

Network Processors

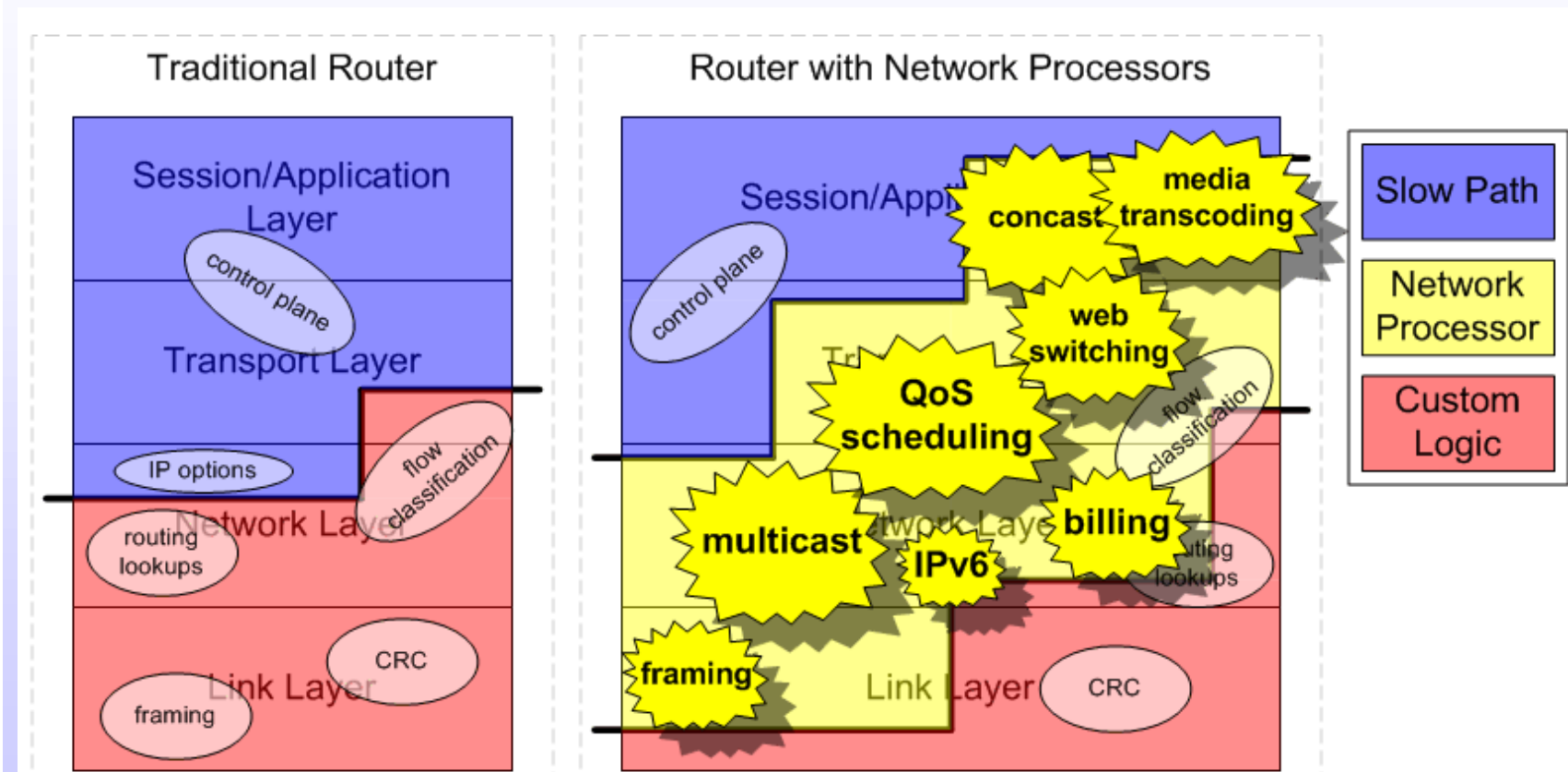
Flexibility and Performance for Next-Generation Networks

Tilman Wolf

Department of Computer Science, Washington University, St. Louis, MO
wolf@ccrc.wustl.edu

Motivation

- * New protocols and new services in network:



- * Processor in data path:

- ◆ New applications (header and payload processing)
- ◆ Rapid deployment (software changes not hardware)

Problem

"Design a network processor for Gigabit link rates."

- * Challenges:

- ◆ Performance (inherent lower performance of software)
- ◆ System complexity (maintain code and processing state)
- ◆ Technology limits (size of chip)
- ◆ Scheduling (many short tasks)

- * Why is this interesting / novel ?

- ◆ Driven by I/O-centric applications
- ◆ Workload, OS, data-flow different from workstations
- ◆ Enables new applications inside the network

Contributions

- * Network Processor Benchmark

- ◆ Workload characterization
- ◆ Estimation of processing requirements

- * Multiprocessor System Design

- ◆ Efficient queuing system
- ◆ Scalable processor design

- * Scheduling Algorithm

- ◆ Fair, efficient scheduling of network processor

- * Optimization of System-On-A-Chip

- ◆ Optimal processor configuration for given workload

- **Better understanding of implications from packet processing inside the network**

Workload Characterization

- * Processing requirements for processing in software:
- ◆ RISC instructions executed per byte of packet data:

HEADER	64 byte packet	HEADER	Encoding	Decoding
Routing lookup	2.1	Encryption	104	104
IP fragmentation	7.7	Compression	226	35
DRR scheduling	4.1	FEC	603	1052
TCP monitoring	10.3	JPEG coding	81	60

- ◆ Routing at 1.2 Gb/s: $1.2/8 \cdot 2.1 = 315$ MIPS
- ◆ Encryption at 1.2 Gb/s: $1.2/8 \cdot 104 = 15,600$ MIPS

- * Payload processing computationally expensive!

- ◆ Exploit parallelism with multiprocessor

- * Also measured:

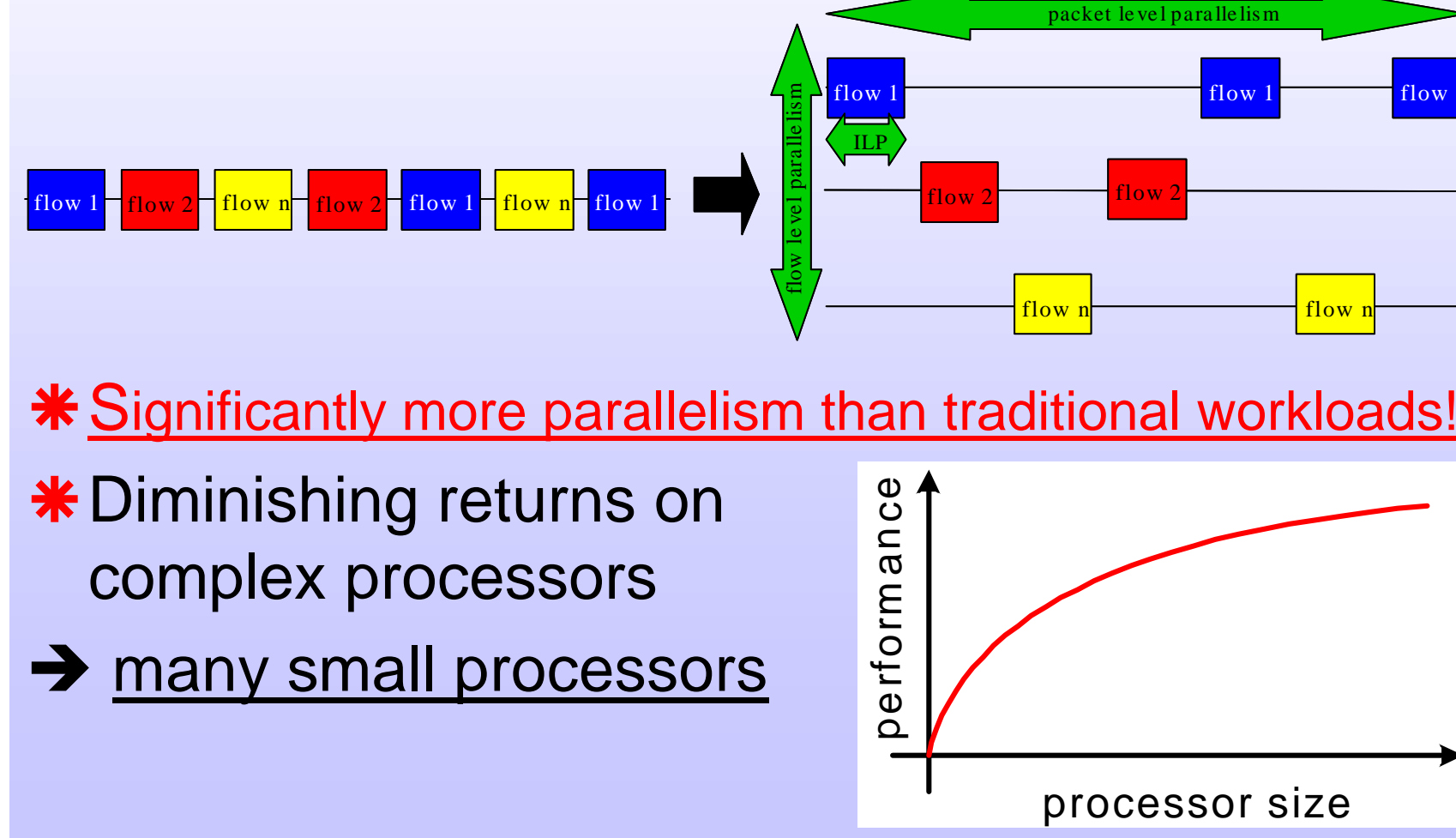
- ◆ Code size, cache performance, instruction mix [1]

Parallelism in Networks

- * Parallelism "everywhere":

- ◆ Flow-level, packet-level, instruction-level

- * Packet stream:



- * Significantly more parallelism than traditional workloads!

- * Diminishing returns on complex processors

- many small processors

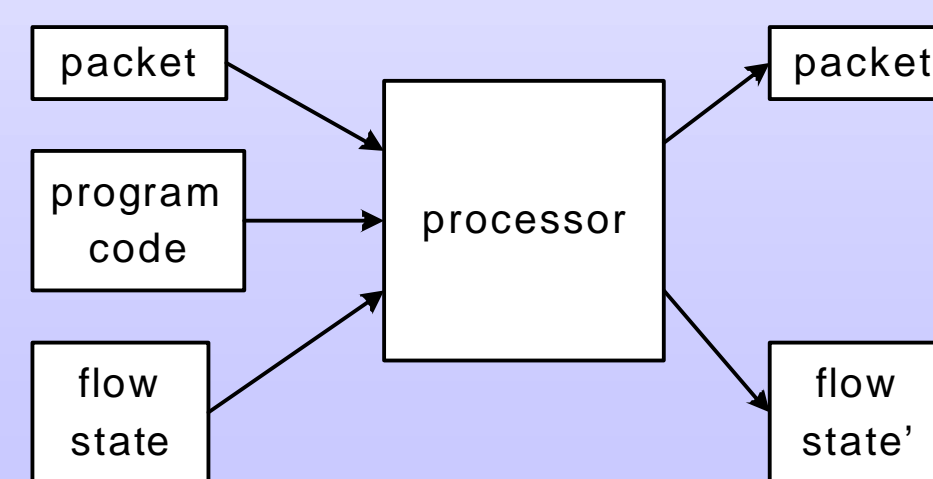
System Issues

- * System characteristics [2]:

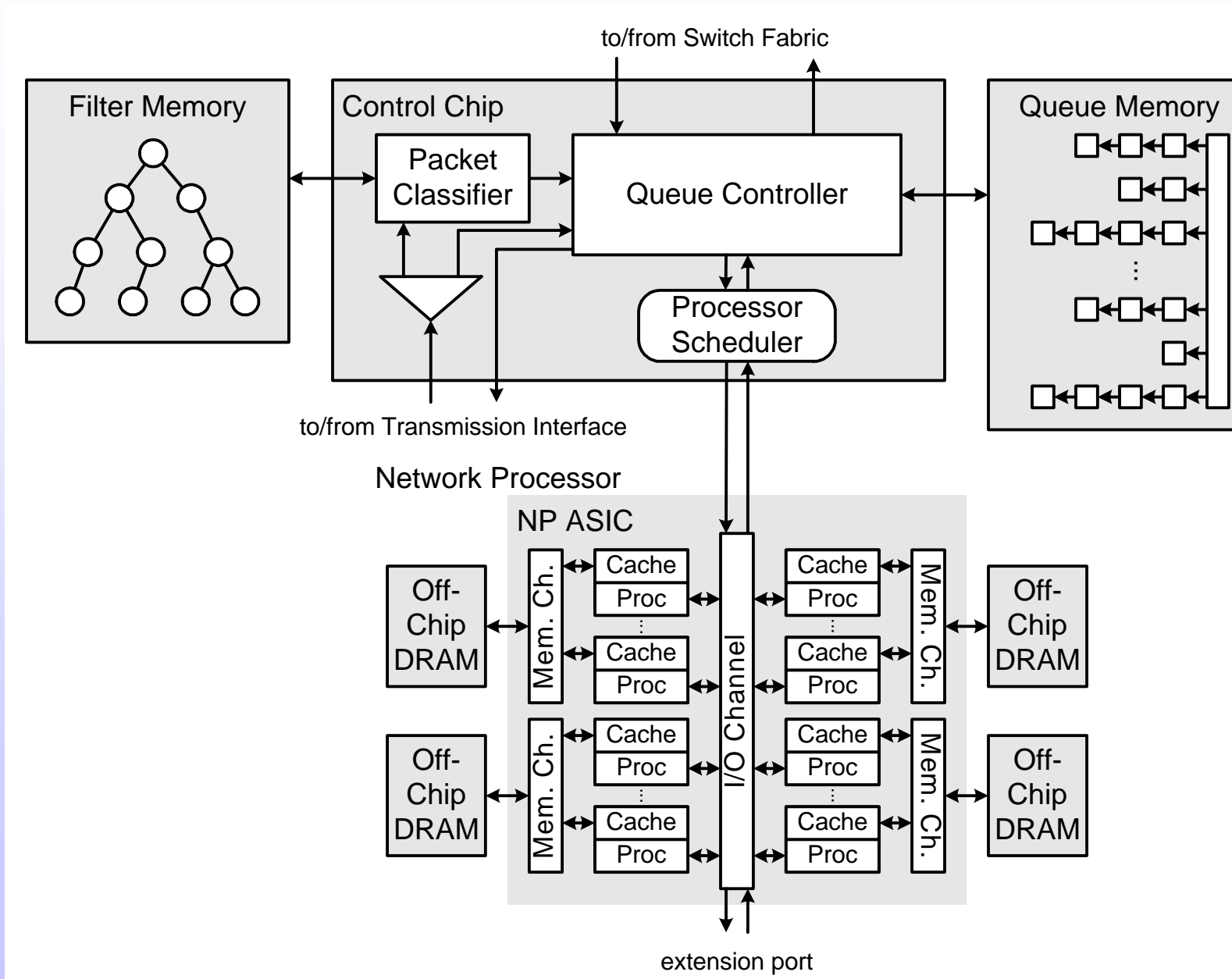
- ◆ Very short tasks (1-100 i sec)
- ◆ Large number of tasks (> 1,000,000 packets/sec)
- ◆ Highly heterogeneous applications (forwarding, MPEG coding)

- * Queuing system / OS must maintain:

- ◆ Packets
- ◆ Flow state
- ◆ Instruction code



Router Port Design

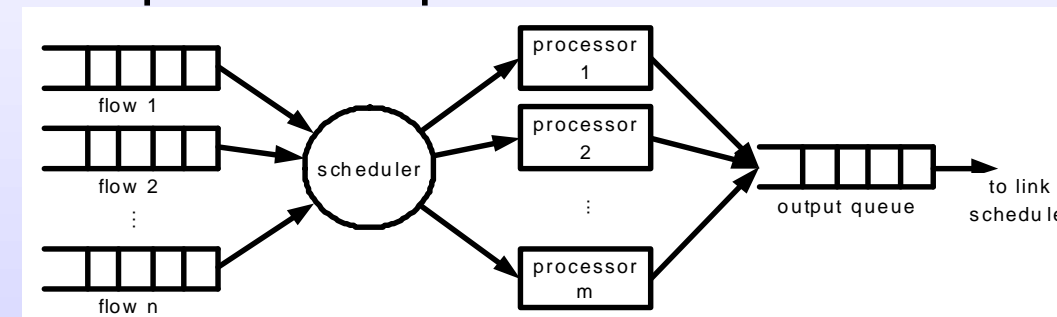


- * System-On-A-Chip: processors, cache, I/O on ASIC

Scheduling

- * Which packet to process on which processor?

- ◆ Processing time unknown!
- ◆ Enforce fairness / reservation
- ◆ Preserve packet dependencies



- * Scheduling strategy:

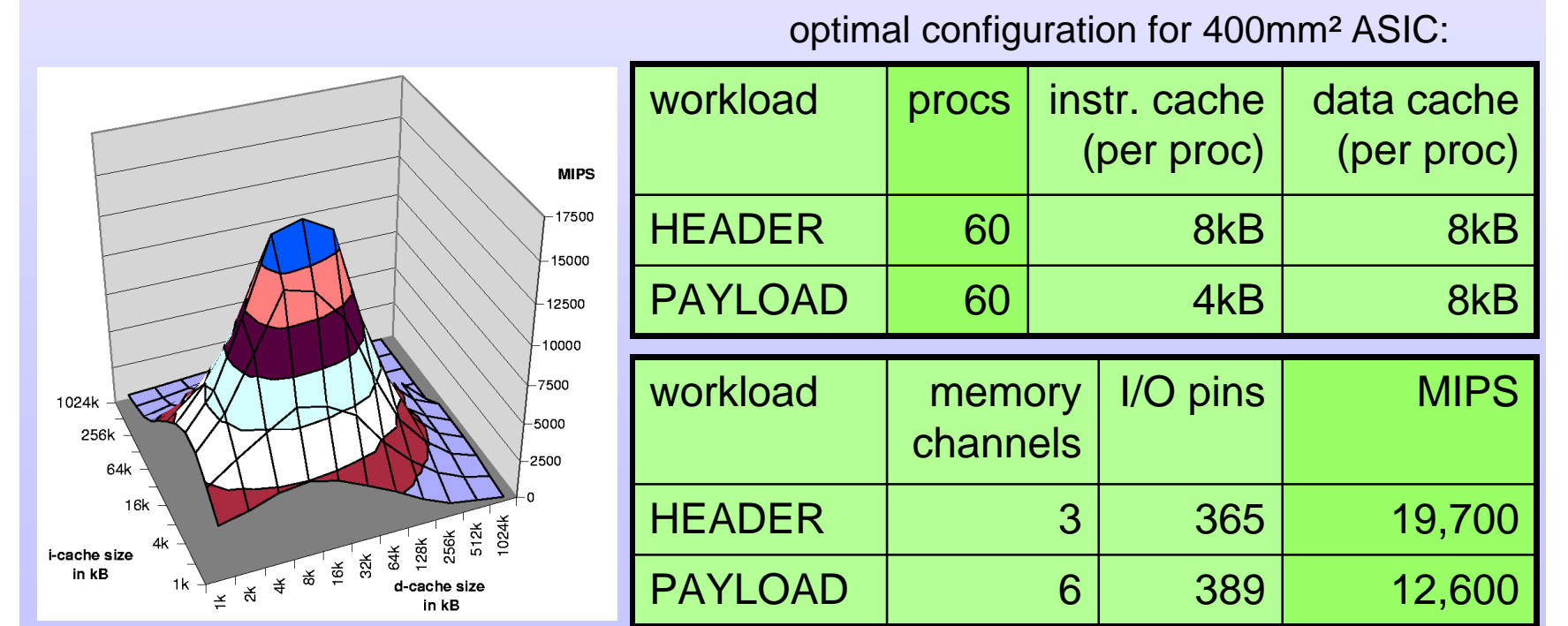
- ◆ Use processing time prediction [3]
- ◆ Exploit instruction-cache locality [4]
- ◆ Feedback actual processing time

- * Fast context-switching: context prefetch in background

- * Load balancing: just move flow-state

System-On-A-Chip Optimization

- * On ASIC: How many processors? How much memory?
- * Analytic model for optimal configuration, given workload [5]
- * Too many processors on chip:
 - Smaller caches due to chip are limit
 - Smaller caches increase cache misses
 - Higher cache misses cause more memory stalls
 - Lower performance



Status and Future Work

- * Current status:

- ◆ Completed NP benchmark analysis
- ◆ Completed system design
- ◆ Analyzed locality-aware and predictive scheduling
- ◆ Completed system-on-a-chip optimization
- ◆ Applications (sensor aggregation, web content transcoding)

- * Future work:

- ◆ Analysis of scheduling with context switching (with and without context prefetching)
- ◆ Investigation of hardware accelerators and programmable logic
- ◆ System simulator

Related Work

- * Active Networks / Programmable Networks:

- ◆ Instruction code distribution
- ◆ Safe program execution
- ◆ Resource sharing

- * Computer architecture

- ◆ Processor components (RISC cores, accelerators, ...)
- ◆ Faster memory, I/O

- * Commercial Network Processors:

- ◆ IBM PowerNP, Intel IXP1200, Motorola C-5, ...
- ◆ Few parallel processors, some hardware accelerators
- ◆ Mainly for header processing

Further Reading

- [1] Wolf, T., Franklin, M.: "CommBench - A Telecommunication Benchmark for Network Processors," *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 154-162, Austin, TX, April 2000.
- [2] Wolf, T., Turner, J.: "Design Issues for High Performance Active Routers," *IEEE Journal on Selected Areas of Communications - Special Issue on Active and Programmable Networks*, vol. 19, no. 3, pp. 404-409, March 2001.
- [3] Pappu, P., Wolf, T.: "Scheduling Processing Resources in Programmable Routers," submitted to *IEEE INFOCOM 2002*.
- [4] Wolf, T., Franklin, M.: "Locality-aware Predictive Scheduling of Network Processors," to appear in *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Tucson, AZ, November 2001.
- [5] Wolf, T., Franklin, M., Spitznagel, E.: "Design Tradeoffs for Embedded Network Processors," *Tech. Report WUCS-00-24*, Dept. of Computer Science, Washington University in St. Louis, MO, July 2000.