CS 856 – Latency in Communication Systems

Winter 2010

Latency Techniques: Reduction vs. Mitigation

Overview

- Addendum – Latency Challenges
- Latency Reduction
- Latency Mitigation
- Other Approaches

Addendum: State Synchronization

- e.g. routing state
- e.g. synchronous multicast
- e.g. CENTAUR

Parallel Execution

- replace sequential computation with parallel
- reduce processing time

BUT:

- watch Amdahl's Law: 1/(P/S + 1- P) -> 1/(1-P)
- synchronization needed!
  - details of work segmentation important
  - example: packet reordering in Internet

Multi-Threading

- blocking, stalling
- context-switch to other thread

BUT:

- design complexity
- execution cost

Synchronous Execution

- avoid gratuitous blocking
- synchronous call stack

BUT:

- sequential execution
- design flexibility?

Scheduling

- identify latency-critical tasks
  - processing, network flows
- control latency for certain tasks

BUT:

- runtime overhead
- other tasks?

Pipelining

- send multiple requests, before replies arrive
  - e.g. TCP reliable transmission
  - e.g. HTTP pipelining

BUT:

- buffering
- loss and retransmission

Caching

- ***the*** latency mitigation technique
  - memory, disk, network names, web/content

BUT:

- cache consistency
- authenticity
- other security implications?
  - unauthorized access

Coding

- order data transmission for incremental display
  - e.g. streaming
  - e.g. web pages

- compress data for faster transmission
  - BUT: reduced redundancy

Speculation

- cache prefetching
  - memory, disk, network
- branch prediction
- speculative execution

BUT:

- wasted effort
- extra undo overhead?

Discard

- packet arrival discard
- packet drop from front
- transaction abort

BUT:

- wasted effort

Cost/Performance Trade-Off

- cache size
- priorities
- processing power
- overhead

Other Ideas

- change expectations? e.g. cell phone call setup
- deal with it... e.g. playout buffer
- at least measure it
  - slicing
  - time stamping: granularity, correlation