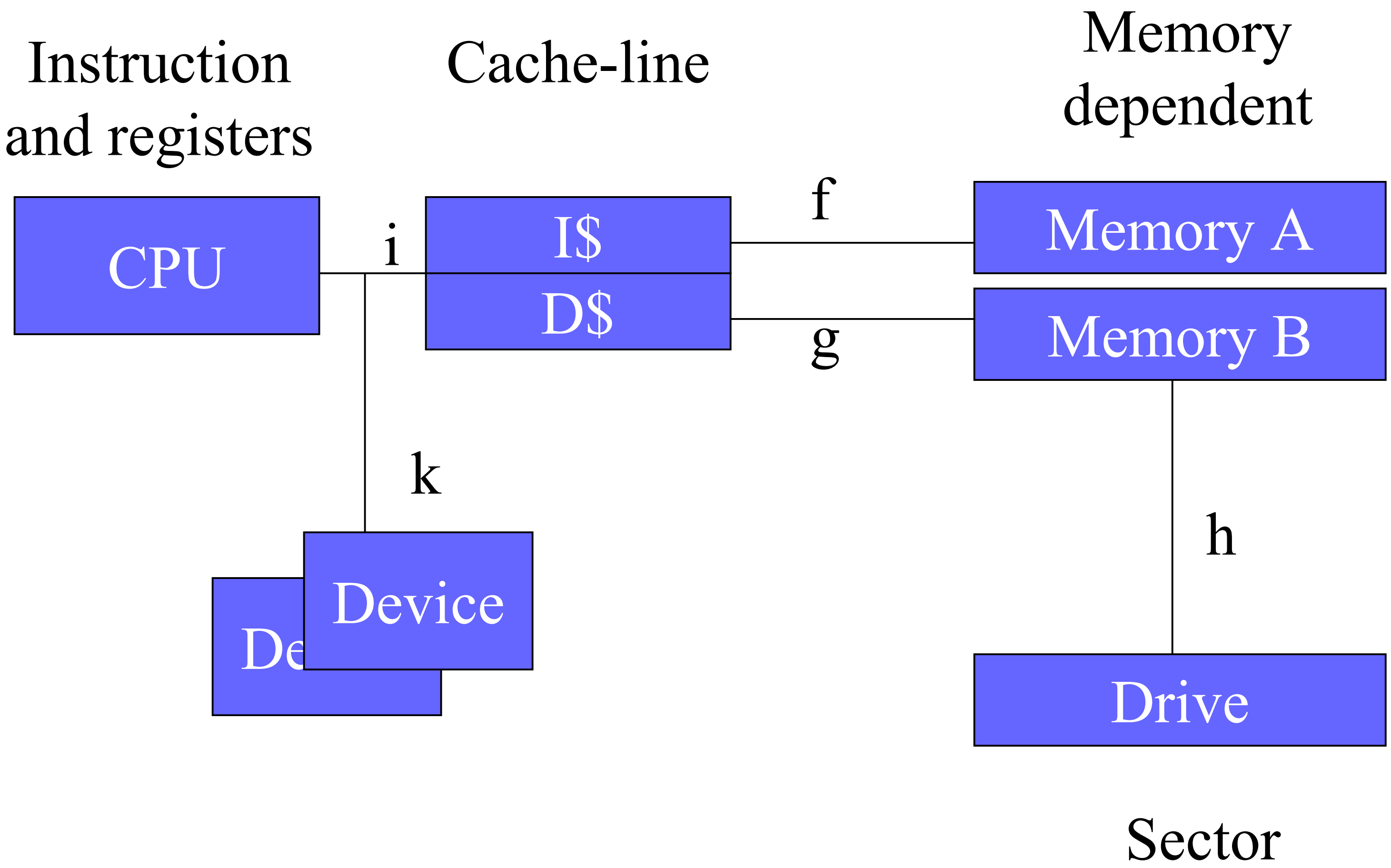
Typical iteration in time



‘Data’ transfer latency model



‘Data’ transfer latency model



Data transfer latency model

- Major input to understand HW/SW bottleneck
- Indicates for a given cpu-instruction, and context, the cpu-delay + cpu-stall cycles.
- Derive model by measuring a forced predefined number of bus-transactions.

  Example:

  ```
  I$_flush;measure;1000 nops; measure;
  ```

- Focus on typical transfers.

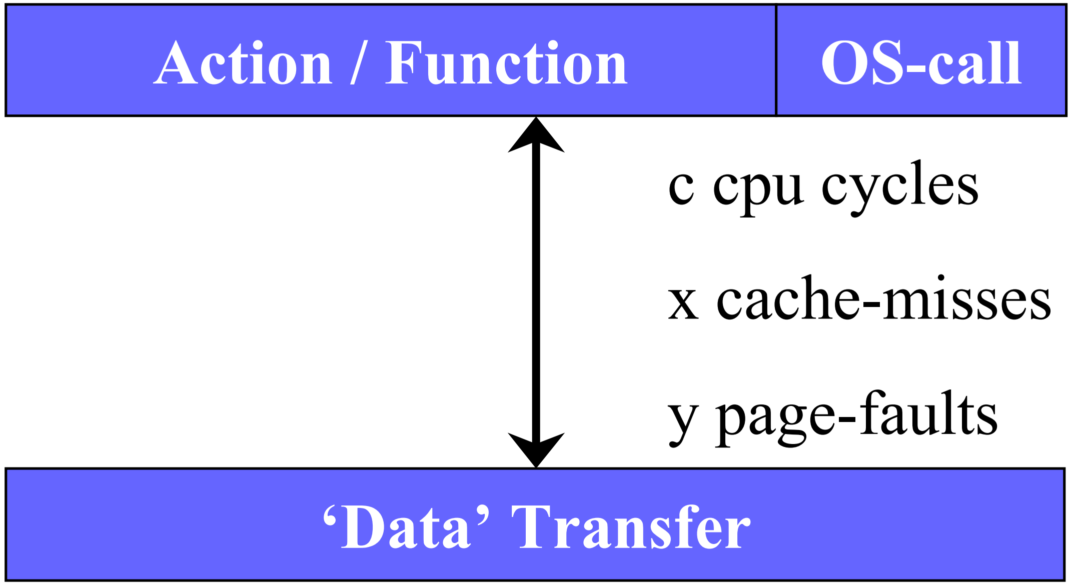
Data transfer - pitfalls

- Incorrect HW-settings.
- Unnoticed bus transfers.
- CPU-instruction pipe-line stalls.
- Overhead due to measurement.
- Precision is delicate. Subtraction of timing overhead is often required.
- Effect of large cache-lines

Function / OS-call

- Indicates for a given function, and context, the total run-time (latency).
- This includes interrupt routines and -overhead.
- Major input for e.g. RMA.

Function / OS-call & Data Transfer



$T = c + x.f + y.h$ $\qquad$ $T = 1000 + 50 * 10 + 5 * 1000$

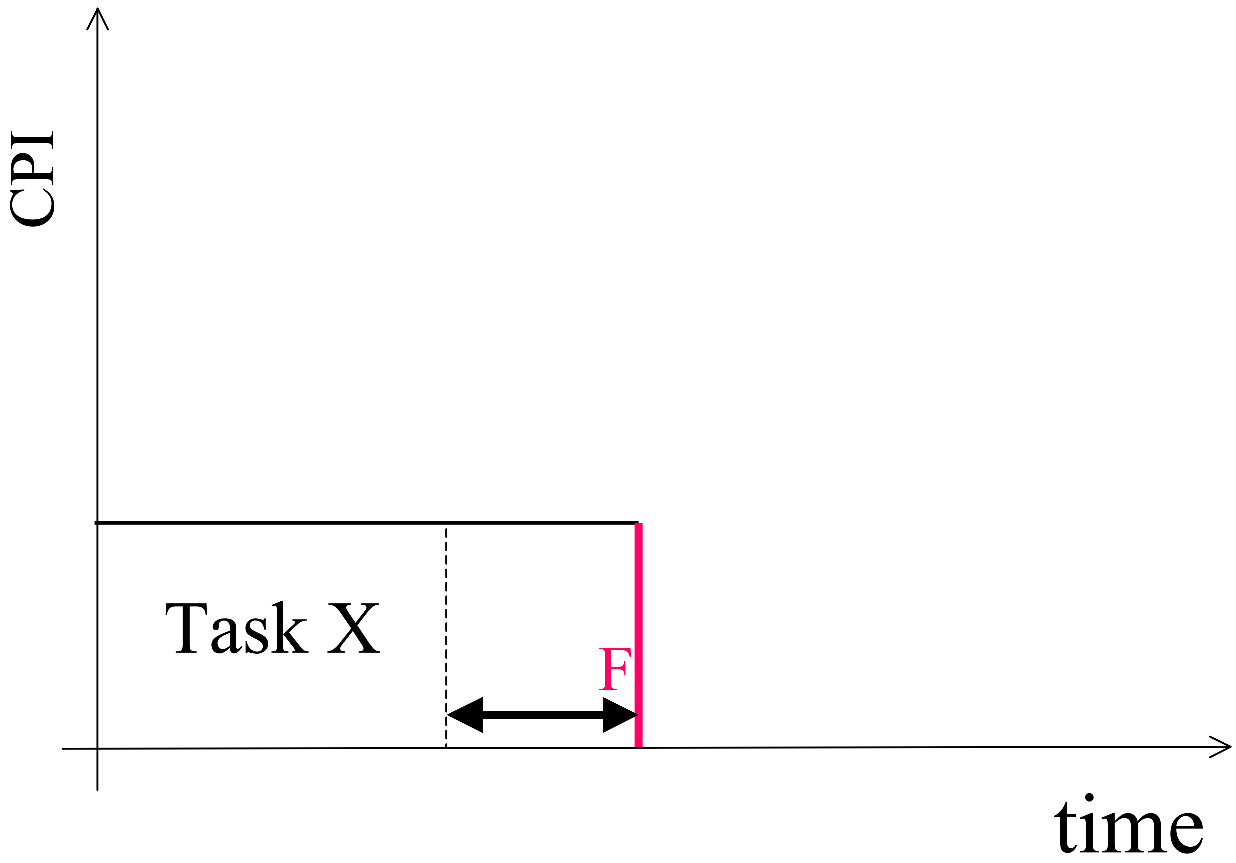
Function / OS-call / Transfer

- Simplified model:
- CPI: cycles per instruction
  - statistical number of cache-misses
  - For a MIPS: 3 - 10 typically for control code

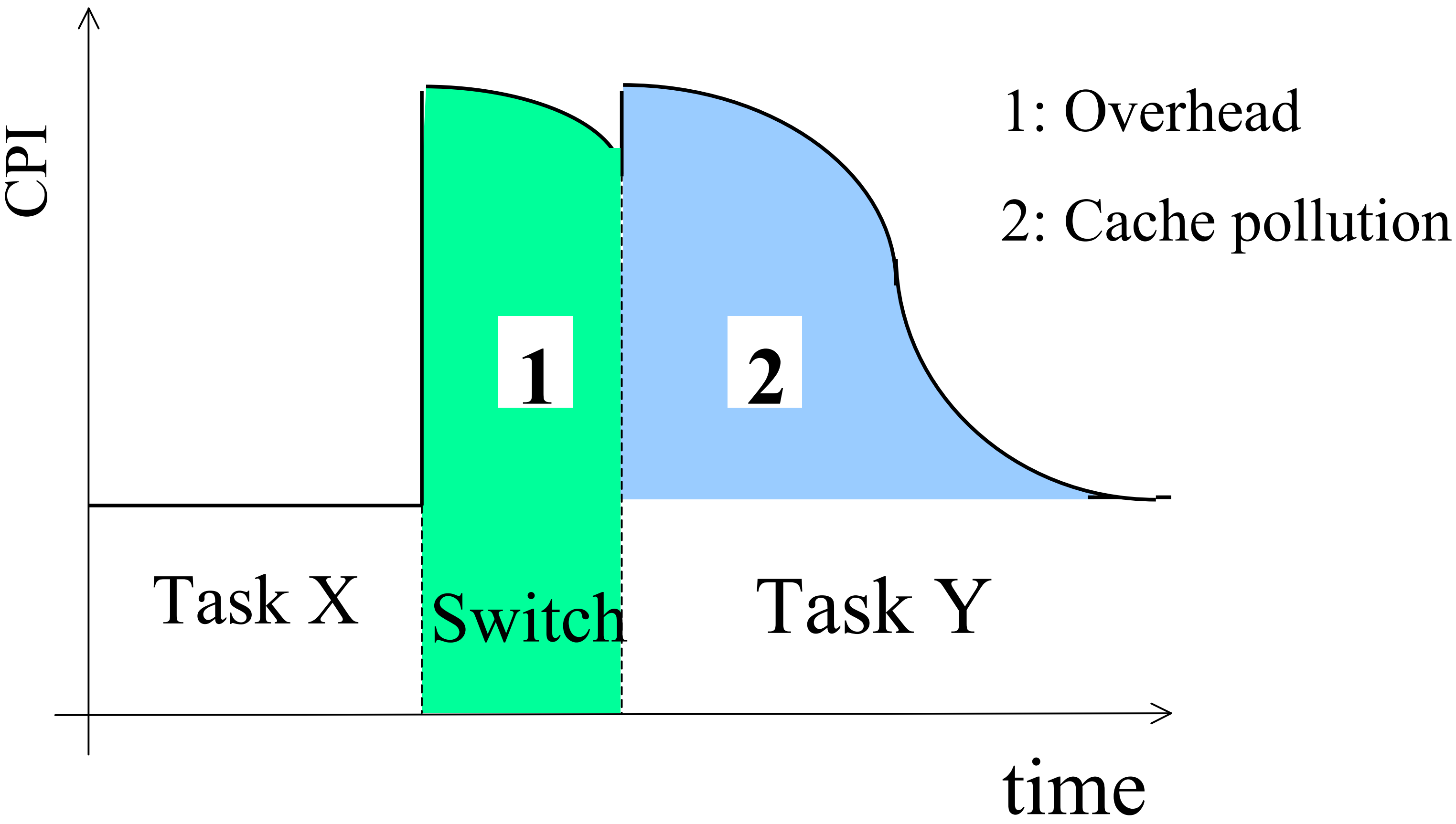
Function / OS-call pitfalls

- Timer precision.
- Cache-line trashing.
- Inconvenient compaction of data
  (e.g. char / int / char, iso. Char/char/int)
- Link-address may influence 5 - 15%
- Additional busload interference.
- Forget write-backs in a loaded system.
- Hidden taskswitches and interrupts.

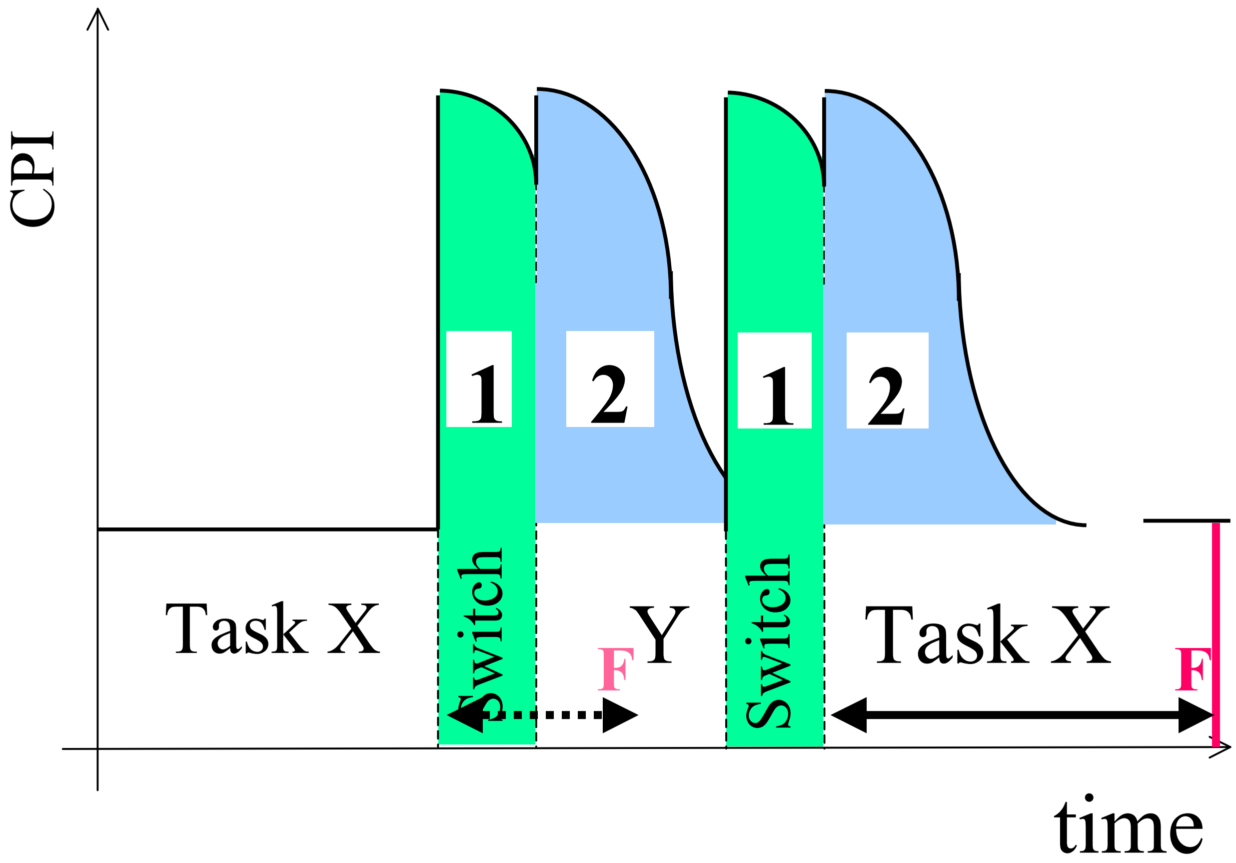
Task switching



Task switching



Task interference



Task interference

For small interruptions y
context switch and cache-pollution

- Resulting load = (switch + f_{load}) x 2.

Hard RT tasks

- From high-priority (interrupts) to middle-priority (threads), successive function processing measurements and analysis (e.g. RMA) are done to check the timing requirement feasibility.
- Include Thread switches, Interrupts pollution, Mutex-latencies.
- Example: RT-streaming platform

Soft RT tasks

- For middle to low priority threads, do response-time measurements where the rest of the system is also fully active.
- Measure relevant jitter in response-time by understanding the underlying causes.
- Trace substantial losses due to overhead.

Conclusion

- Measurements serve to build a reliable predicting multi-level time model of a system.
- Analysis, verification and simplification are major means to reach this goal.
- Without aim, context, expectation and precision, the actual outcome of a measurement is useless.