# Impact of memory size on ECM and E$^2$CM

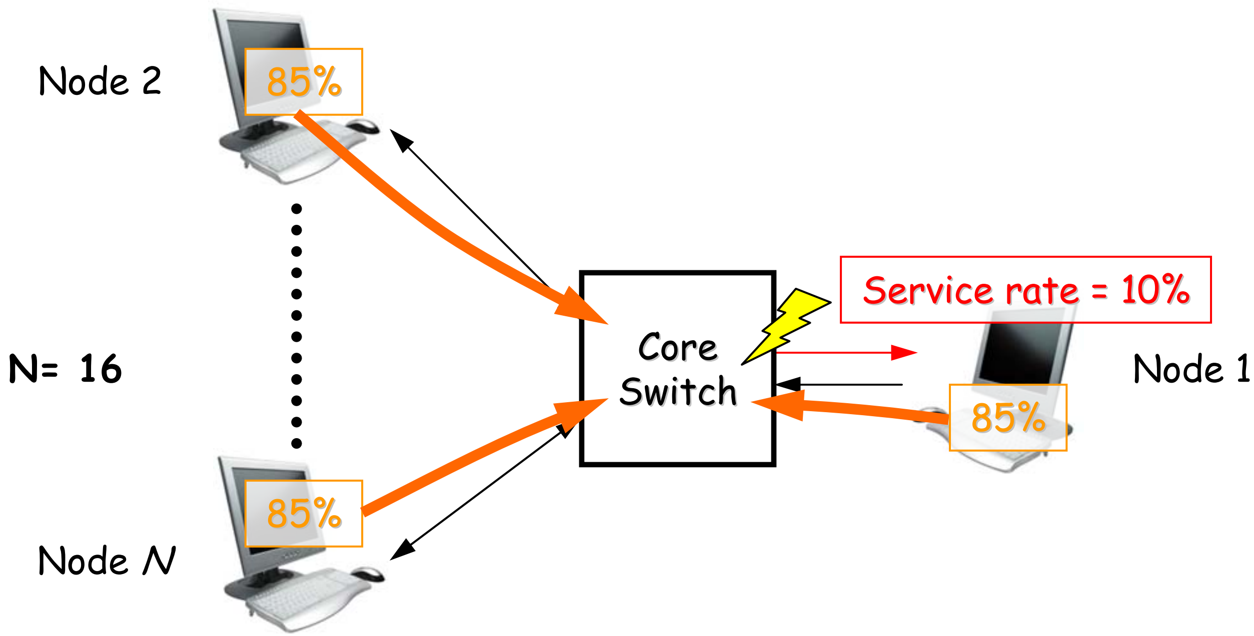
## Single-Hop High Degree Hotspot

Cyriel Minkenberg & Mitch Gusat
IBM Research GmbH, Zurich
May 10, 2007

# Targets

- Measure mean flow completion time, number of flows completed, number of frames dropped
  - Exponential and Pareto flow size distributions
    - Mean = 60 KB/flow (40 frames, 48 us), 75% load
    - Actual Pareto mean flow size = 17.2 KB, load = 57%
    - Traffic pattern read from trace file
  - PAUSE on/off
  - BCN(0,0) on/off

# Output-Generated Single-Hop High HSD

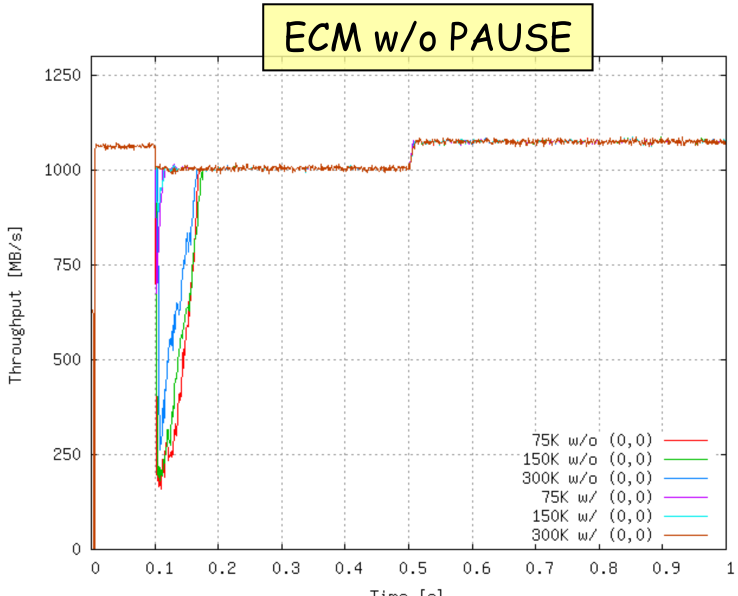


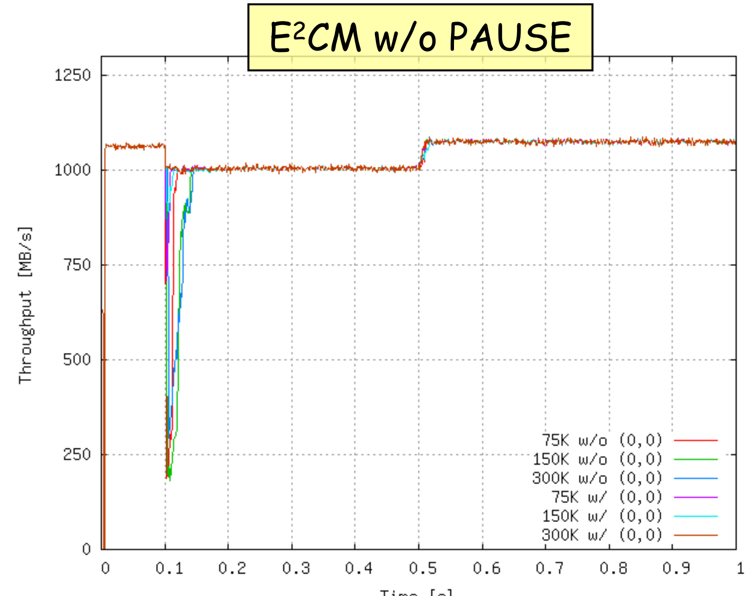

- All nodes: Uniform destination distribution, load = 85% (8.5 Gb/s)
- Node 1 service rate = 10%

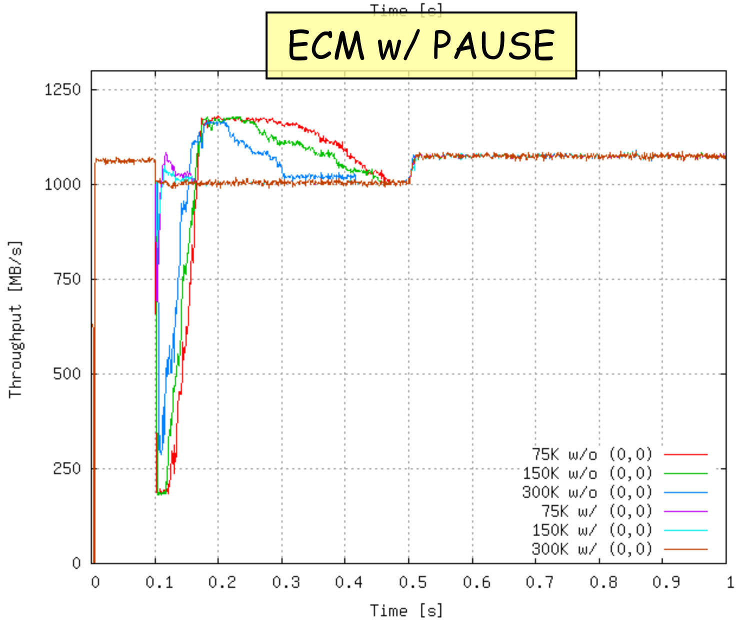
# Simulation Setup & Parameters (same as before)

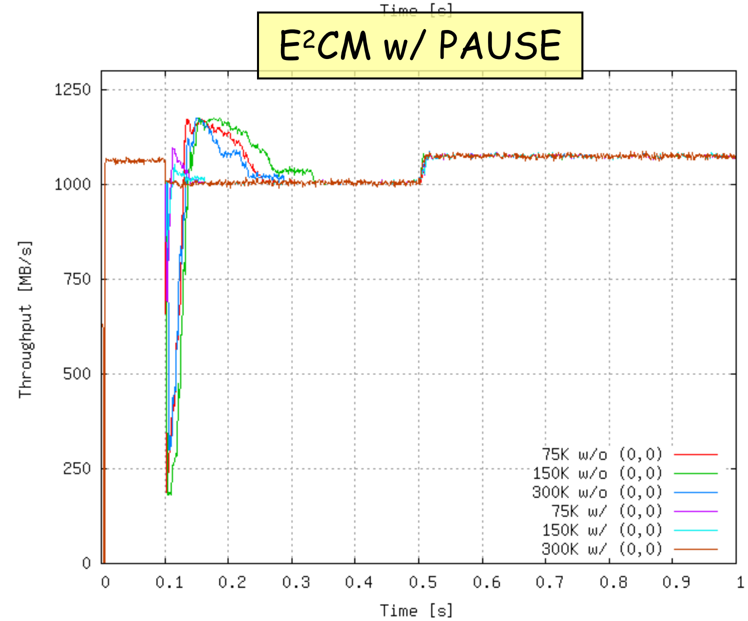
- Traffic
  - Bernoulli
  - Uniform destination distribution (to all nodes except self)
  - Fixed frame size = 1500 B
- Scenario
  1. Single-hop output-generated hotspot
- Switch
  - Radix N = 16
  - **M = [75, 150, 300] KB/port**
  - Link time of flight = 1 us
  - Partitioned memory per input, shared among all outputs
  - No limit on per-output memory usage
  - PAUSE enabled or disabled
    - Applied on a per input basis based on local high/low watermarks
    - $watermark_{high}$ = M – rtt*bw KB
    - $watermark_{low}$ = M – rtt*bw KB
    - If disabled, frames dropped when input partition full
- Adapter
  - Per-node virtual output queuing, round-robin scheduling
  - No limit on number of rate limiters
  - Ingress buffer size = infinite, round-robin VOQ service
  - Egress buffer size = 150 KB
  - PAUSE enabled
    - $watermark_{high}$ = 150 – rtt*bw KB
    - $watermark_{low}$ = $watermark_{high}$ - 10 KB
- ECM
  - W = 2.0
  - $Q_{eq} = M/4$
  - $G_d = 0.5 / ((2*W+1)*Q_{eq})$
  - $G_{i0} = (R_{link} / R_{unit}) * ((2*W+1)*Q_{eq})$
  - $G_i = 0.1 * G_{i0}$
  - $P_{sample}$ = 2% (on average 1 sample every 75 KB
  - $R_{unit} = R_{min}$ = 1 Mb/s
  - BCN_MAX enabled, threshold = M KB
  - BCN(0,0) dis/enabled, threshold = 4*M KB
  - **Drift enabled**
- $E^2CM$ (per-flow)
  - W = 2.0
  - $Q_{eq,flow}$ = M/20 KB
  - $G_{d,\ flow} = 0.5 / ((2*W+1)*Q_{eq,flow})$
  - $G_{i,\ flow} = 0.005 * (R_{link} / R_{unit}) / ((2*W+1)*Q_{eq,flow})$
  - $P_{sample}$ = 2% (on average 1 sample every 75 KB)
  - $R_{unit} = R_{min}$ = 1 Mb/s
  - BCN_MAX enabled, threshold = M/5 KB
  - BCN(0,0) dis/enabled, threshold = 4*M/5 KB

# Aggregate throughput



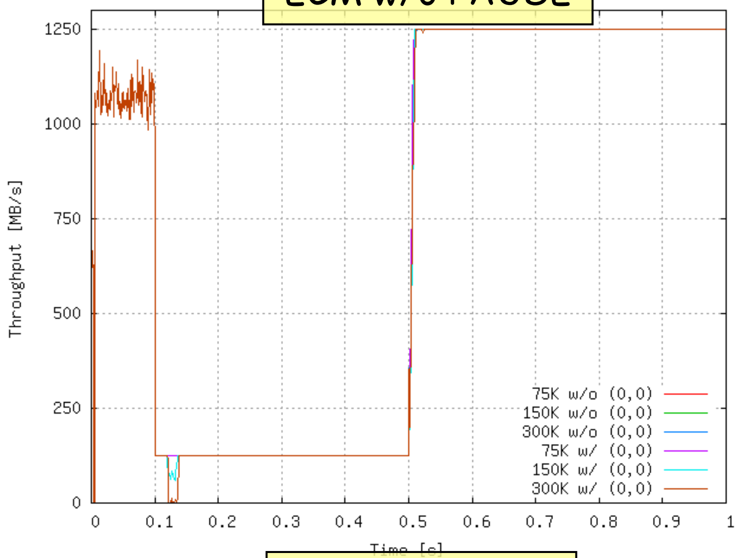



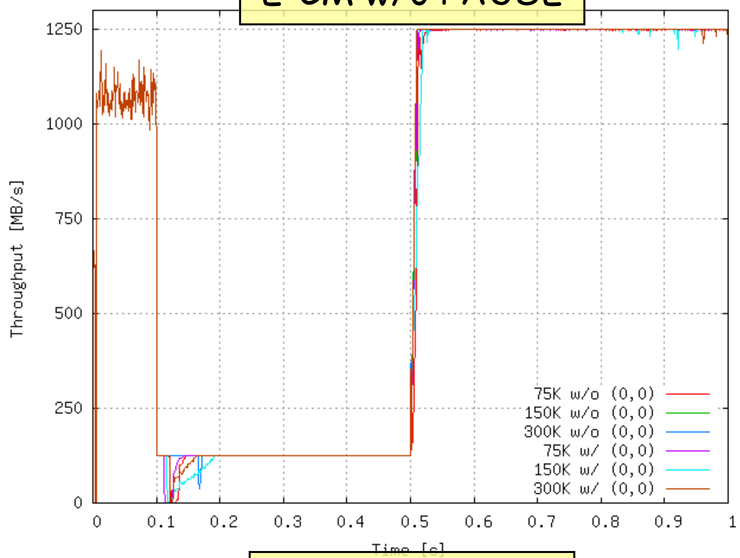



# Hot port throughput

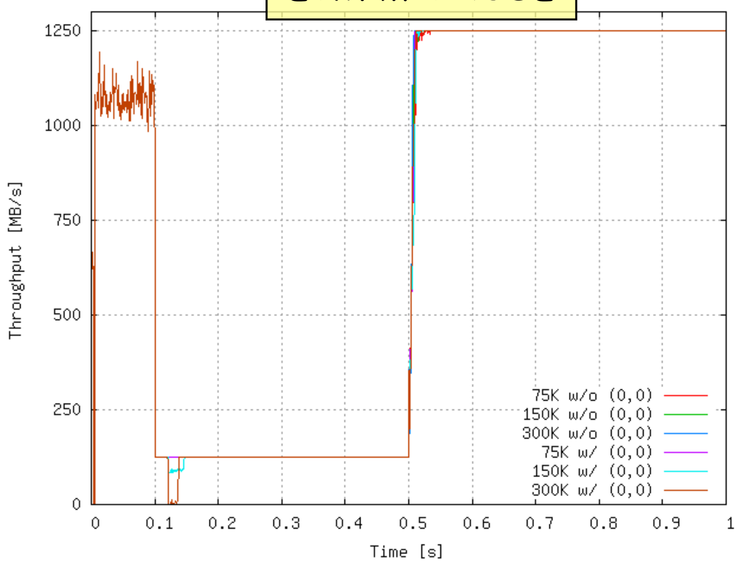
ECM w/o PAUSE



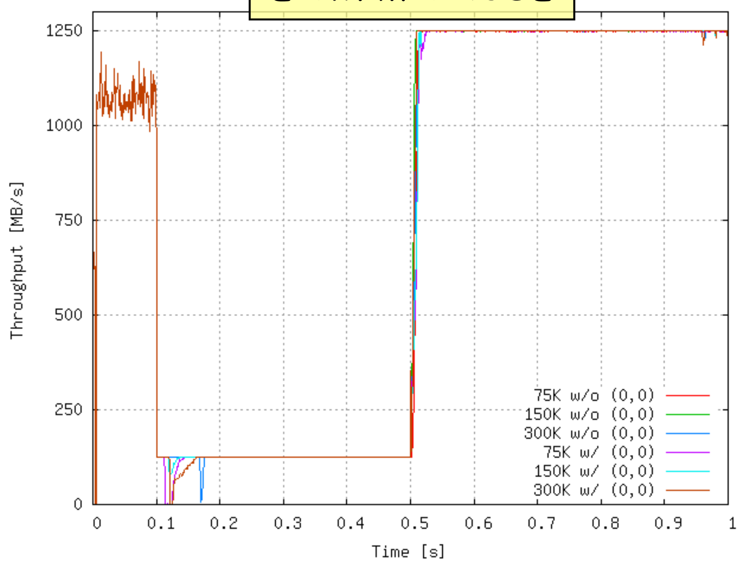

E²CM w/o PAUSE




ECM w/ PAUSE

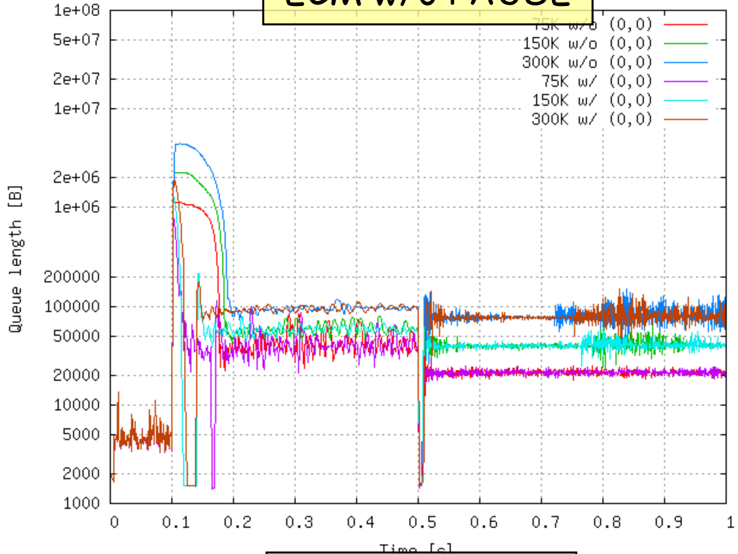



E²CM w/ PAUSE

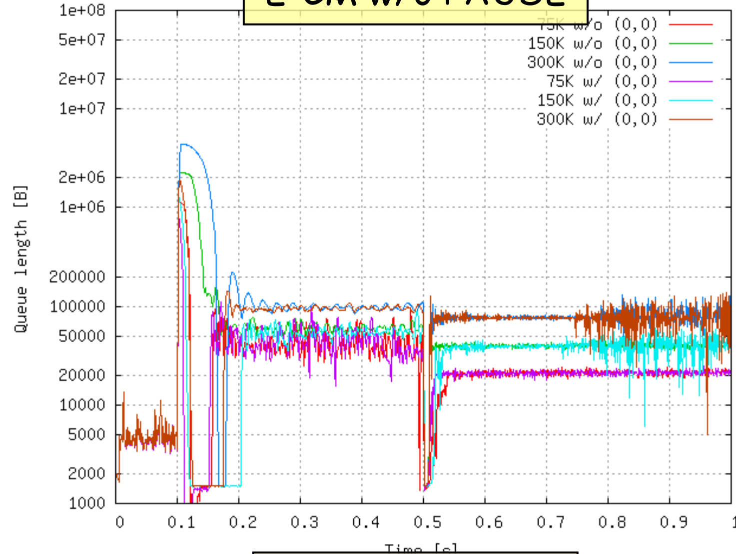


# Hot port queue length

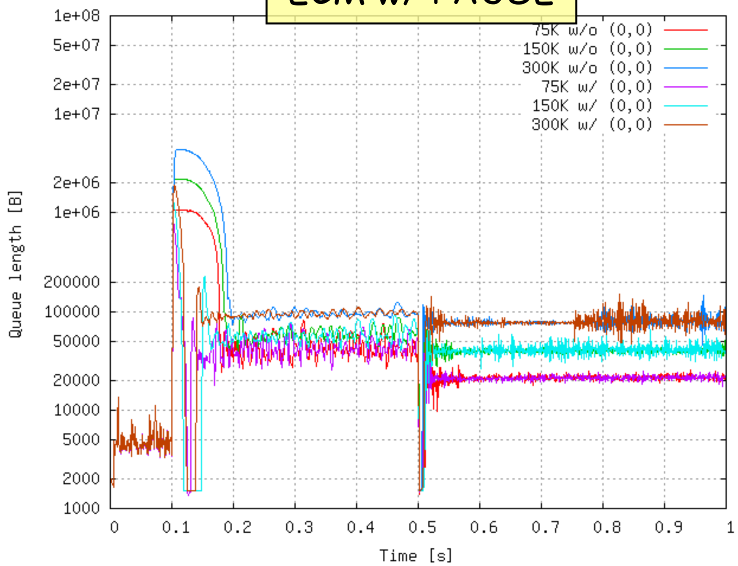
ECM w/o PAUSE



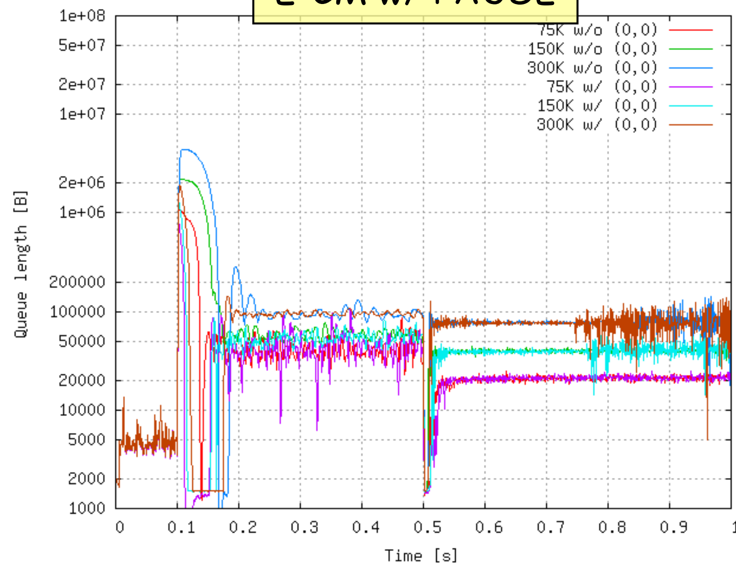

E²CM w/o PAUSE



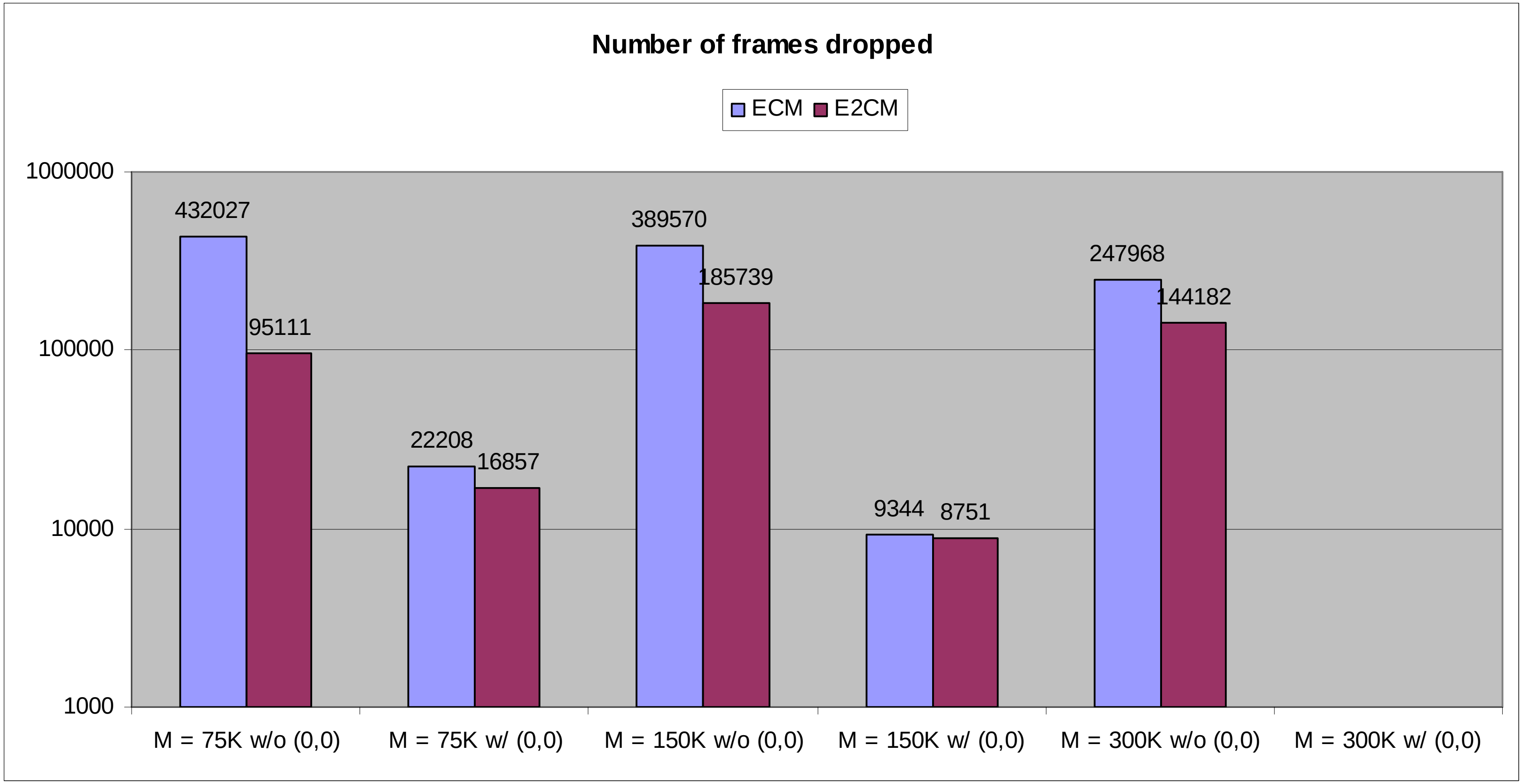

ECM w/ PAUSE




E²CM w/ PAUSE

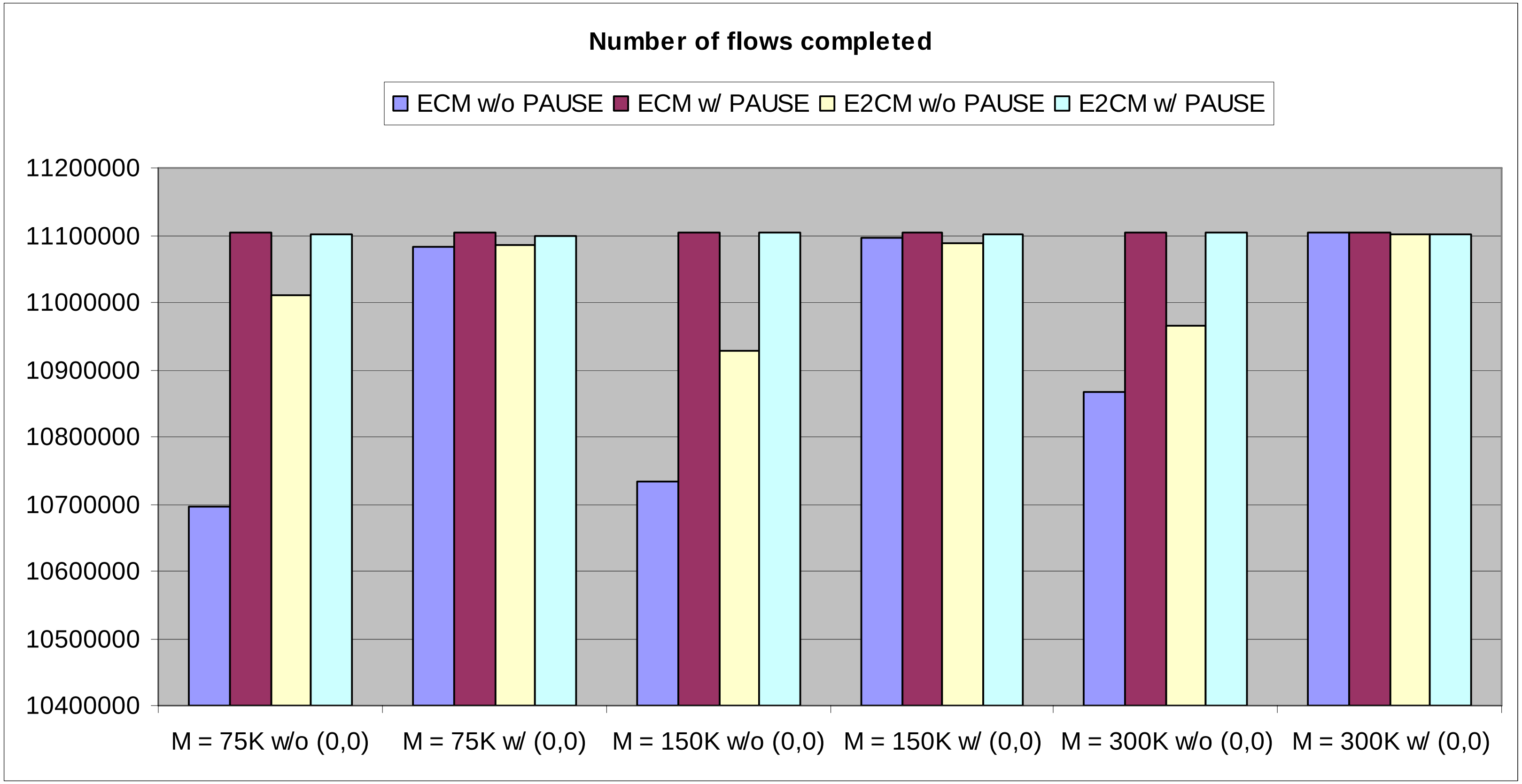


# Number of frames dropped (no PAUSE)



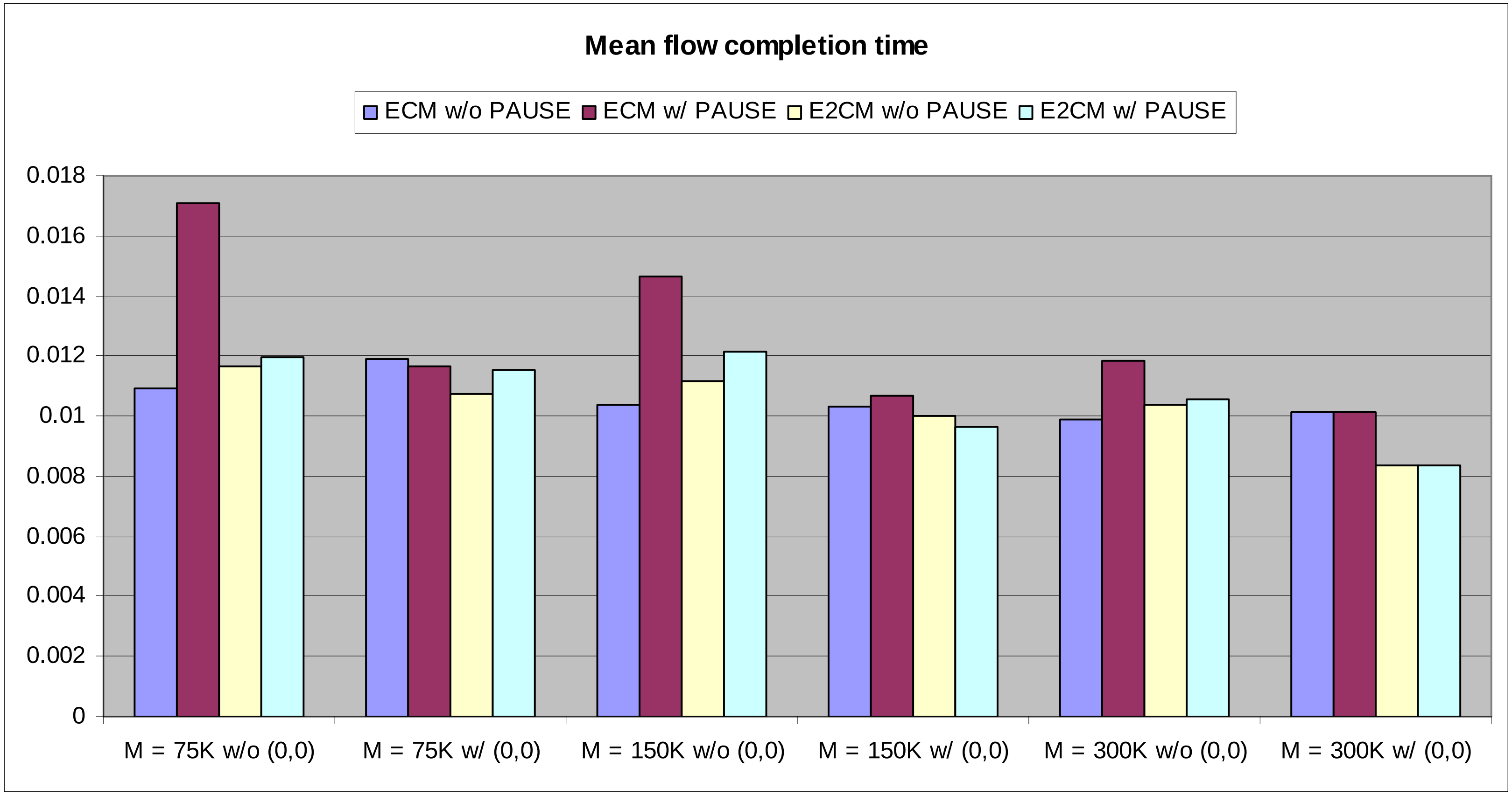

- $E^2CM$ drops fewer frames

# Number of flows completed




- When either PAUSE or BCN(0,0) are enabled numbers are virtually identical
- Without PAUSE and BCN(0,) $E^2CM$ tends to do somewhat better

# Mean flow completion time




- Larger memory ➔ shorter flow completion time
- ECM with PAUSE tends to perform worst
- With largest memory, $E^2CM$ has about 20% lower FCT than ECM

# Conclusions

- Chairman has raised the issue of more realistic (shallow) onchip buffers
  - Will our CM schemes still work - and how well?
- Findings: Baseline ECM and E2CM show robust performance even w/ reduced memory
  - Resilience: both loops have sufficient stability phase margin built-in
- Performance is comparable, E2CM sometimes better