DFS and Load Balancing in Mixed-Critical Systems

http://www.dreams-project.eu/

TEI of Crete
www.teicrete.gr

Miltos D. Grammatikakis
George Kornaros
Kyprianos Papadimitriou
Ioannis Christoforakis
Antonis Papagrigoriou
Polydoros Petrakis

Mixed-critical Systems + Our approach

- Mixed-critical systems
  - Serve all workloads and don't sacrifice low-critical
  - Balance required performance and power/temperature
- We explore
  - Power- and thermal- aware computing at RTL and system-level
  - Dynamic frequency scaling (DFS) and thermal management in the presence of CPU load balancing

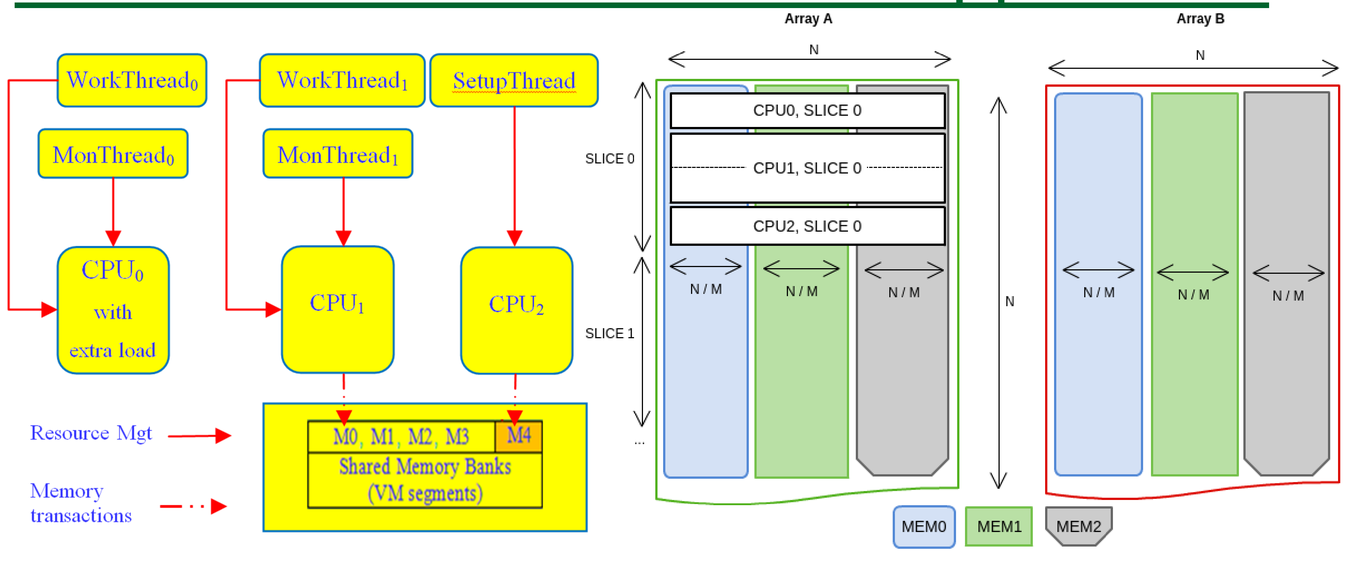
CPU Load Balancing (LB)

Processor CPU_i ($0 \le i \le x-1$)

```
Monitoring Thread;
start_Work_Thread(tid);
for(i=0; i<Slices; i++) {
  start_job_exec_timer();
  // wait workthreads
  // to complete
  // current slice
  barrier_wait(bar2);
  end_job_exec_timer();
  // workthread delay
  read_cpu_loads_from_mem();
  load_correction();
  set_next_loads();
  barrier_wait(bar3);
}
```

```
WorkThread;
for(i=0;i<Slices; i++) {
   read_next_load();
   matrix_multiply(Slice,
      start_index,next_load);
   // sync workthreads
   start_job_exec_timer();
   barrier_wait(bar1);
   end_job_exec_timer();
   write_cpu_load_to_mem();
   barrier_wait(bar2);
   // workthreads wait for
   // monitoring threads to read,
   // compute, write next loads
   // barrier_wait(bar3);
}
```

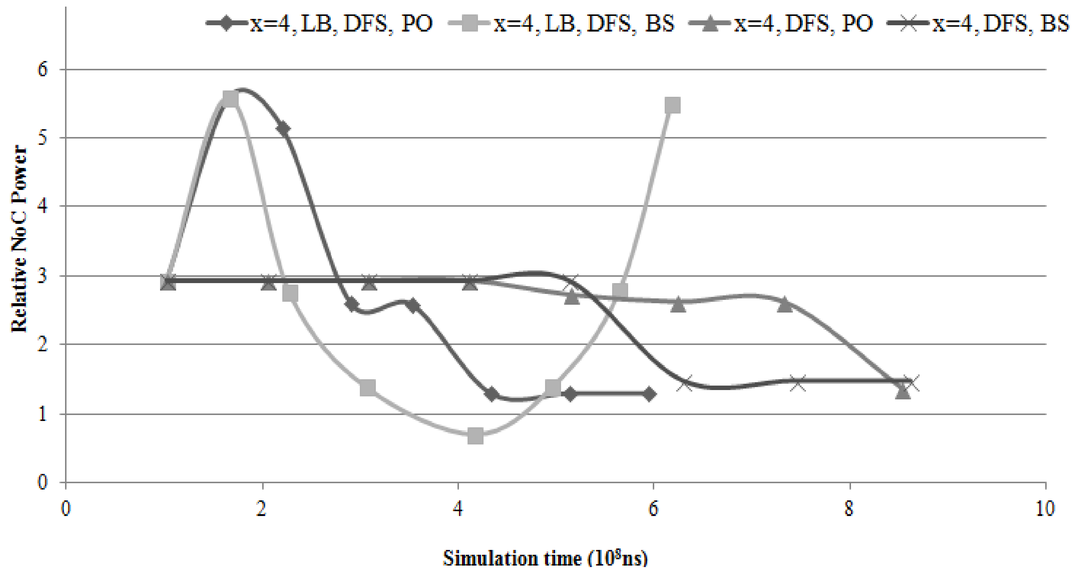
Architectural Framework & Application



- Working + monitoring threads
- 1 CPU/Memory for sync only
- Shared memory
- Connected via Hypercube NoC [1]
  - few hops; flexibility in paths
- CPU_0 extra load causes imbalance

- Matrix multiplication performed by 3 CPUs and 3 memories
- Array size N = 192 with 8 slices; size of slice determined from N/Slices

NoC Power vs. Execution Time

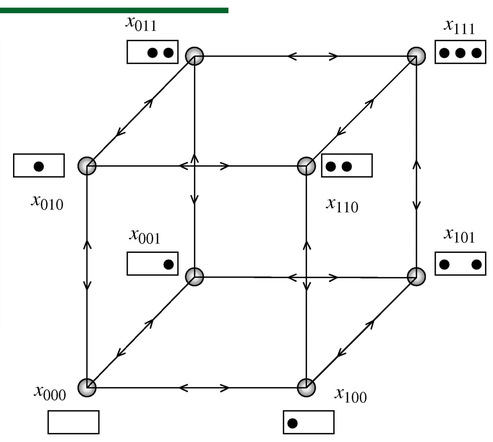


- Load balancing challenges DFS runtime decisions

1. *HSoC library for SystemC*, http://hsoc.sourceforge.net

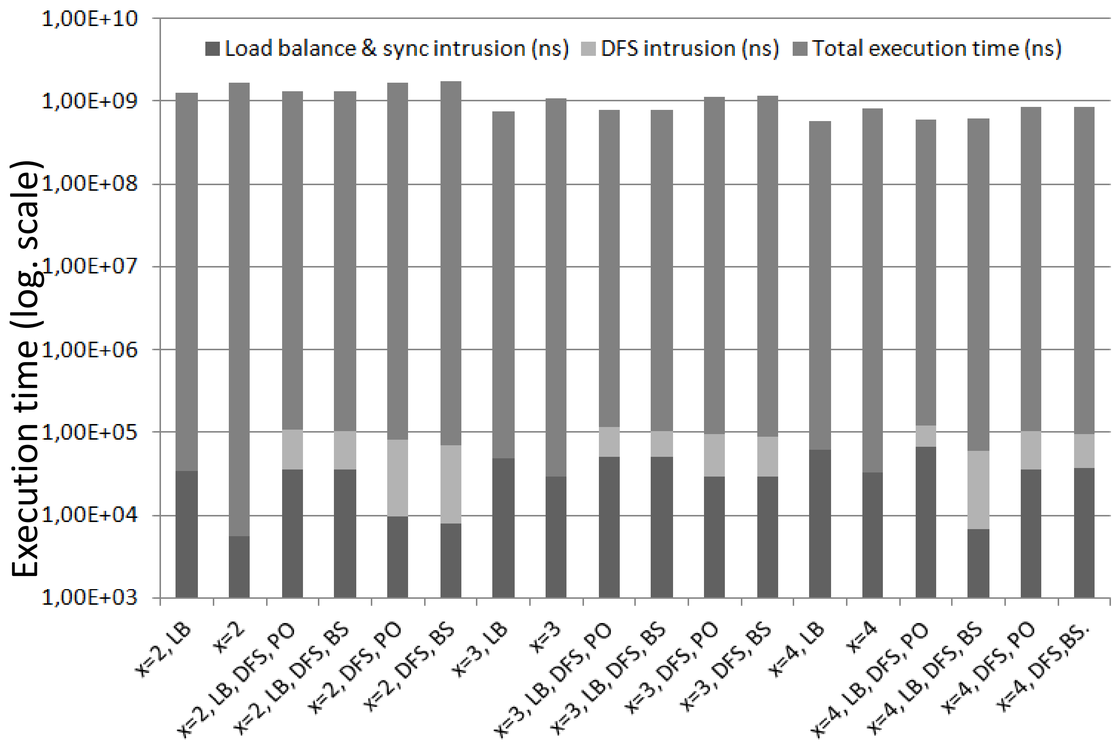
IPs onto a Binary Hypercube NoC

3-cube	0	1	2	3	4	5	6	7
X=2	M_4	CPU_0	CPU_1	CPU_2	M_0	M_1	M_2	M_3
X=3	M_3	M_0	M_1	M_2	CPU_3	CPU_0	CPU_1	CPU_2
X=4	M_3	M_0	M_1	CPU_4	CPU_0	CPU_1	CPU_2	CPU_3



- Array stored onto shared memory banks: M_0, M_1, ..., M_{6-x} in row major (x is CPU's ID)

LB & DFS vs. Execution time



- LB improves execution time by 22% - 32% ; small intrusion
- DFS with PO is more intrusive, but better than BS

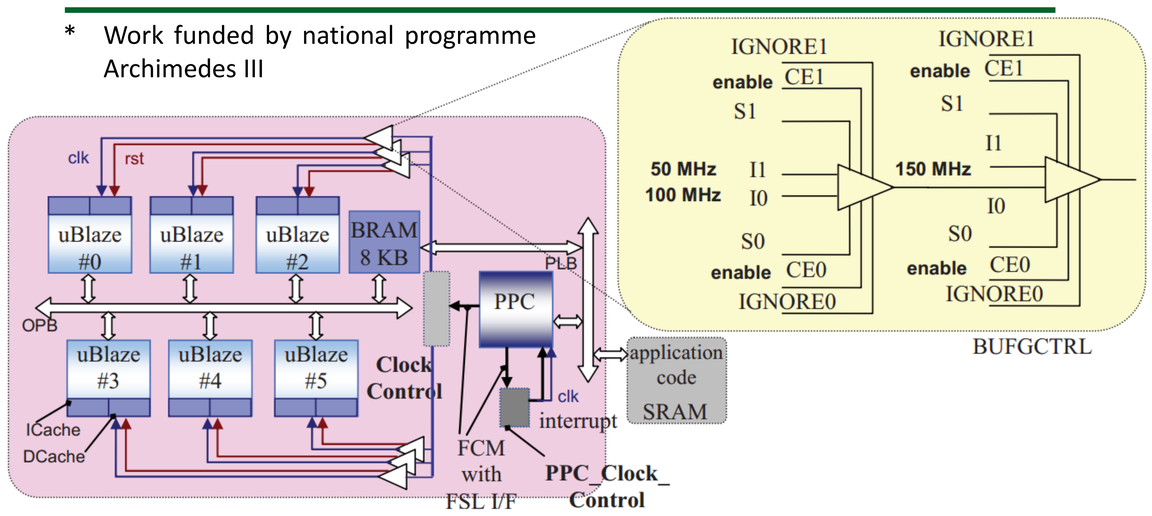
DFS Policies@NoC Router

- BS: Cumulative Buffer Size utilization within a time-window **+** comparing with sum of utilization of all router queues
- PO: Cumulative Packet Output rate from all queues of router within a time-window **+** comparing with the harmonic average packet rate for all routers
- DFS decisions = F (expected finish time, deadline)
- Thresholds to decide up/down frequency scaling

RANGE	EXPECTED FINISH TIME – DEADLINE	DFS Decision
A	> DPM_RT_HIGH	All Routers: Scale Up
B	in [DPM_RT_LOW, DPM_RT_HIGH]	BS or PO Policy
C	in [-DPM_RT_LOW, DPM_RT_LOW]	All Routers: No Scaling
D	in [-DPM_RT_HIGH, -DPM_RT_LOW]	BS or PO Policy
E	< -DPM_RT_HIGH	All Routers: Scale Down

DFS in a multi-CPU island node

* Work funded by national programme Archimedes III



- Adaptating a dithering algorithm [2], to reduce power hot-spots and smooth power spikes among the neighboring cores
- Each island performs weighted DFS adjustments of its local tasks

2. R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," Proc. of the Soc. of Inf. Display, vol. 17, pp. 75–77, 1976

System parameters

- Queue size of network router and memory controller = 8 packets
- Router clock period: T_{router} = 2, 4, 8, or 16ns
- CPU clock period: T_{cpu} = 4ns
- Memory controller clock period: T_{memory} = 8ns