@fudef

L.EIC

#3.2

Computação Paralela e Distribuída

Cheat sheet for the exam

1. Parallel Programming

PAPI

Get statistics from the CPU (used in Lab 1)

Sensors: PAPI_L1_DCM ; PAPI_L2_DCM ; ...

L1 and L2 cache → Cache Locality

Performance Metrics

- **MIPS** : Million Instructions Per Second
- **FLOPS** : Floating-point Operations Per Second
- R_{peak} : Theoretical PFlop/s
- R_{max} : Max. "LINPACK" performance achieved
- N_{max} : Problem size for achieving R_{max}

Amdahl Law

$$Speedup = \frac{T_1}{T_P} = \frac{1}{\frac{1-s}{P} + s}$$

- P : # processors ($P \geq 1$)
- s : % of sequential code ($0 \leq s \leq 1$)

Open MP

	parallel	for	sections	single (critical zone)	parallel For	parallel sections
Execute in parallel only if <exp> is true — if(<exp>)	✓				✓	✓
All shared variables must be explicitly declared — default()	✓				✓	✓
Variables are duplicated with local access — private(<list>)	✓	✓	✓	✓	✓	✓
Private variables initialized with original v. — firstprivate(<list>)	✓	✓	✓	✓	✓	✓
Private var. from last iter. gives value to orig. — lastprivate(<list>)		✓	✓		✓	✓
Variables shared by all threads — shared(<list>)	✓	✓			✓	✓
All operations are saved to the end — reduction(<op>:<list>)	✓	✓	✓		✓	✓
Copy original value into a threadprivate var. — copyin(<list>)	✓				✓	✓
No need for synchronization at the end — nowait		✓	✓	✓		
Loop iterations must be executed in order — ordered		✓			✓	
Load balancing: static, dynamic, guided or runtime — schedule(<type>,<N>)		✓			✓	

UDP

- Based on messages (connectionless)
- Datagrams up to 65535 bytes
- No reliability or flow control
- Supports multicast

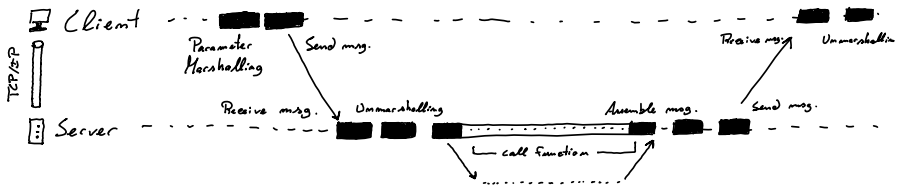
TCP

- Based on streams (connections)
- Made of 3 phases: setup, data exchange and teardown
- Ensures reliability and flow control
- Communication only between two endpoints

End-to-End Argument

"If you have to implement a function end-to-end don't implement it on the lower layers unless there is a compelling performance enhancement.

Flow



Transparency

- Due to platform heterogeneity, receiver converts messages to their format
- References are passed via call-by-copy/restore

⚠ A client cannot identify a fault with the absence of a response

Presence of faults

- At-least-once: If executed many times or none
- At-most-once: If executed once or none
- Exactly-once: Not always feasible to ensure

State vs Stateless

State

- If client crashes, resource is allocated forever → leaking
- Protocols need to give information to relate to the previous communication → Cookie

Stateless

- Operations may not be idempotent
- Protocols need to give all the information needed

Security

Due to data confidentiality, it is needed to encrypt data transmitted over the network.

Concurrency

- **Threads** | Easier to implement with SMs, but slower
- **Events** | Need non-blocking I/O, but faster

4. Fault Tolerance

Reliability

- $R(t)$ | Probability that a system has not failed until t
- MTTF | Mean Time To Failure

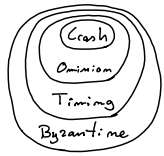
Availability

Assumes that a system may be repaired after failure

- MTTR | Mean Time To Repair

$$\alpha = \frac{MTTF}{MTTF + MTTR}$$

Failure Models



| Component suddenly does not respond to any input
| Component does not respond to some inputs
| Component does not respond on time
| Component behaves in a totally arbitrary way

5. Leader election

Specification Properties

- Safety: Something "bad" will not happen
- Liveness: Something "good" must happen

G.M. Specification

- **State**: {
  - .s = Down | Election | Normal,
  - .e = < coordinator_mode >,

  }
- **Assertions**:
  - $\forall_{i,j}$: $S(i).s = S(j).s$ = Normal
    ⇓
    $S(i).e == S(j).e$
  - If no failures occur, the system will eventually have:
    - All modes with .s = Normal
    - A mode i such that $S(i).e = i$

Bully Election Algorithm

- In the case of a mode **failure**:
  - A process initiates a new election if a leader or a candidate failed;
  - Nothing if other process failed.

Ex: If P_3 does not receive AUT:
- P_3 starts an election
- HALT sent by P_3 and P_5
- P_6 sets NL_3 and NL_5
  - NL_3 first → Discard due to .s = Normal;
  - NL_5 first → Discard due to .e ≠ Halt origin.