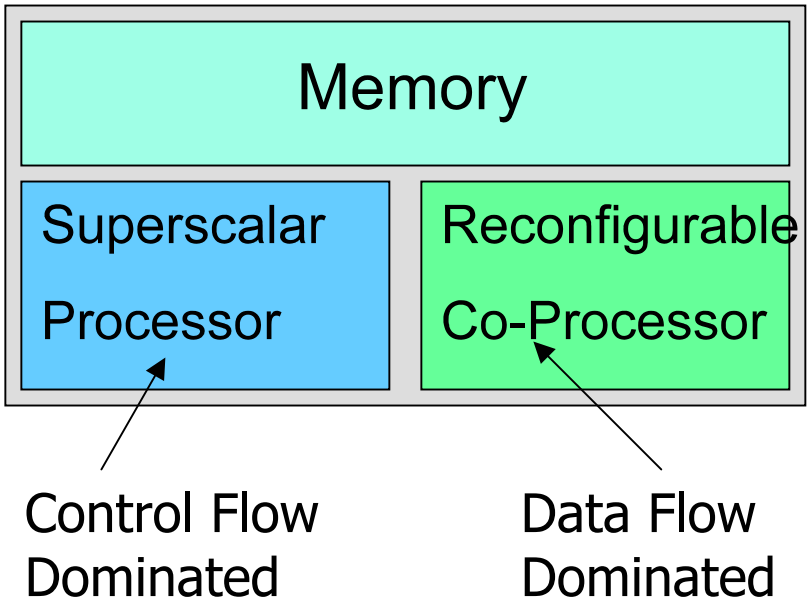
Configuration Caching and Swapping

Reconfigurable System

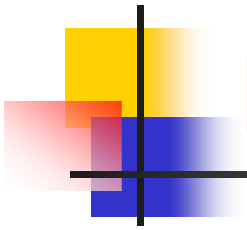
Performance Limited by

- Configuration Cost
- Configuration Swapping



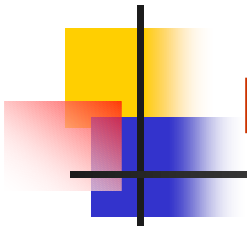
One Possible Solution

- Retain rfuOps on chip until required again
- Variable-size, variable-cost caching (NP-Hard)



FPGA Models

- ***Single Context:*** Configure WHOLE chip area
- ***Multi Context:*** Multiple complete configurations
- ***Partial Run-Time Reconfigurable (PRTR)***
  - Configure needed portion of chip
  - Variants:
    - *Relocation:* Dynamically load at any location, decide at run time
    - *Relocation+Defragmentation*: Compact rfuOps on chip



rfuOp Replacement

PRTR with Relocation

- Evict contiguous rfuOps
- Consider size, load latency and occurrence in past
- Use *History*: Predict future pattern of rfuOps
- Use *Penalty*: Replace *relatively* old rfuOps

Relocation+Defragmentation

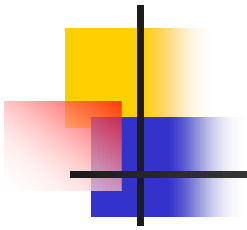
- Can select any rfuOps
- Same as VM if any part of rfuOp can be evicted

Three Level Hierarchy

Inclusive Model
(rfuOps in chip are *also* in cache)

Exclusive Model
(rfuOps in chip are *not* in cache)

Apply replacement algorithm in both chip and cache

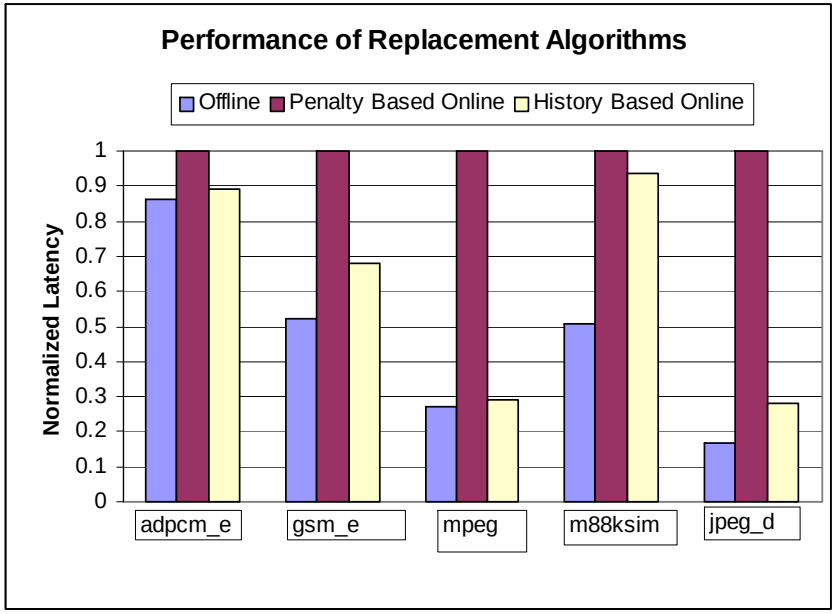


Configuration Swapping

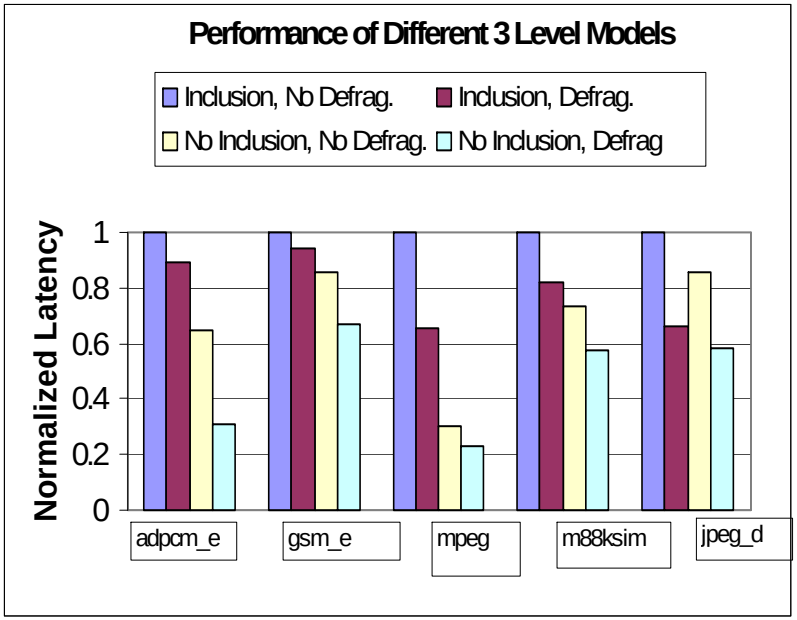
- ***Problem:*** Saving & Restoring complete fabric + cache state
- ***Motivation:*** Multitasking, Multithreading & other ILP technique on processor
- ***Tradeoffs:*** *# of processes, time/process, cache & fabric sizes*
- ***Requirements:*** *Speed, Low Overhead*

Performance

History based Algorithm performs better than LRU like penalty based algorithms

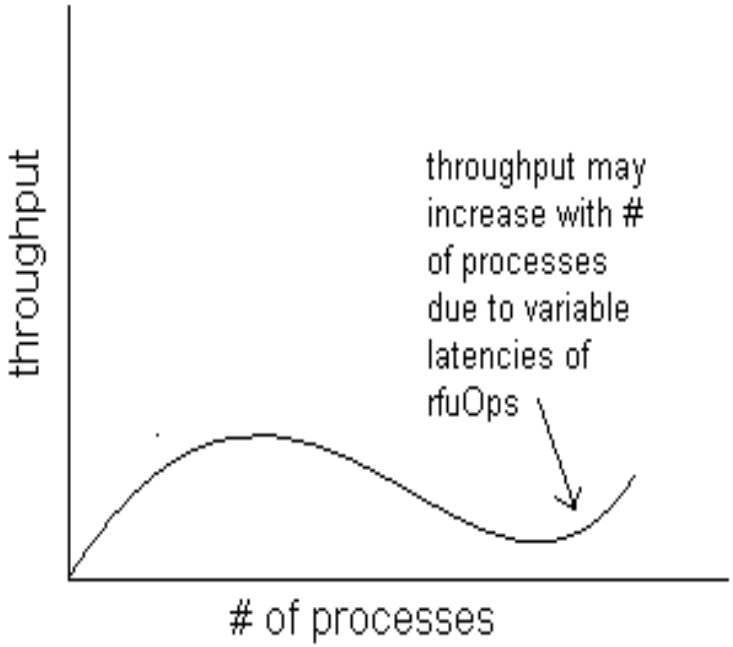


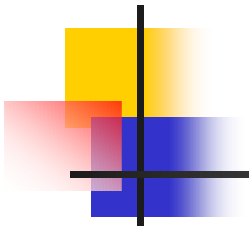
Three-level exclusion model performs best



Swapping Characteristics

- Performance decreases as time/processes decreases
- Increasing number of process may not reduce throughput.
- Variable rfuOp sizes and latencies affect throughput characteristics
- Size of cache and fabric affect save/restore overhead





Summary

- History-based online caching holds promise. Contemporary studies have focused on penalty-based techniques only
- 3-level exclusion model provides very good performance
- Defragmentation not always suitable in caching; thrashing can hamper performance
- Multitasking throughput not dependent on number of processes alone, due to variable-size, variable-cost nature of caching.