



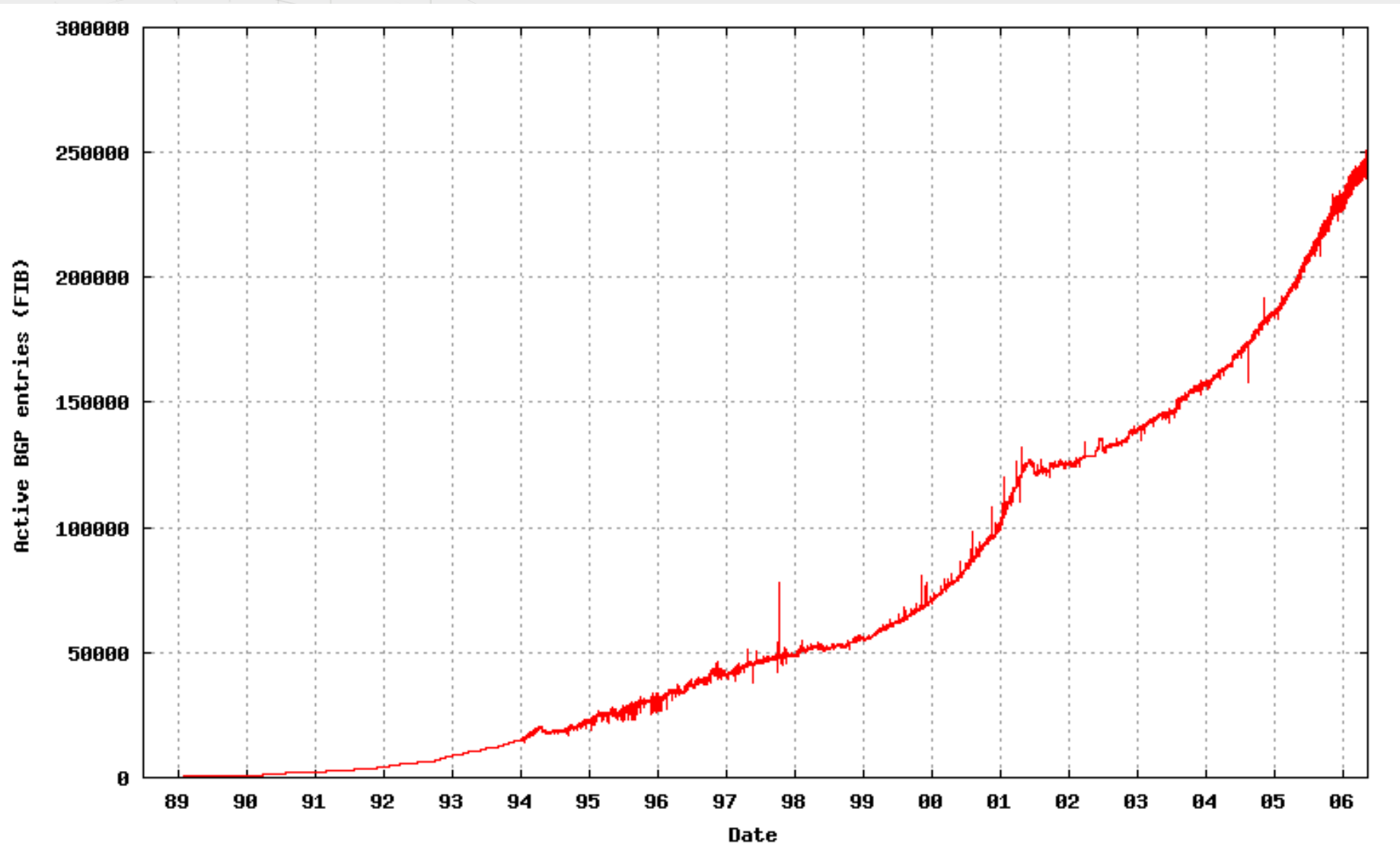
Maximizing Parallelism in NP

Ran Giladi
EZchip Technologies Ltd.,
Communication Systems Engineering
Ben-Gurion University, Israel

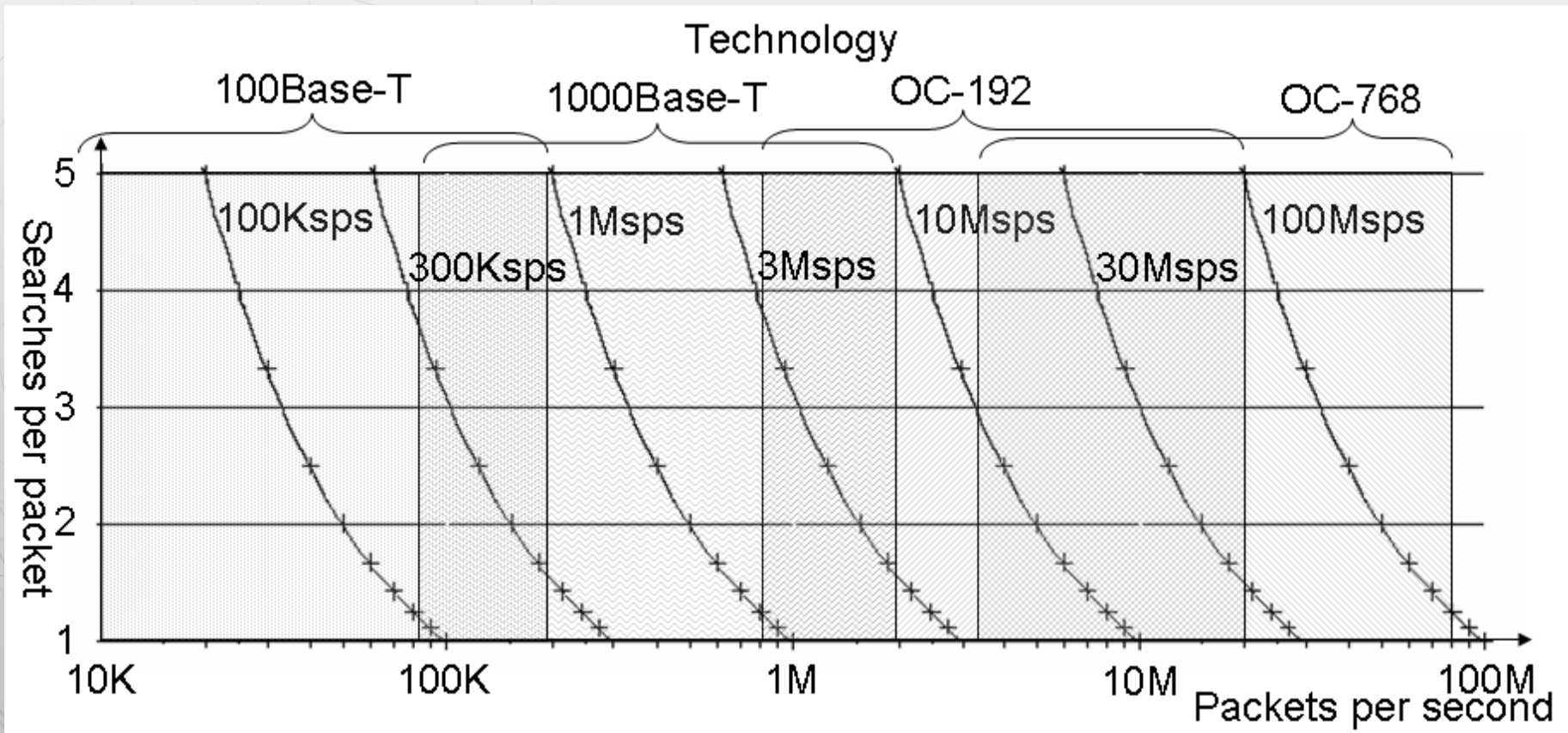
Outline

- Why Network Processors?
- What are Network Processors?
- How networking environment affect performance and parallelism of NP?
- What NP architectures are there to support parallelism (and performance)?
- What is good for what?
- Questions?

Network growth: Macro view



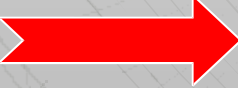


Search requirements: Micro view



Or, ~10nSec for a complete packet processing!

Networking requirements

- Exponential growth – higher than processing growth
 - Classifying, lookups, (modifications), forwarding
- 
- Multi processors and parallelism is a must
- 
- Von Neumann (MSISD, MIMD, SIMD, MISD) / Dataflow
 - NoC provide switching and not processing
 -  Traffic Managers, Switch Fabrics, Packet processors, GP multi core, co-processors, search-engines, application processors,...

NP Typology

- Low entry (<1Gbps) – Access
 - Wintegra, Agere, PMC sierra, Intel (IXP300)
- Medium level (2-5Gbps) – Legacy and multi service, service cards, data center and L7 applications
 - Legacy: AMCC, Intel, C-port, Agere, Vittese, IBM (Hifn)
 - Multi core: Cavium, RMI, Broadcom, Silverback, Chelsio
- High end (10-40Gbps) – Metro, line cards
 - EZchip, Xelerated, Sanburst, Bay micro

Performance

- Black box (system wise): Throughput (MPPS), latency, implementation issues (power, cost)
- White box (chip level):
 - Internal aggregated bandwidth (Gbps)
 - Internal memory, inter-processors
 - External aggregated bandwidth (Gbps)
 - mainly memories for frames, rules DB, look-aside data & statistics
 - Operations (& searches) per second
 - How much “real work” rate is achieved?
 - Or, how suffice is a wire from ingress to egress...

Parallelism: Enablers

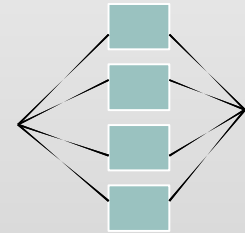
- Multiprocessors - the usual suspects
- **Hardware assists – essential in packet processing**
 - **Traffic managers**
 - **Search and classification engines**
- Data paths – critical for lots of parallel, asynchronous processing
- **Schedulers – resource optimization, and usage transparency**

Parallelism: NP

- Search engines & traffic managers - again
- Packets independence
 - Line card apps vs. L7 and stateful processing
 - Implications: buffer usage, processing stall, functionality
- Instruction set for packet processing
 - Heterogeneous vs. homogeneous processors
 - General purpose vs. optimized processors
 - “work” per instruction
 - Flexibility – changing networking tasks & protocols
- Tight real time constraints
- Parallelism transparency and ease of use – programming model

NP architectures

- Homogenous parallelized processors (μ /pico engines)
 - IBM (Hifn), Intel, Multi-cores (mainly MIPS), C-Port



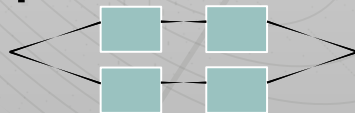
- Pipeline of homogenous processors
 - Xelerated, C-Port, Intel



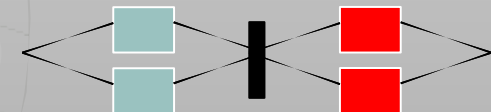
- Pipeline of heterogonous processors (Agere)



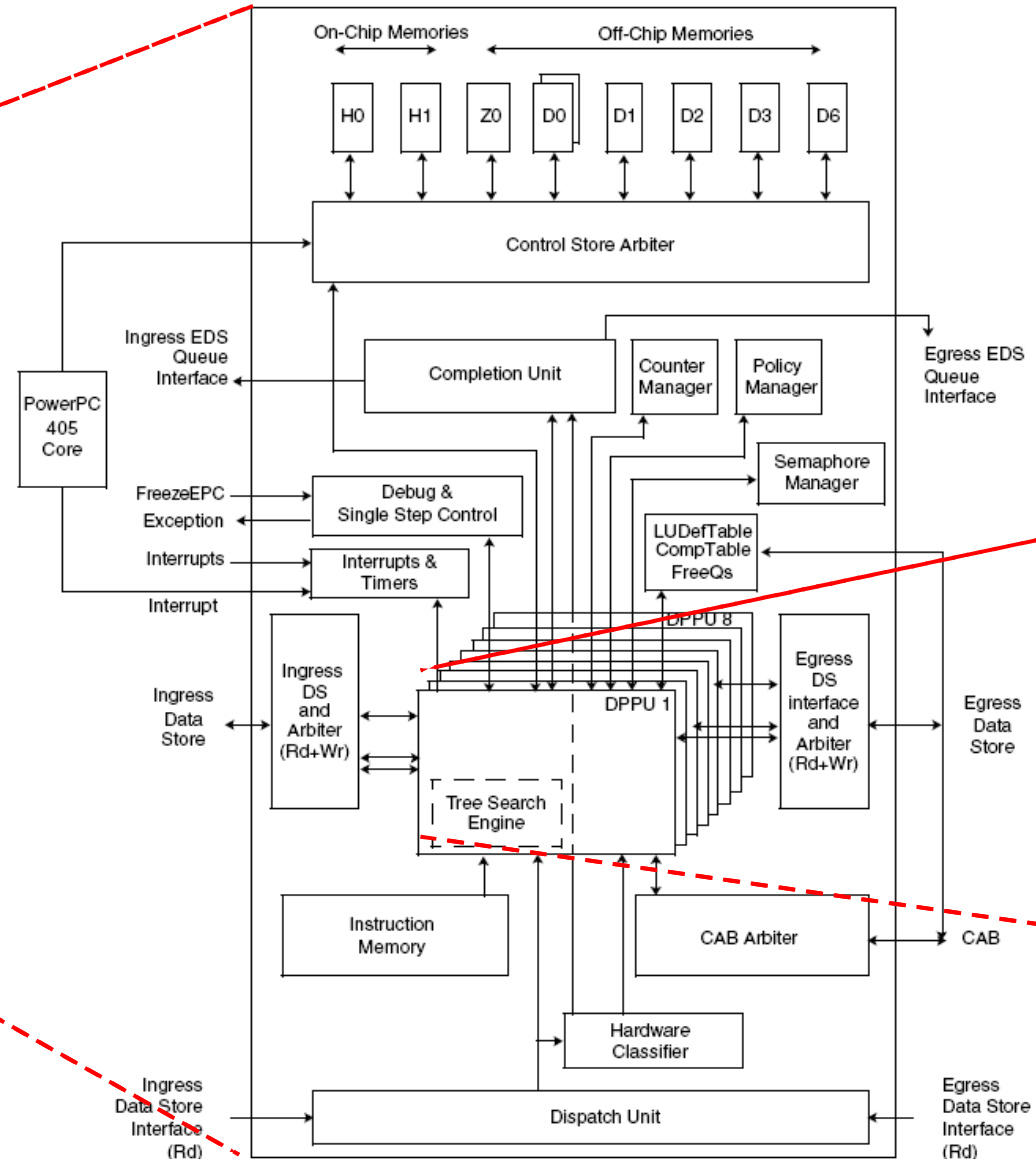
- Parallel pipelines of homogenous processors (Cisco)



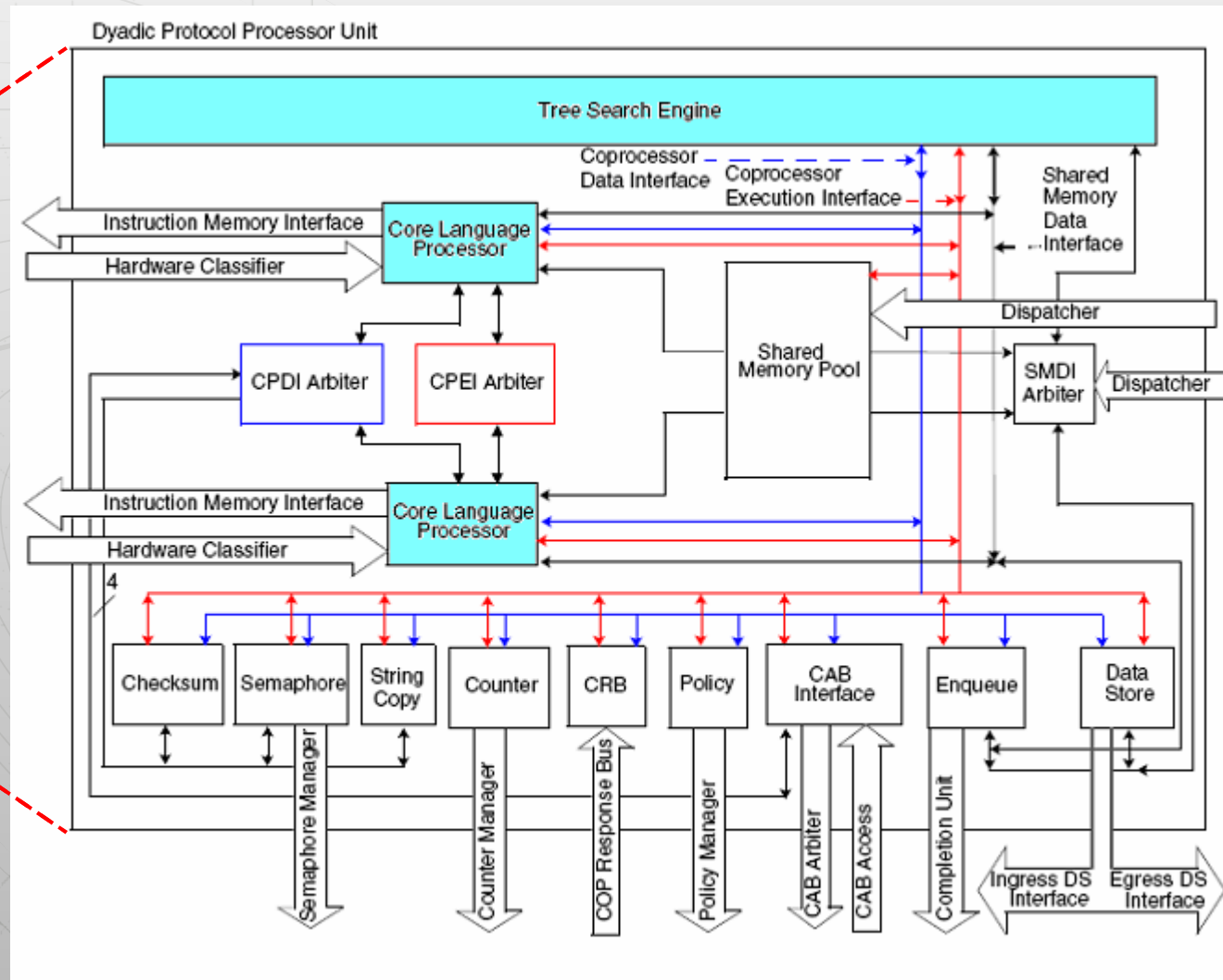
- Pipeline of parallel processors (EZchip)



IBM NP Architecture

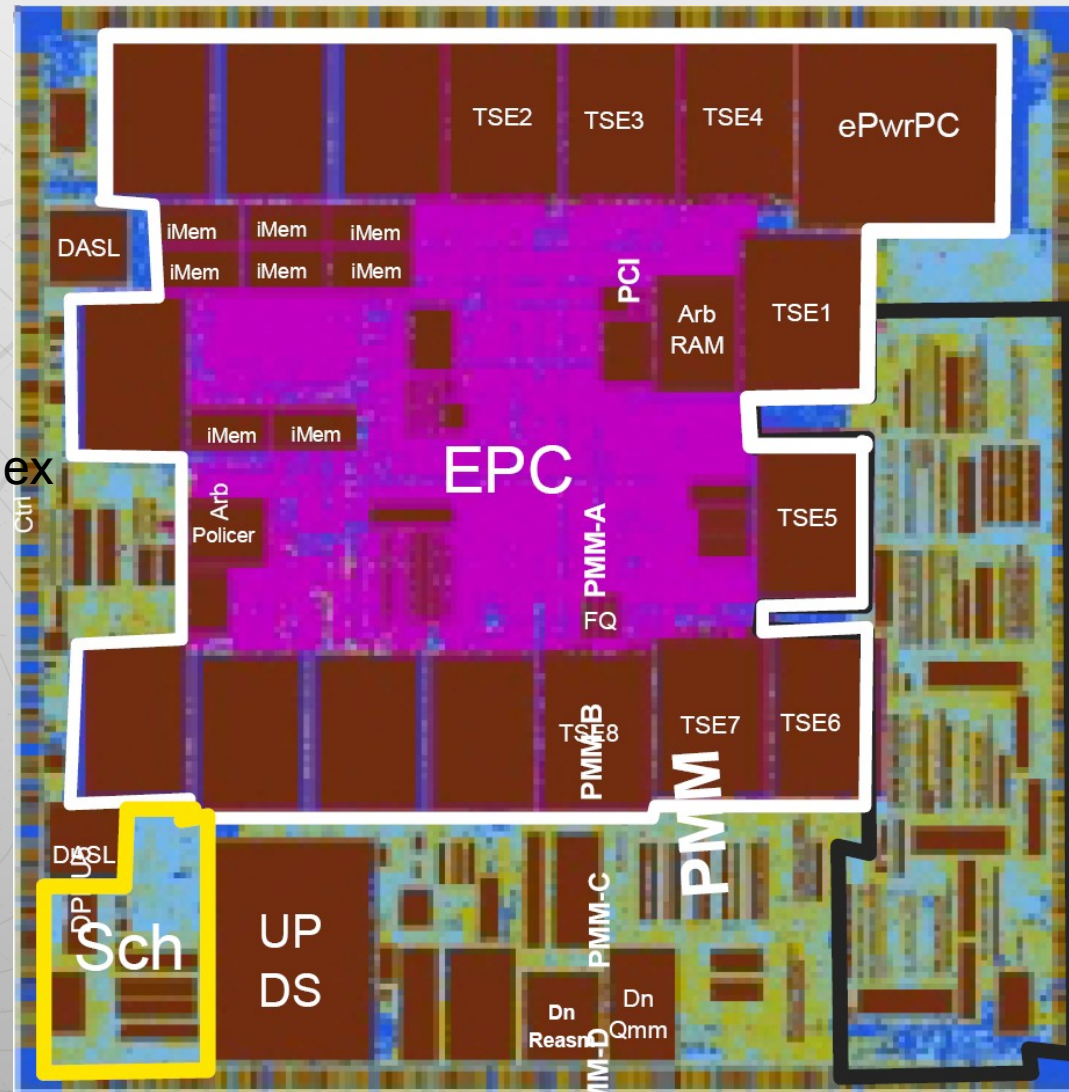


Zooming in – IBM NP



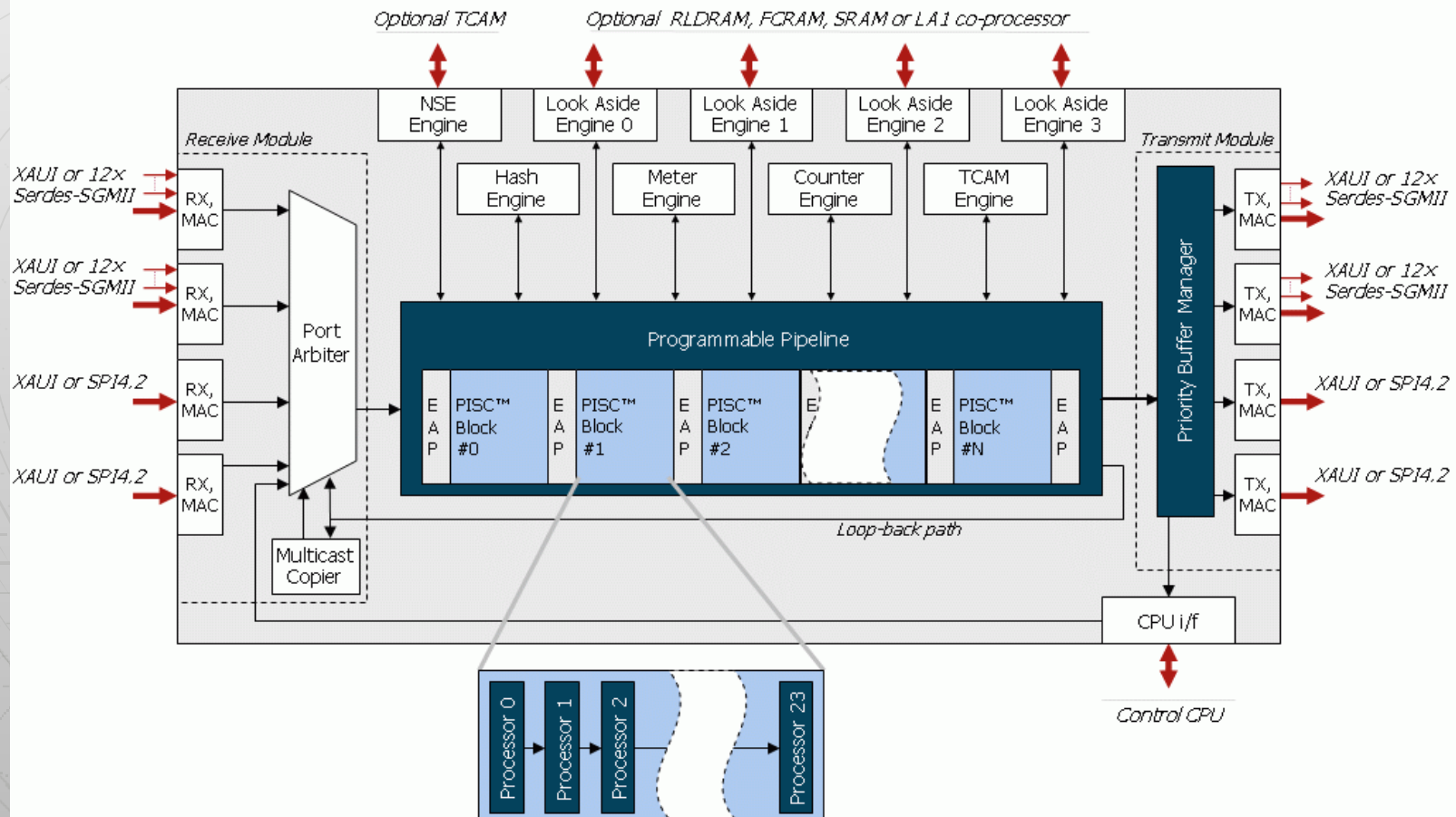
IBM NP - MPSoC

EPC – Embedded Processor Complex
ePwrPC – embedded PowerPC Complex
TSE – Tree Search Engine
PMM – Physical MAC Multiplexer
DASL – Data aligned synchronous link

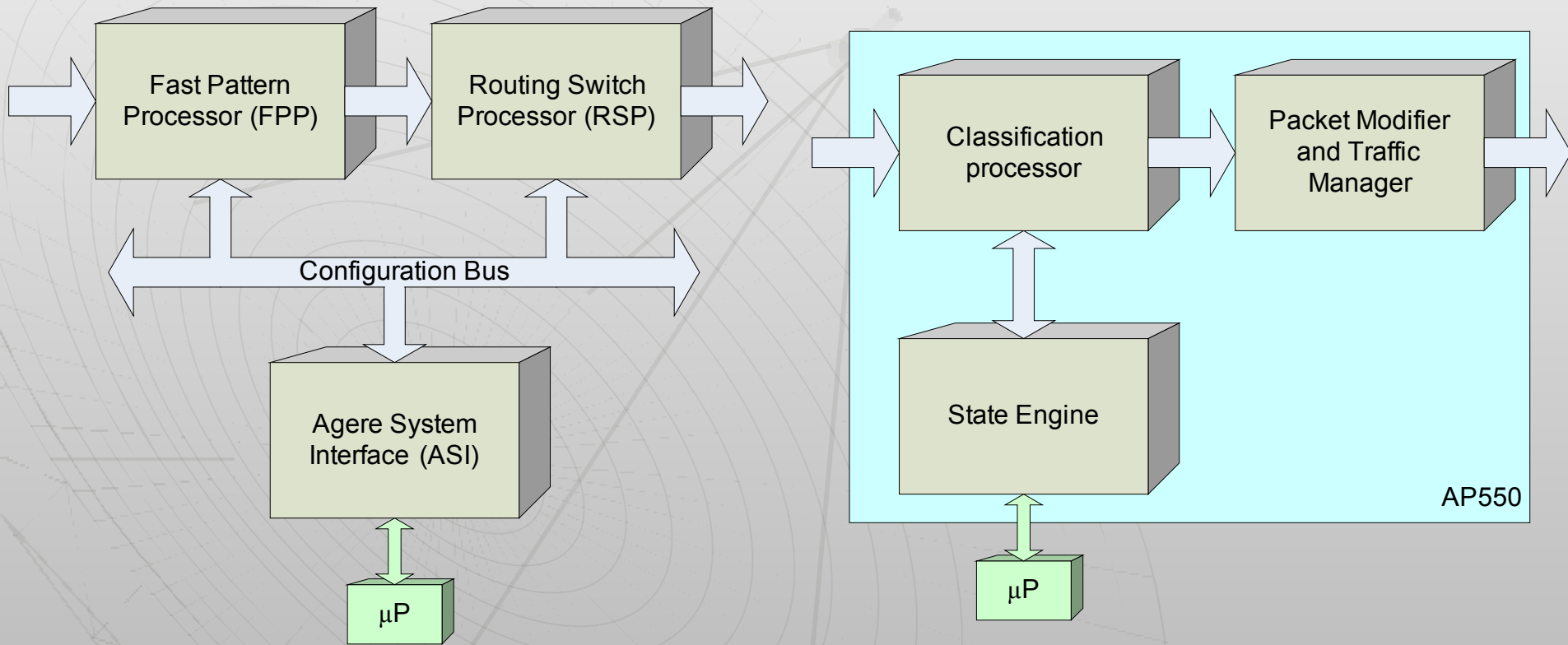


Pipeline of homogenous processors

Xelerated architecture



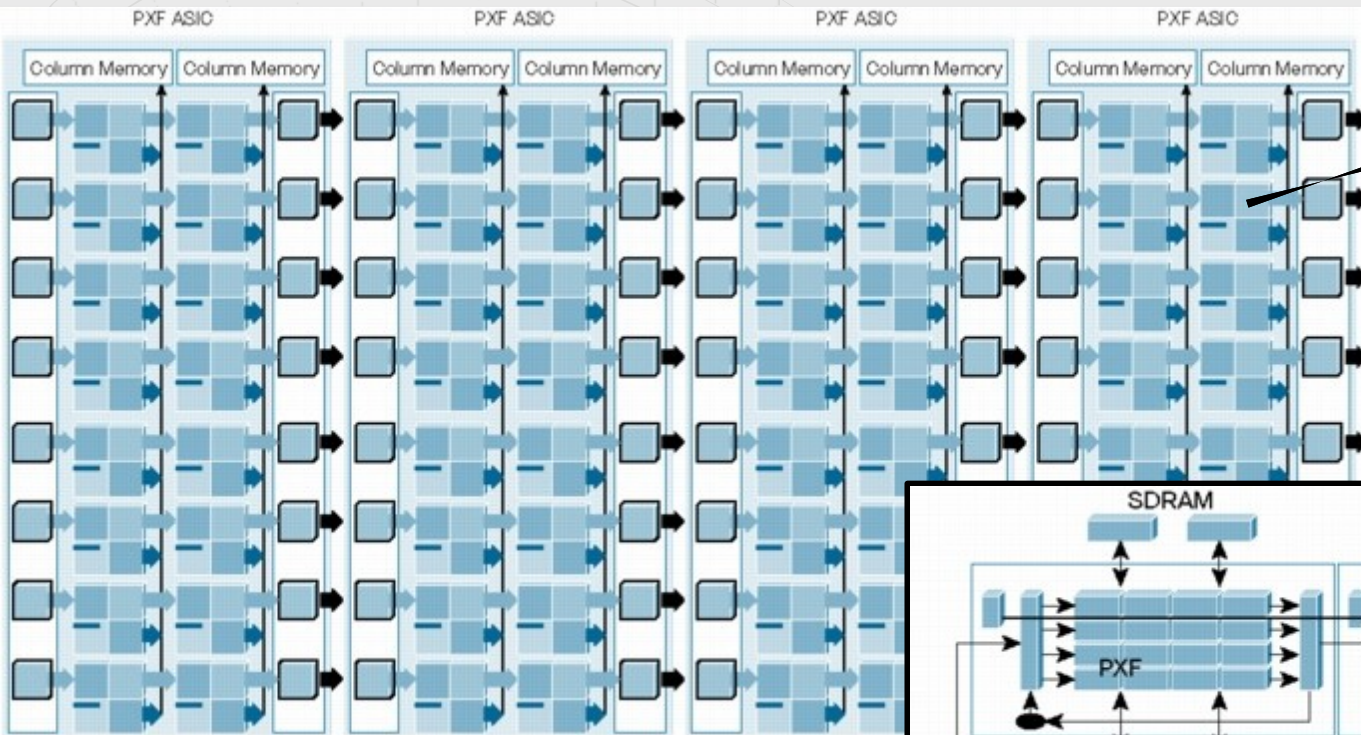
Pipeline of heterogonous processors



Agere's PayloadPlus architecture

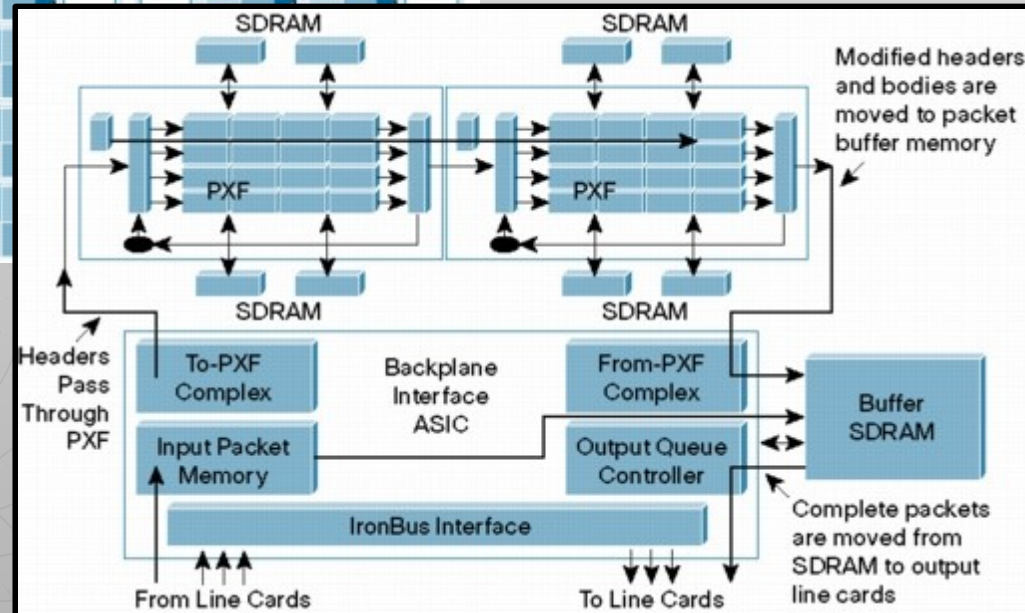
Parallel pipelines

Cisco's Toaster Architecture



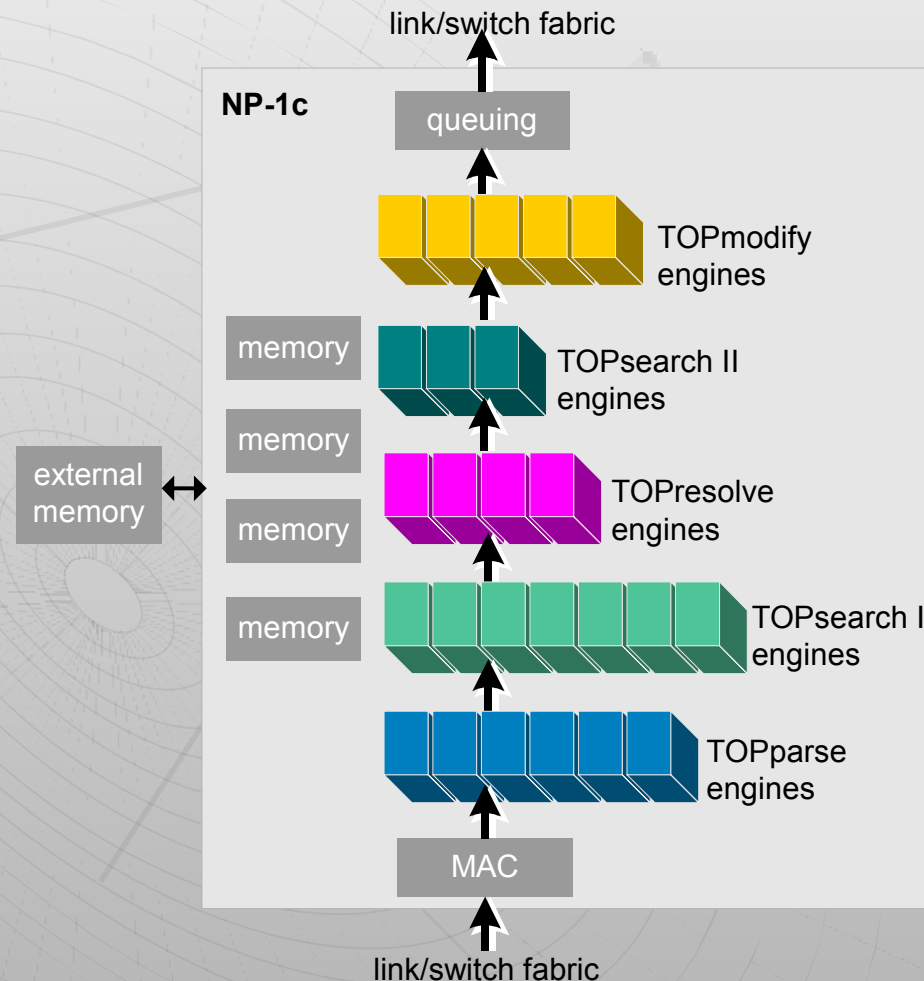
*Express Micro Controller (XMC)
(dual instruction issue execution unit)*

*Parallel eXpress Forwarding (PXF)
(8 parallel pipelines of 16 XMC)*



Pipelines of parallel processors

EZchip Architecture



Pipeline or Parallel?

- Programming Simplicity

Parallel	Pipeline
Packet ordering	Pipeline balancing (HW/SW)
Shared resources and semaphores	Slowest stage determines throughput
	Control and data hazards

- Processing flexibility

- Critical to NPs for adopting new protocols, feature sets, or applications
- Determined by large code space, scalable
- Excellent in parallel, restricted in pure pipeline
- Deterministic and run-to-completion processing

Heterogeneous or Homogenous?

- Programming simplicity
 - Similar instruction set
 - Flexibility in function locations
- Silicon efficiency
 - Unique capabilities are either restricted or wasted
 - Unique hardware accelerators are still a must

Simulations

- What to measure?
- Simulation models (OMNeT++) of pure parallel (Intel), pure pipeline (Xelerated), and pipeline of parallel processors (Ezchip) were used to evaluate processors utilization
- Results are highly dependent on networking application

What is good for what

- Multicore:
 - Greater flexibility
 - Applications, multi services, L7 tasks
 - => service cards
- Deterministic Pipeline:
 - Classifying, forwarding
 - => line cards
- Run to completion Pipeline:
 - Deep packet inspection, classifying, L4-L7 apps
 - => line cards, service cards

Conclusions

- Parallelism enables maximal utilization of silicon for providing power (work rate = packets per second multiplied by processing per packet)
- Heterogeneous, task optimized processors, with similar instruction set, organized in a pipeline of parallel units, enables maximum parallelism while preserving simple programming model.

References (& origin of title)

- Shah, N. & Keutzer, K. “**Network Processors: Origin of Species**”, *Proc. of ISCIS XVII*, 2002.
- Tullsen, D.M., Eggers, S.J., & Levy, H.M., “**Simultaneous Multithreading: Maximizing On-Chip Parallelism**”, *Proceedings of ISCA XXII*, 1995.

