

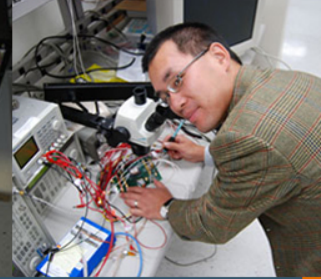


# Scalable Architectures for Large Switches

**Wladek Olesinski**

Staff Engineer

Sun Laboratories



**2008  
Sun Labs  
Open House**



# Outline

- The problem: scalable architectures for large, single-stage switches
- Data path: Output Buffered Switch with Input Groups (OBIG)
- Scheduler: Parallel Wrapped Wave Front Arbiter with Fast Scheduler (PWWFA-FS)
- Simulation results
- Future work and conclusions
- References

# Challenges of Large Switches

- IO limitations
  - > A limited number of ports per chip
- Memory limitations
- Scheduling difficulties

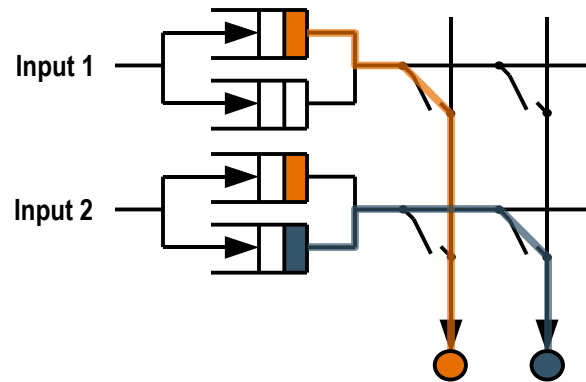
Our approach: distribute a switch over multiple chips connected with high-speed interconnects

# The Existing Solutions

- Large switches require scalable data paths and fast schedulers
- Typical architectures:
  - > Crossbars
  - > Buffered crossbars
  - > Multi-stage switches

# Crossbar Switch

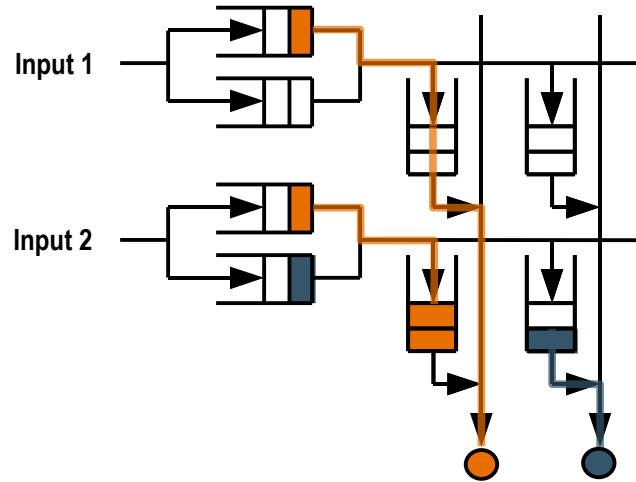
## 2x2 crossbar with Virtual Output Queues (VOQs)



### Coordinated Scheduling

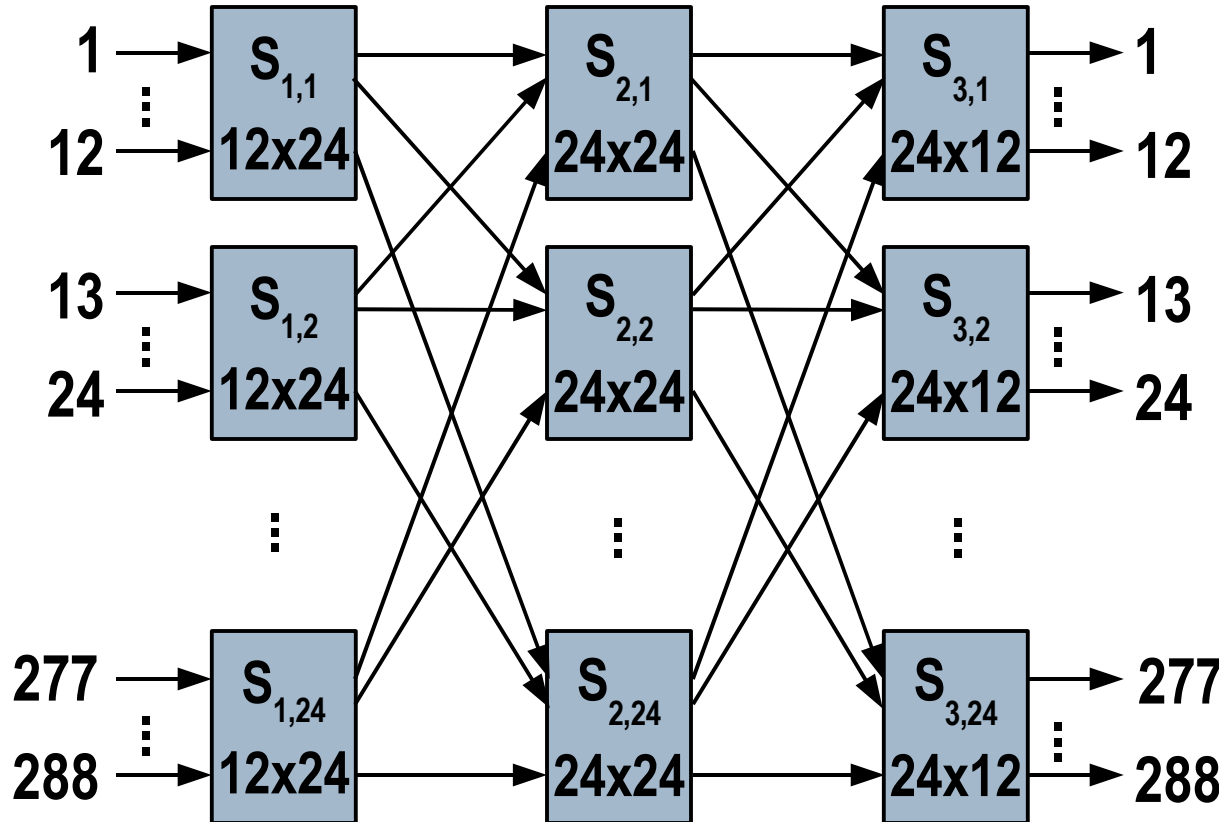
Crossbars do not scale well because of a large number of crosspoints and difficult scheduling

# Buffered Crossbar Switch



Buffered crossbars simplify scheduling but  
require a lot of memory

# Non-blocking Multi-stage Switch



- 72 24-port switches
- 3 stages
- 1,152 internal links

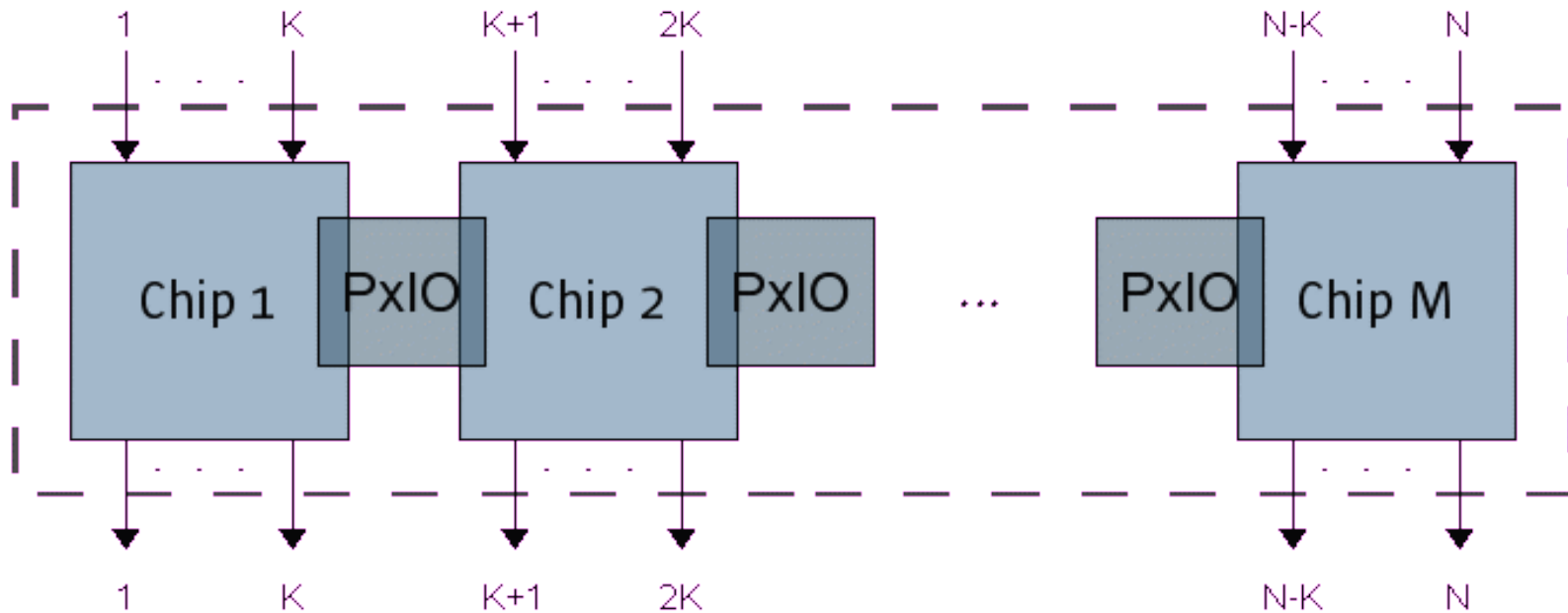
Long latencies, easily saturates, high connectivity, difficult to schedule

# Our Approach

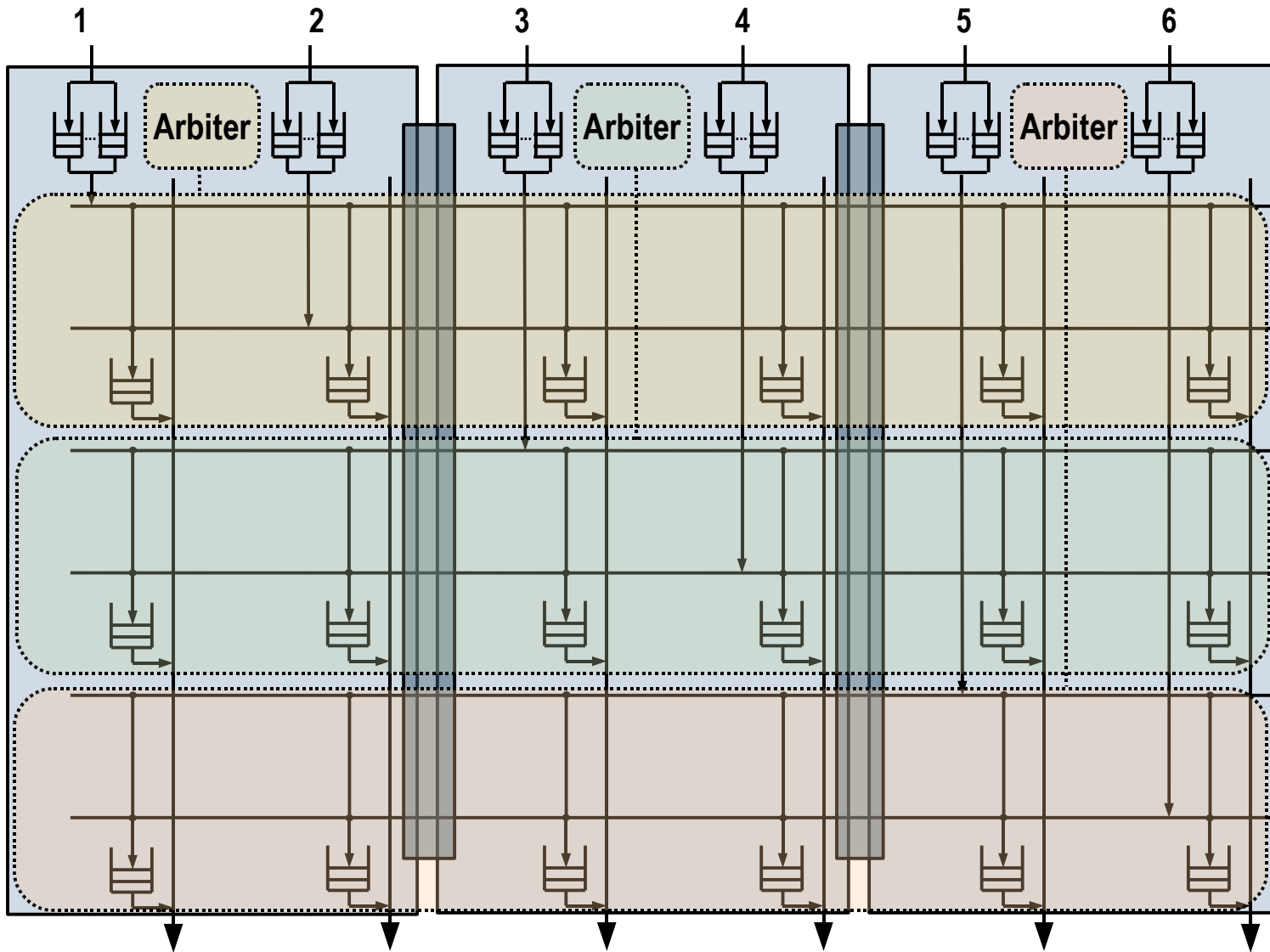
- Multi-chip module based on fast interconnect (e.g., Proximity Communication)
- Partition a crossbar into several slices, one per chip
- Put two types of schedulers in each chip
  - > The input scheduler arbitrates between local input ports and all output ports
  - > The output scheduler collects cells from several buffers (similar to a buffered crossbar)
- The outcome: scalable switch architecture

# The Architecture

- The switch consists of  $M$  chips
- Every chip has  $K$  inputs/outputs
  - > Total number of ports is  $N=KM$



# Partitioned Buffered Crossbar



# Advantages of OBIG Architecture

- It is scalable because a crossbar is distributed among multiple chips
  - > Each chip has a subset of input ports and a reduced number of crosspoints
  - > An input scheduler does not deal with all input ports but just a local subset
  - > When compared with a buffered crossbar, memory requirements are reduced from  $O(N^2)$  to  $O(NM)$
- OBIG combines the advantages of conventional and buffered crossbars

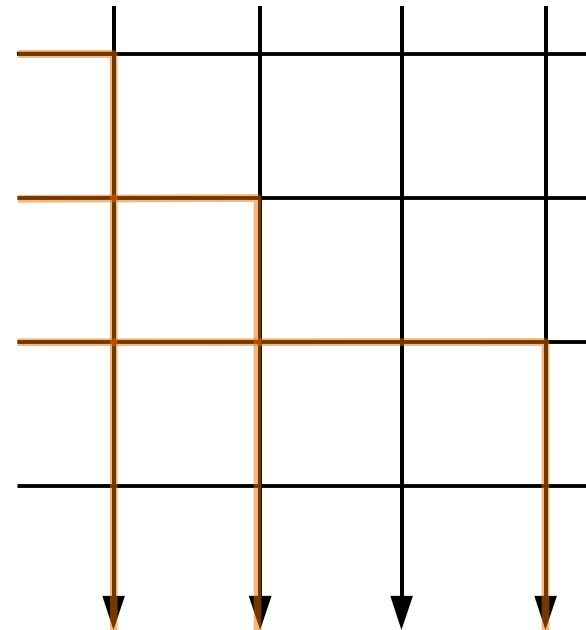
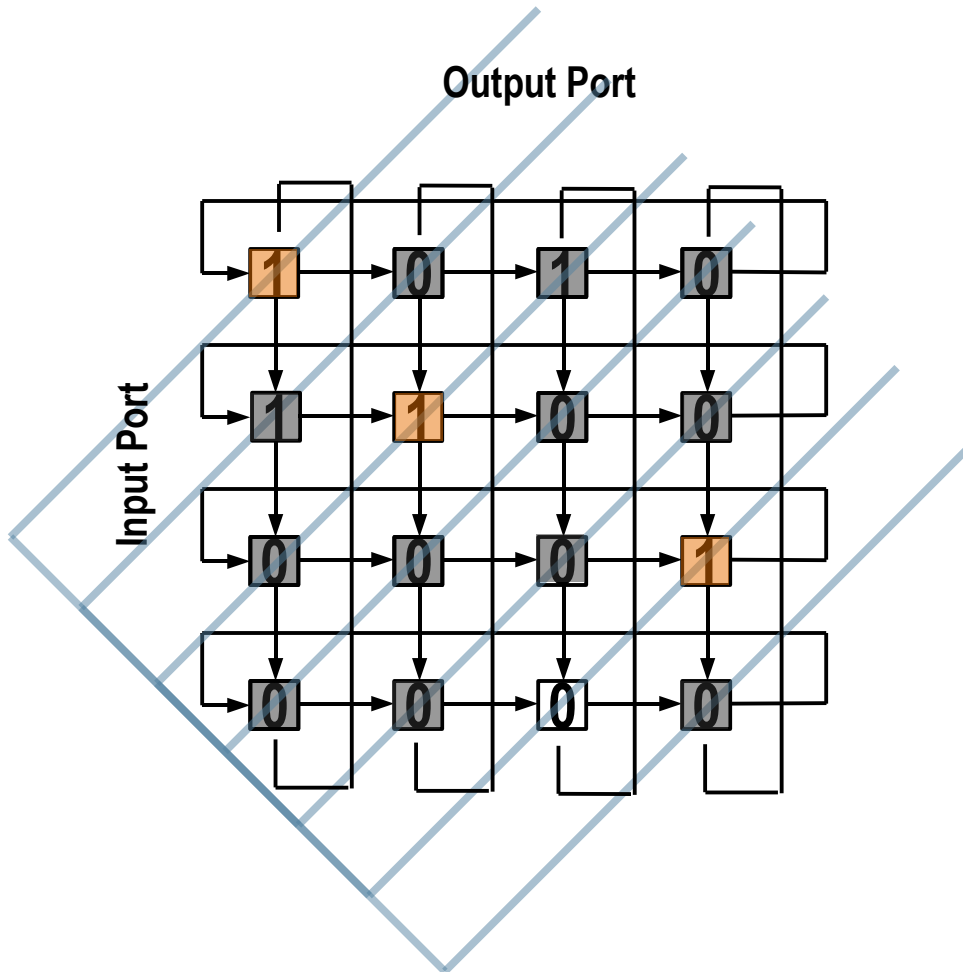
# Input Scheduler

- We have a scalable data path but what about the scheduler?
- PWWFA-FS: Parallel Wrapped Wave Front Arbiter with Fast Scheduler

# Our Approach to Scheduling

- Take Wrapped Wave Front Arbiter (*WWFA*)
- Make it parallel such that it schedules multiple slots concurrently
- The outcome: Parallel Wrapped Wave Front Arbiter (*PWWFA*)

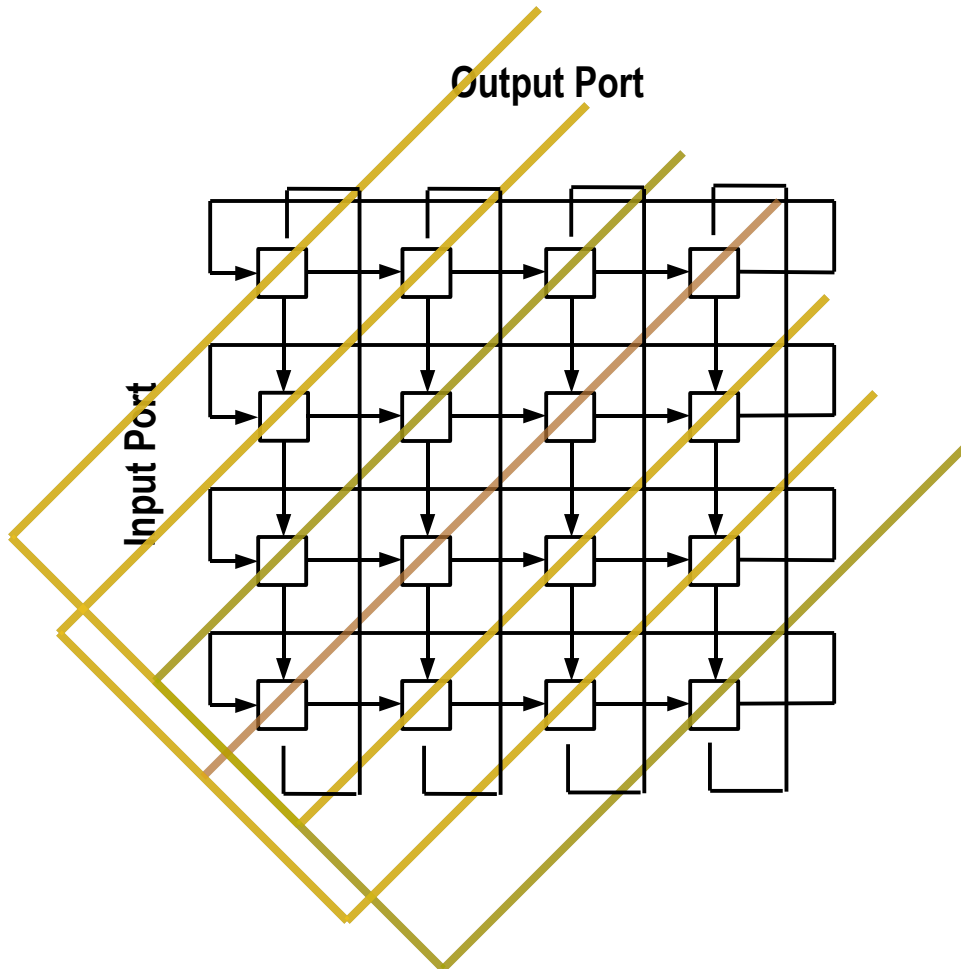
# Wrapped Wave Front Arbiter



# WWFA Observations

- Assuming
  - >  $N$  inputs/outputs
  - > processing of one wave takes time  $T$ 
    - $\Rightarrow$  WWFA produces one schedule every  $NT$
- Scheduling phase  $NT$  must be less than cell transmission time
- In a large switch, WWFA is too slow
  - > Assuming  $N=256$ , capacity  $C=10$  Gbps, cell length  $L=128$  bytes,  $T$  would have to be less than 0.4 ns

# Parallel Wrapped Wave Front Arbiter



- Time  $NT$ : 1<sup>st</sup> schedule ready
- Time  $NT+T$ : 2<sup>nd</sup> schedule ready

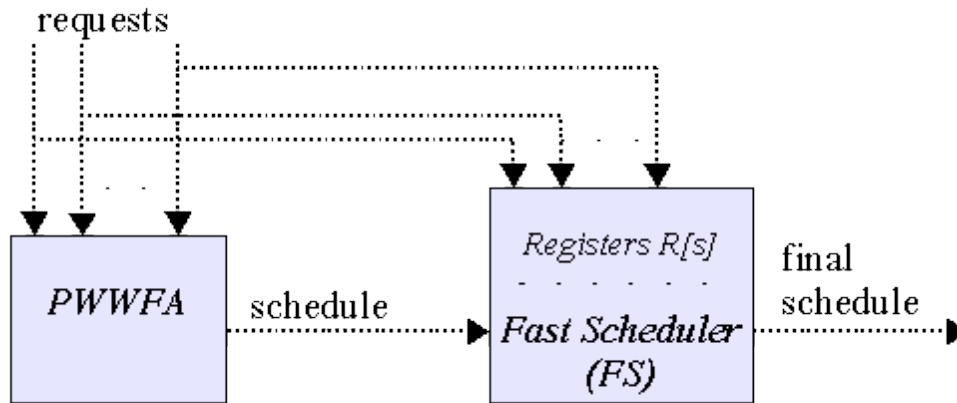
...and so on. In the next 2 periods of  $T$ , the other 2 subschedulers produce schedules.

To provide fairness, subschedulers start at different waves in every scheduling cycle.

# PWWFA Observations

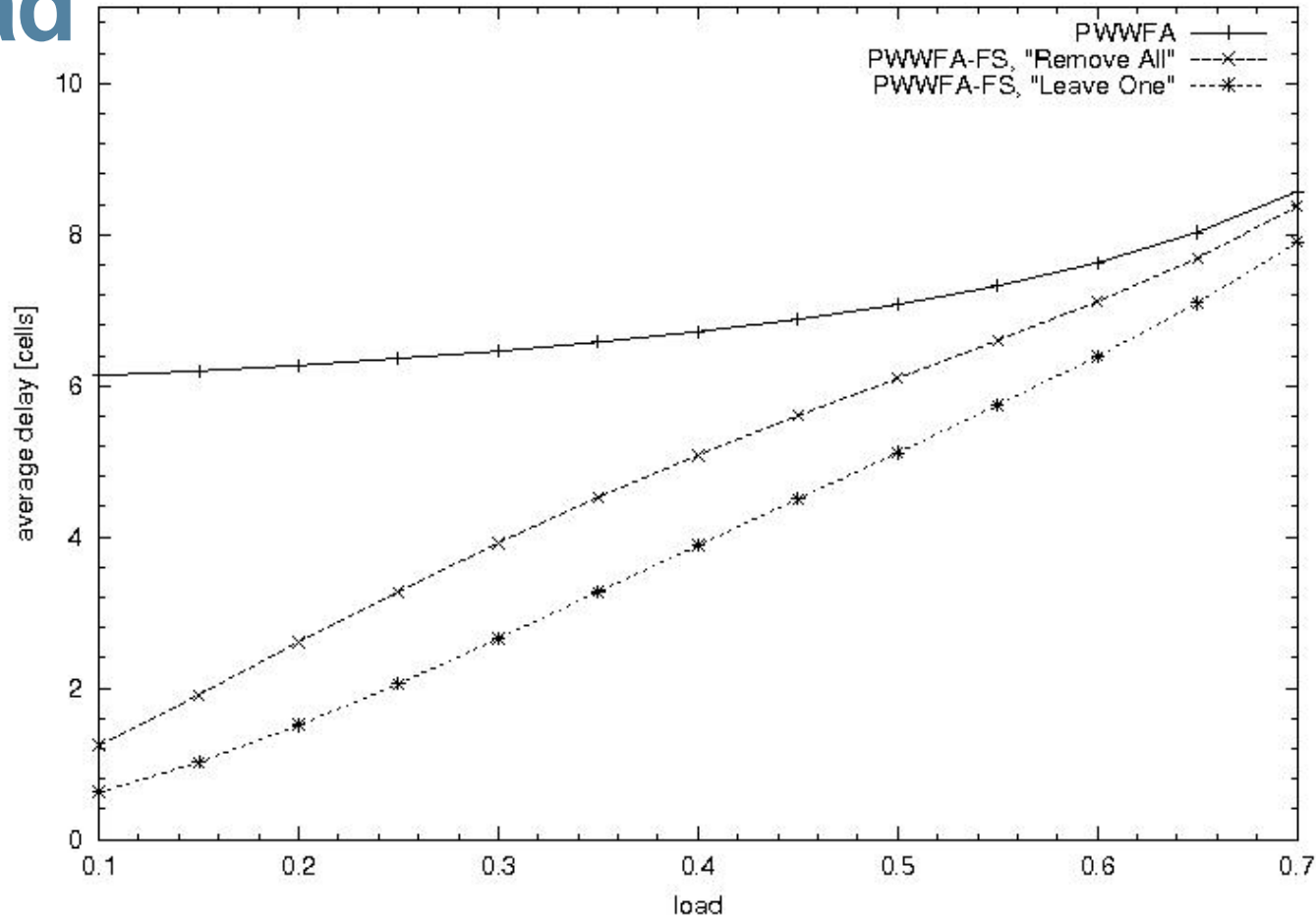
- Assuming
  - >  $N$  inputs/outputs
  - >  $N$  subschedulers
  - > processing of one wave takes time  $T$ 
    - => PWWFA may generate one schedule every  $\underline{T}$
- Compared with WWFA, throughput is improved by a factor of up to  $N$
- PWWFA can be implemented in a large switch
  - > With assumptions like before,  $T$  has to be less than 102.4 ns

# Fast Scheduler – Low Latency



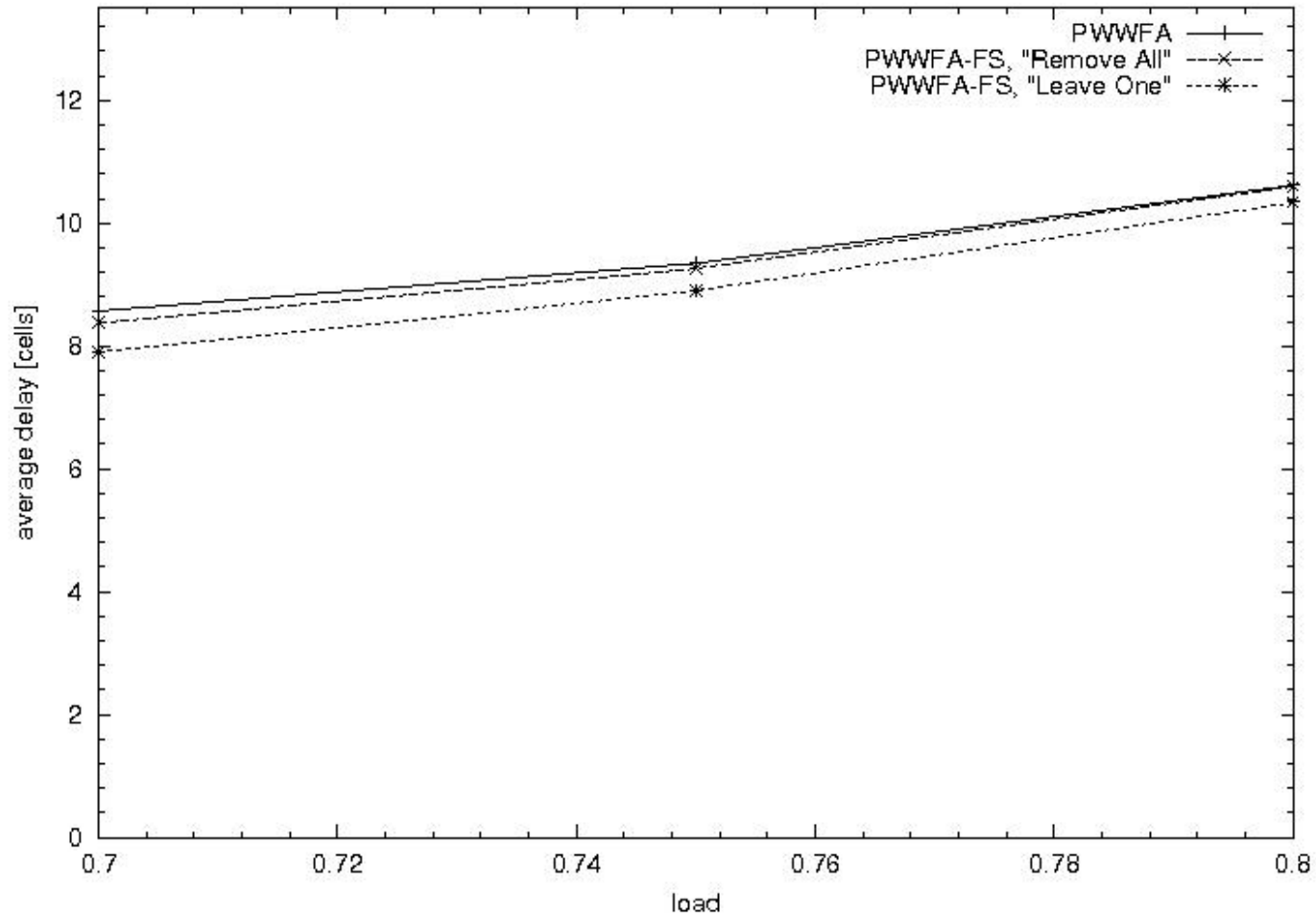
- Enhance PWWFA schedule with grants to most recent request
- Significant performance improvement for low to medium load

# PWWFA Performance – 10% to 70% Load



- Bernoulli traffic, N=256

# PWWFA Performance – 70% to 80% Load



- Bernoulli traffic,  $N=256$

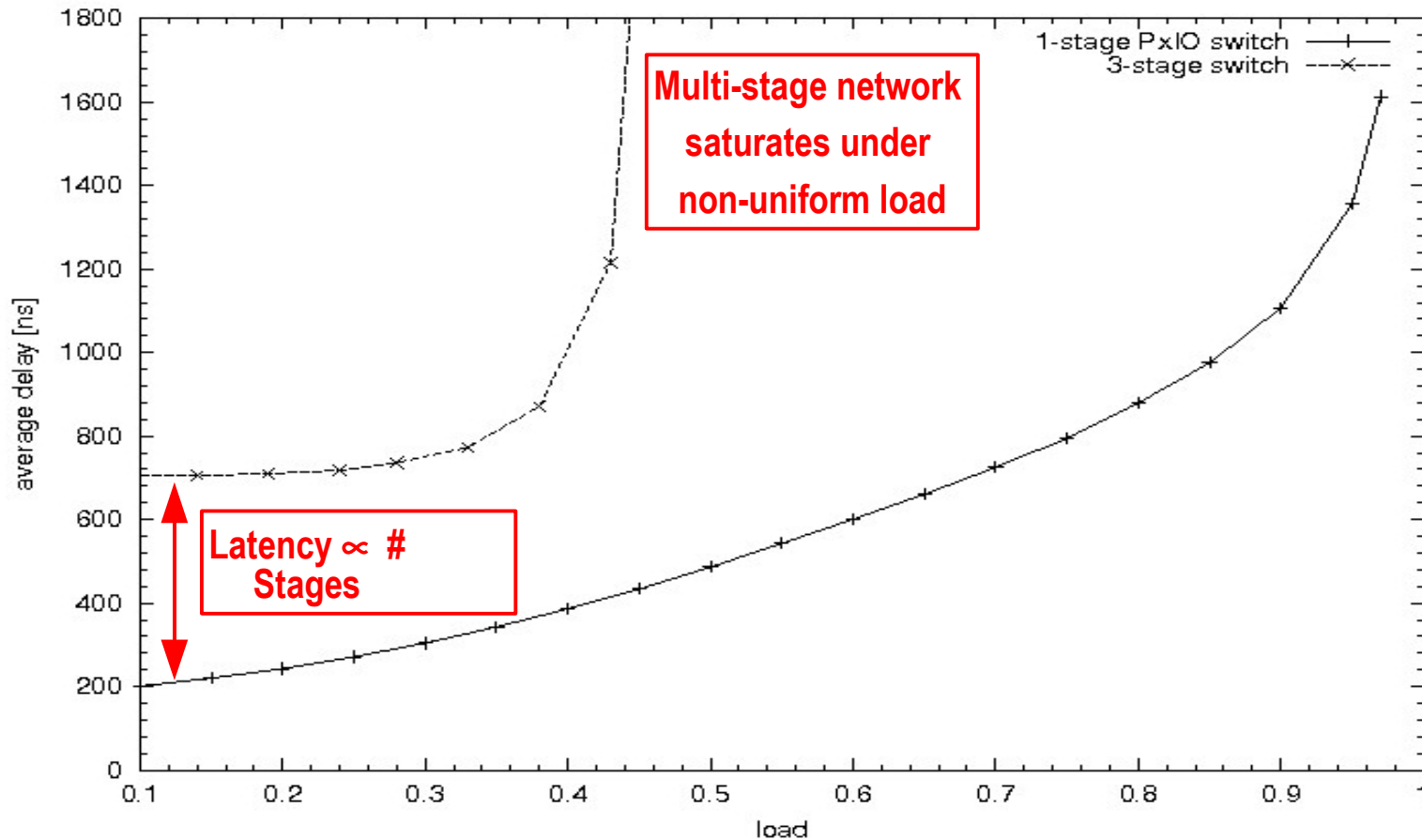
# PWWFA-FS Observations

- Pipelines the well-known WWFA scheduler
- Fast Scheduler (FS) reduces latencies especially under light loads
- Provides latency comparable to shared memory switch

# Simulation Results of OBIG with PWWFA-FS and 3-stage Switch

- Assumptions
  - > N=288 ports, 10Gbps ports --> 2.9Tbps
  - > OBIG architecture with M=16 chips and PWWFA-FS
  - > 3-stage Clos network
  - > Hot spot traffic

# Comparison of OBIG and Multi-stage Switch



Joint work with University of Valencia (Prof. José Duato, Prof. José Flich, Pedro Garcia)

# Conclusions

- OBIG is a practical, scalable architecture that distributes a crossbar over multiple chips using a fast interconnect like PxC
- PWWFA-FS is a practical, scalable scheduler that provides low latency and high throughput
- Promising simulation results of a large, flat, 256-port, 2.5Tbps switch based on OBIG with PWWFA-FS

# References

- W. Olesinski, H. Eberle, and N. Gura, *PWWFA: Parallel Wave Front Arbiter for Large Switches*, HPSR'07, Brooklyn, New York, May 30-June 1, 2007
  - > patent filed
- H. Eberle, A. Chow, B. Coates, J. Cunningham, R. Drost, J. Ebergen, S. Fairbanks, J. Gainsley, N. Gura, R. Ho, D. Hopkins, A. Krishnamoorthy, J. Lexau, W. Olesinski, J. Schauer, and T. Ono, *Multiterabit Switch Fabrics Enabled by Proximity Communication*, Hot Chips'07, Stanford University, August 19-21, 2007
  - > patent filed

# References, cont'd

- W. Olesinski, H. Eberle, and N. Gura, *OBIG: the Architecture of an Output Buffered Switch with Input Groups for Large Switches*, GLOBECOM'07, Washington, DC, Nov 26-30, 2007
  - > patent filed
- W. Olesinski, N. Gura, H. Eberle, and A. Mejia, *Low-Latency Scheduling in Large Switches*, ANCS'07, Orlando, Florida, Dec 3-4, 2007
  - > patent filing in progress
- W. Olesinski, H. Eberle, and N. Gura, *Flow Control in Output Buffered Switch with Input Groups*, To appear at HPSR'08, Shanghai, China, May 15-17, 2008

# References (Stanford Collaboration)

- A. Dua, N. Bambos, W. Olesinski, H. Eberle, and N. Gura, *Backlog Aware Low Complexity Schedulers for Input Queued Packet Switches*, Hot Interconnects'07, Stanford University, August 22-24, 2007
- A. Dua, B. Yolken, N. Bambos, W. Olesinski, H. Eberle, and N. Gura, *Backlog Aware Scheduling for Large Buffered Crossbar Switches*, To appear at ICC'08, Beijing, China, May 19-23, 2008



**Wladek Olesinski**  
wladek.olesinski@sun.com



**2008  
Sun Labs  
Open House**

