

# Data Reservoirシステムと GRAPE-DRシステム: 通信と計 算の超高速化を目指して

平木 敬

東京大学情報理工学系研究科  
創造情報学専攻

# Our goal

- HPC system as an infrastructure of scientific research
  - Performance of HPC system influence results and quality of research
  - Simulation, data intensive computation, searching and data mining
- HPC systems for real scientists
  - High-speed, low-cost, and easy to use
    - High-speed computation
    - High-speed global networking
    - OS/ ALL IP network technology

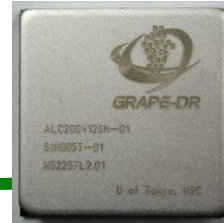
GRAPE-DR  
project

# Our simplified goal

- World fastest system
  - Computation We are now challenging
  - Network We already achieved  
(Internet2 Land Speed Records)
- Practical systems for scientists
  - For real scientists
  - Support of daily research works

# Our activities on Next generation Supercomputing system

1 chip ---512Gflops,  
1 system -- 2Pflops



## GRAPE-DR processor

- very fast (512 Gflops/chip)
- low power (8.5Gflops/W)
- low cost

(University of Tokyo  
National Astronomical  
Observatory Japan)

## Next generation Supercomputing system

## All IP computer architecture

- All components of computers  
Have own IP, and connect  
freely
- Processor, memory,  
Display, keyboard, mouse  
And disks etc.

(Keio Univ. Univ. of Tokyo)

University of Tokyo



## Long distance TCP technology

- same performance  
From local to global  
communication

• 30000Km, 1.1GB/s  
(U. of Tokyo • WIDE)

Kei Hiraki

# GRAPE-DR project (2004 to 2008)

- Development of practical MPP system
  - Efficient Pflops systems for poor real scientists
  - Target performance of GRAPE-DR            2  
Pflops(2008)
- Sub projects
  - Processor system
    - Processor chip、 Supercomputing system
  - System software
    - Compiler, networking software
  - Application software

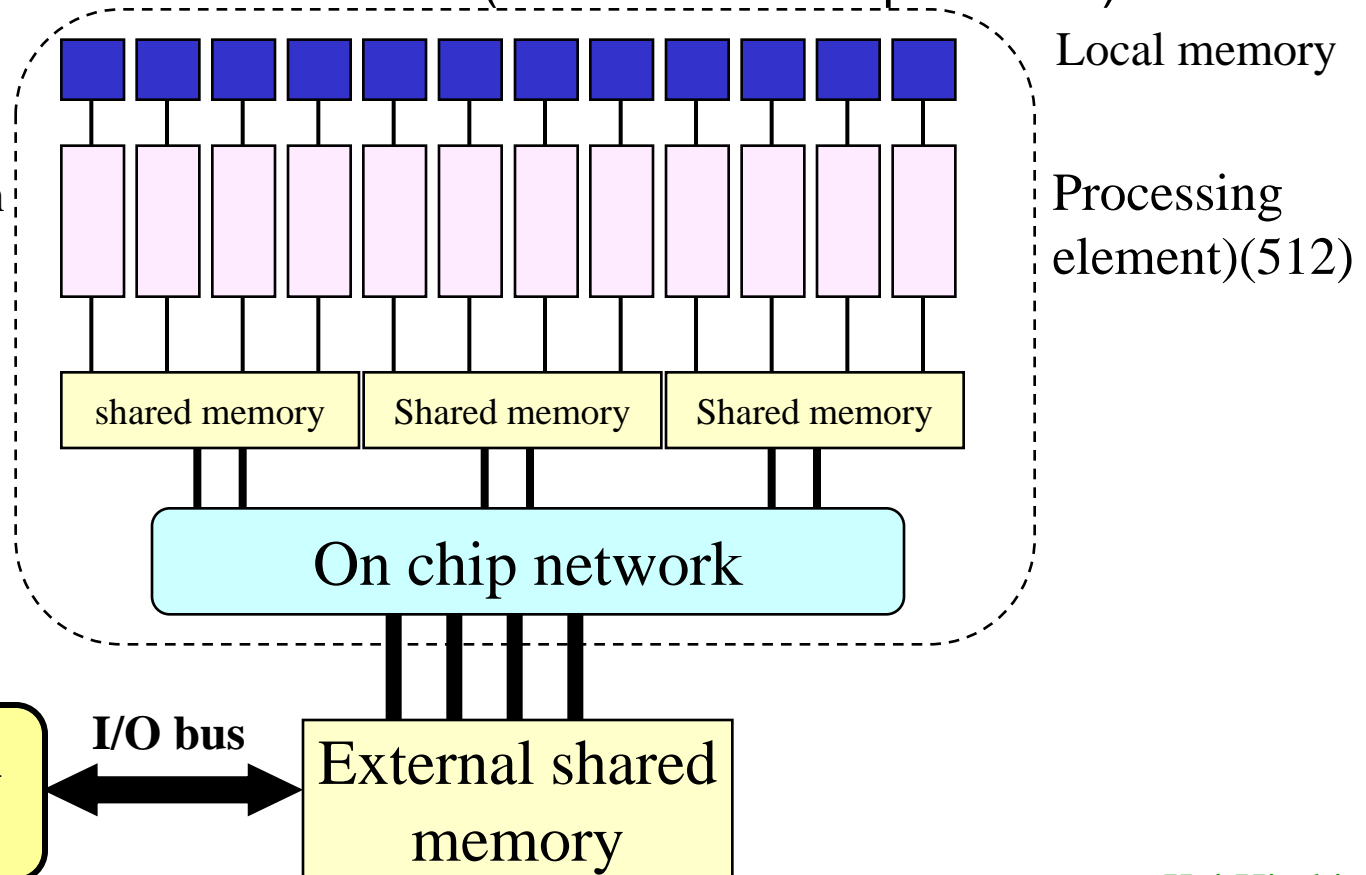
# GRAPE-DR Processor chip

- SIMD like architecture
- 512x64 bit arithmetic units + shared memory + broadcast/reduction network
  - Elimination of inter-PE interconnection
  - Dedicated Reduction network (with arithmetic operations)

- Integer operation
- Floating point operation
- Conditional execution by mask register

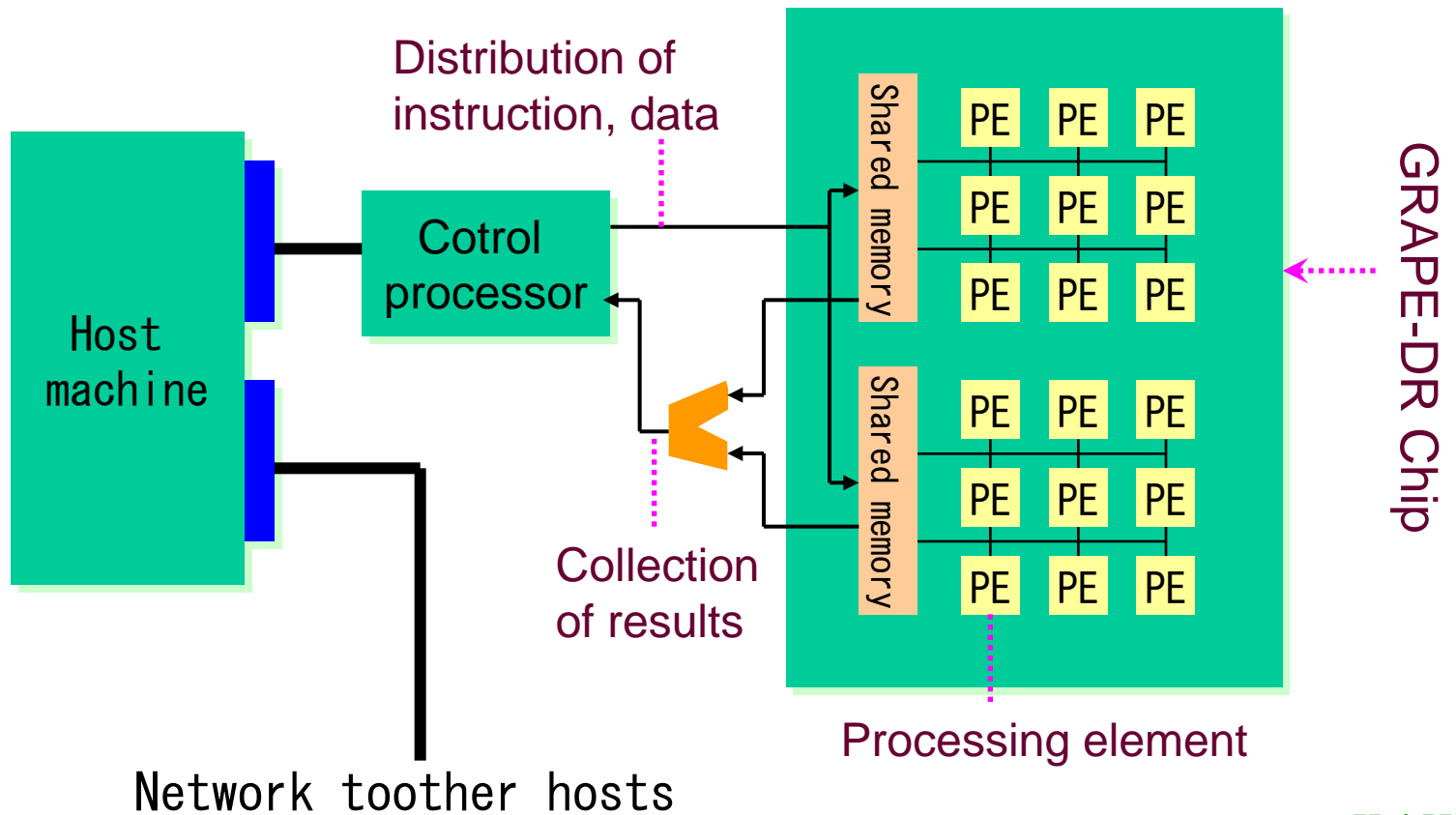
Broadcasting memory accesses

Reduction



# Basic architecture

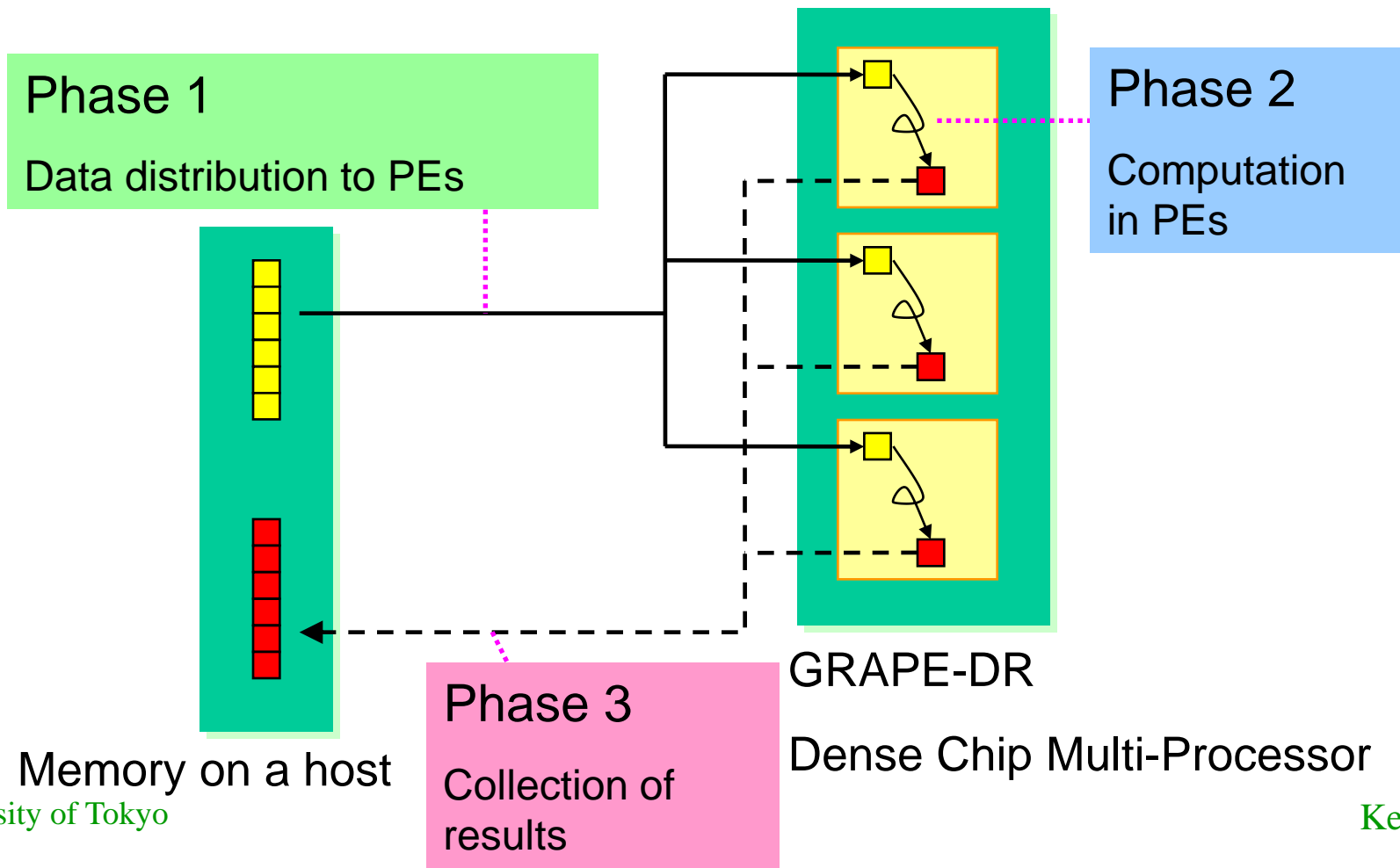
- Instruction execution between data from shared memory and local data
- All the PE in a chip execute the same instruction
- Conditional execution by mask registers



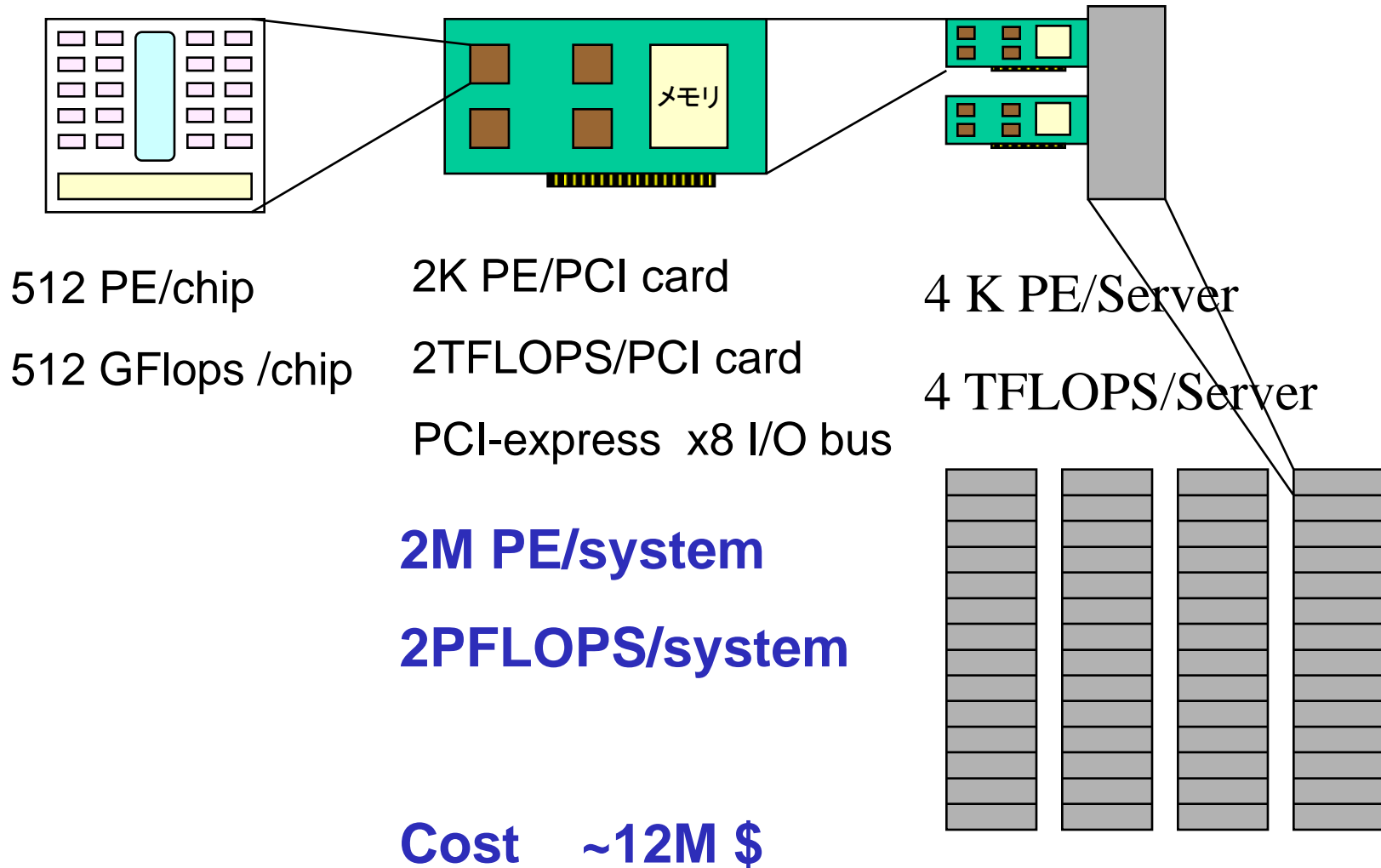


# Pipelined program execution

- Pipeline of three execution phases

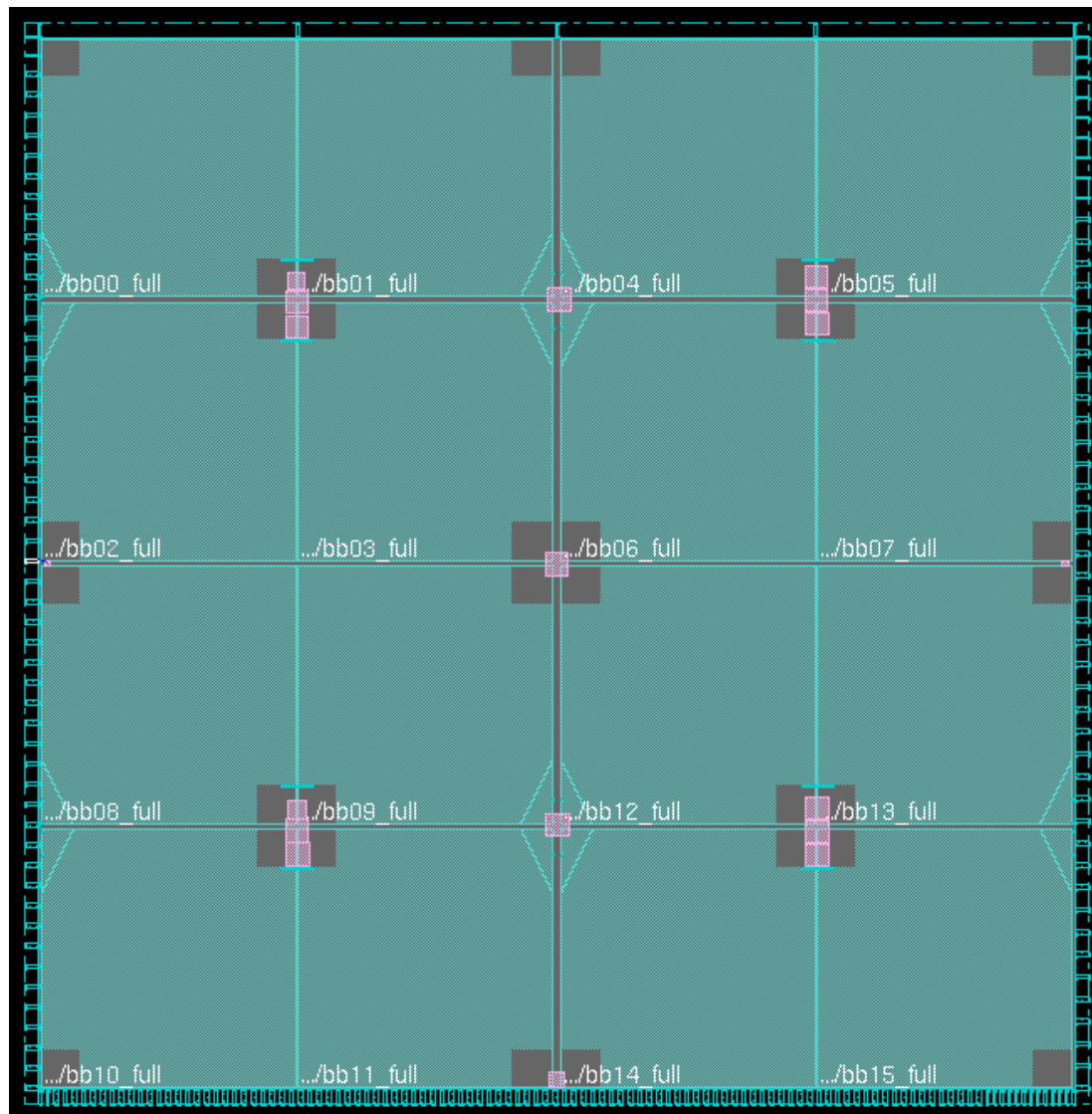


# GRAPE-DR: system architecture



# Floor plan of GRAPE-DR chip

- 90nm CMOS
- 18mm x 18mm  
(320mm<sup>2</sup>)
- ~ 400M Tr
- BGA 725, 280 signals
- First sample 2006



# GRAPE-DR Processor chip

Engineering Sample (2006)

512 Processing Elements/Chip

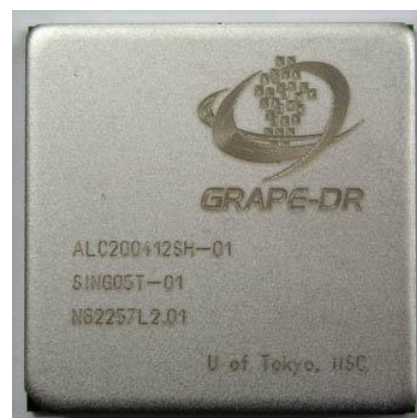
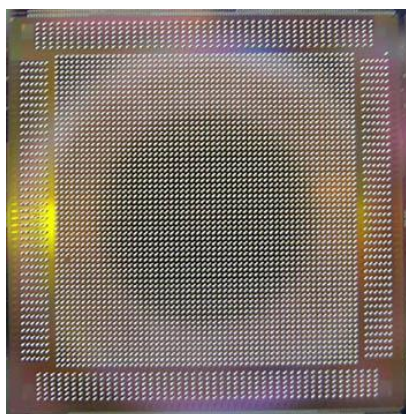
Working on a prototype board (designed speed)

500MHz、512Gflops

Power Consumption max. 60W

Idle 30W

(Lowest power per computation)



# GRAPE-DR 1 chip board

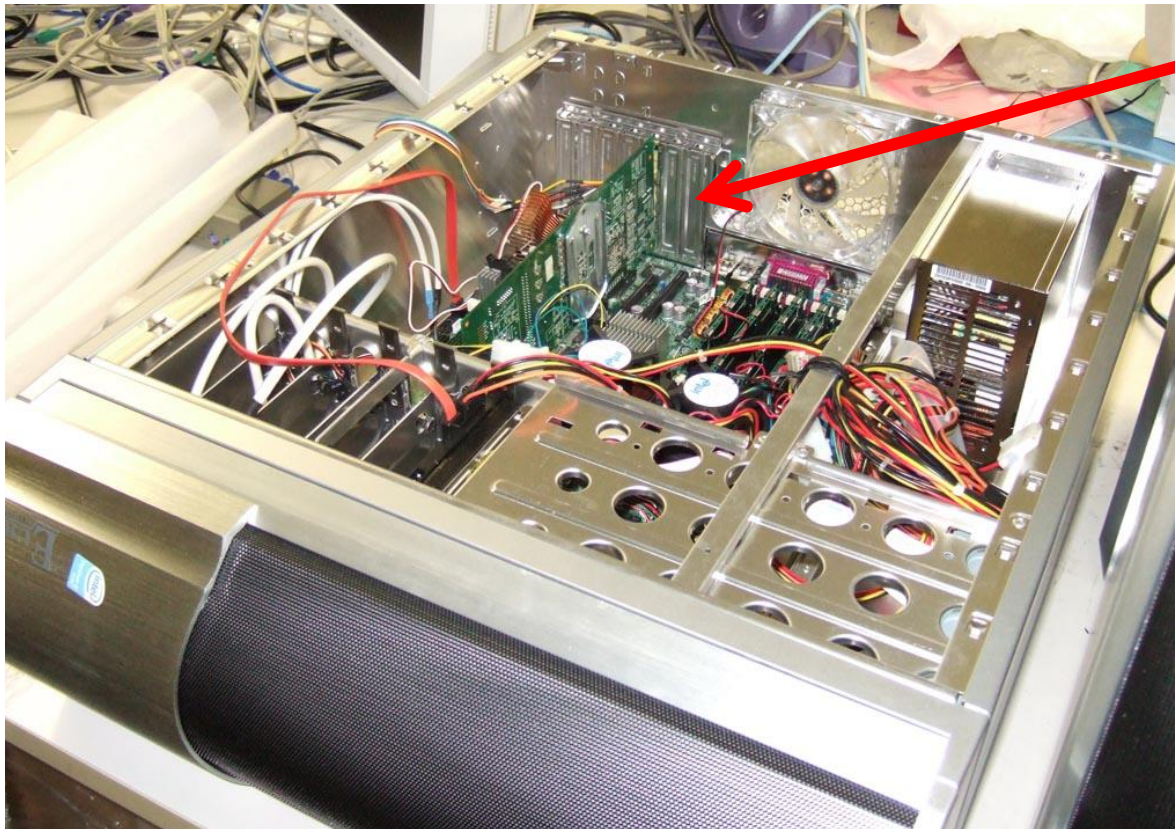
- Evaluation board (1 GRAPE-DR chip, PCI-express bus)



- 4 chip board will be ready this September

# GRAPE-DR Prototype board

- For evaluation (1 GRAPE-DR chip, PCI-X bus)
- Next step is 4 chip board



GRAPE=DR  
Card

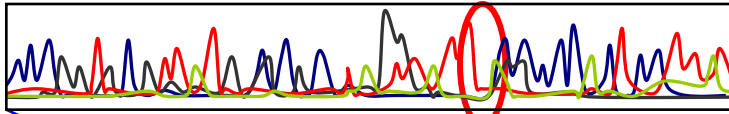
# System software

- GRAPE-DR optimizing compiler
  - Automatic parallelization with global analysis
  - Generation of multi-threaded
- Compiler Ver.1 flat-C compiler(2005)
  - C language + explicit parallel construct
- Compiler Ver.2 Sakura-C compiler (2006)
- Currently, prototype is working
  - Basic optimization (PDG, flow analysis, pointer analysis etc.)
  - Currently working on GRAPE-DR prototype

# Application software(1)

## Sequence matching for high-speed search of SNPs

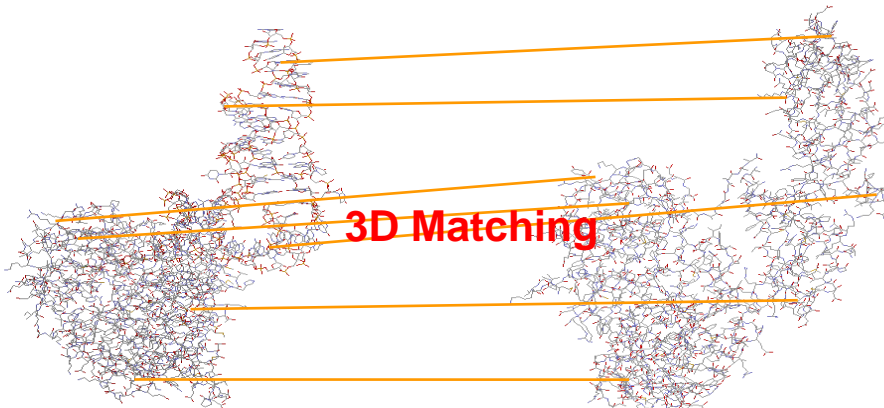
Newly sequenced data



```
caaagtgtgcggttctttgtagccccgaaa  
gttcgcaactgaacgggtccggatatttggttt  
tccacggcacaaaaaccaatcaagtgcgcc  
aatcaaaaaagtagtaatcaaaactgggaa  
atccgggaaataatatgtgaaaataatac  
gttgcctaaaagccattaagagagggccga  
acgcttatagagagctatagagtgaaagct  
gagaagaacccaaaacggagcataaacatga
```

## Graph matching

Find high-level structure of protein, etc



E. Coli Cysteinyl-tRNA

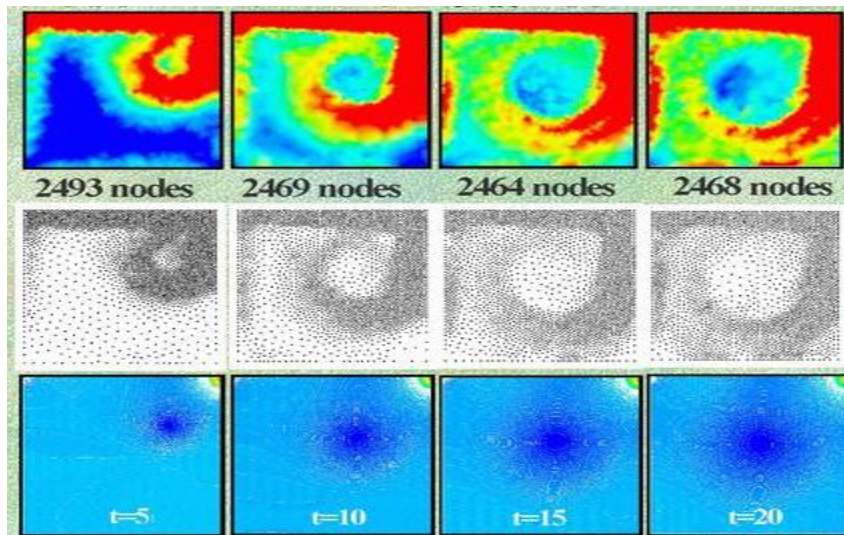
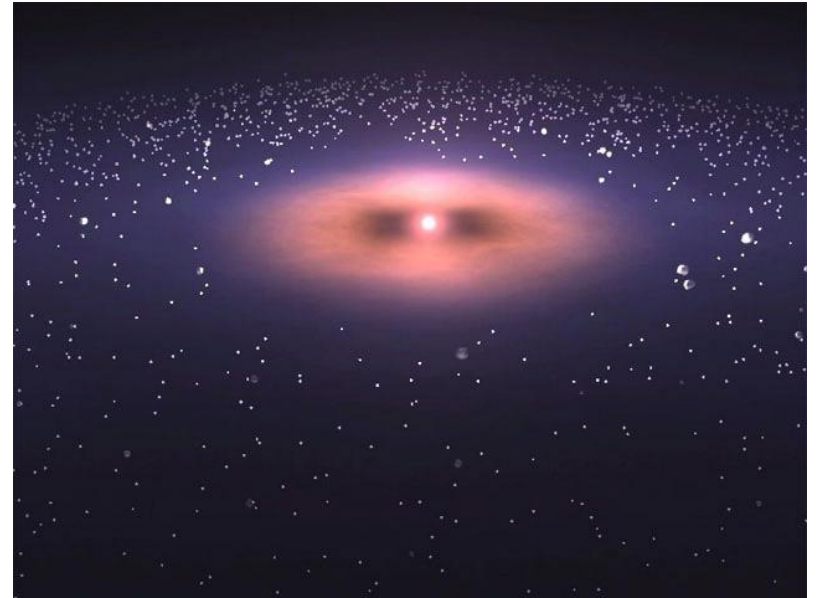
Elongation Factor G In Complex With GDP

Necessary elements

- Comparison of 3D structure
- High-speed graph matching
- Fast parallel searching

## Application software (2)

- N-body simulation
  - Generation of a planet
  - Simulation of a galaxy
  - N-body simulation of  $10^8$  particles

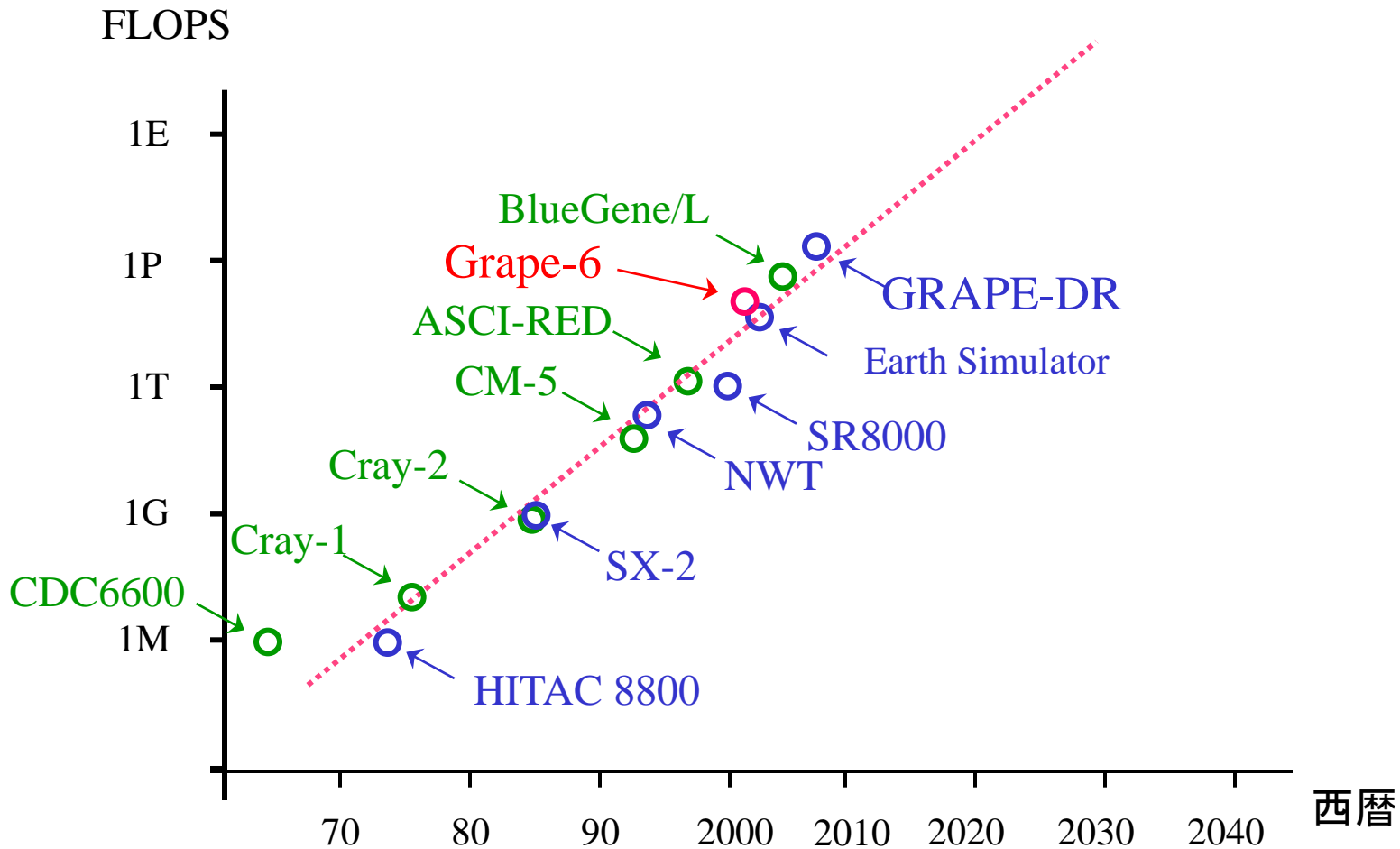


- CFD by Free mesh method
  - Suitable to highly parallel processing
  - High precision

# Comparison

- World fastest computing chips
  - GRAPE-DR (**512Gflops**(single) 、 **256Gflops**(double))
    - nVIDIA 8800 500Gflops(single), 0Gflops(double)
    - IBM/Sony Cell 256Gflops(single) 、 25Gflops(double)
    - IA32 50~Gflops(single、 double)
    - NEC SX-8R 32Gflops(single、 double)
    - ClearSpeed CSX-600 48Gflops(single、 double)
  - Number of PE in a chip (**512PEs**)
    - nVIDIA 8800 128 PEs
    - ClearSpeed CSX-600 96PEs
  - Power efficiency (**8.5Gflops/W**)(Chip number)
    - ClearSpeed CSX-600 2.5Gflops/W
    - IBM BlueGene/P 0.54Gflops/W
    - NEC SX-8 0.13Gflops/W

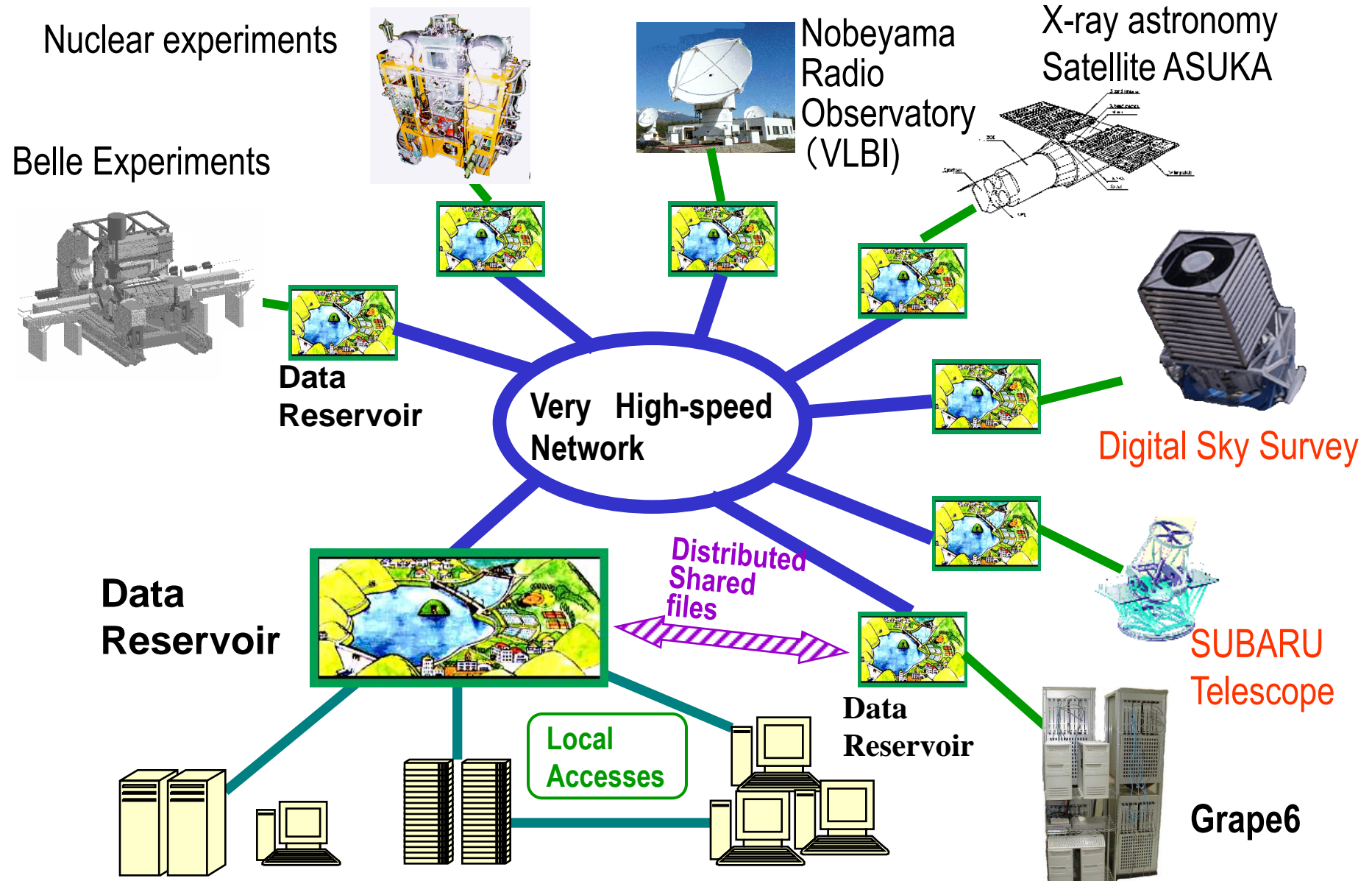
# History of speedup (some important systems)



# Next target: Systems in 2017

- Eflops system
  - Necessary condition
    - Cost  $\sim 1000\$/\text{Tflops}$
    - Power consumption (30Gflops/W  $\rightarrow$  30MW)
  - GRAPE-DR technology
    - 45nm CMOS  $\rightarrow$  4Tflops/chip, 40Gflops/W
    - 22 or 16nm CMOS  $\rightarrow$  16Tflops/chip, 160Gflops/W

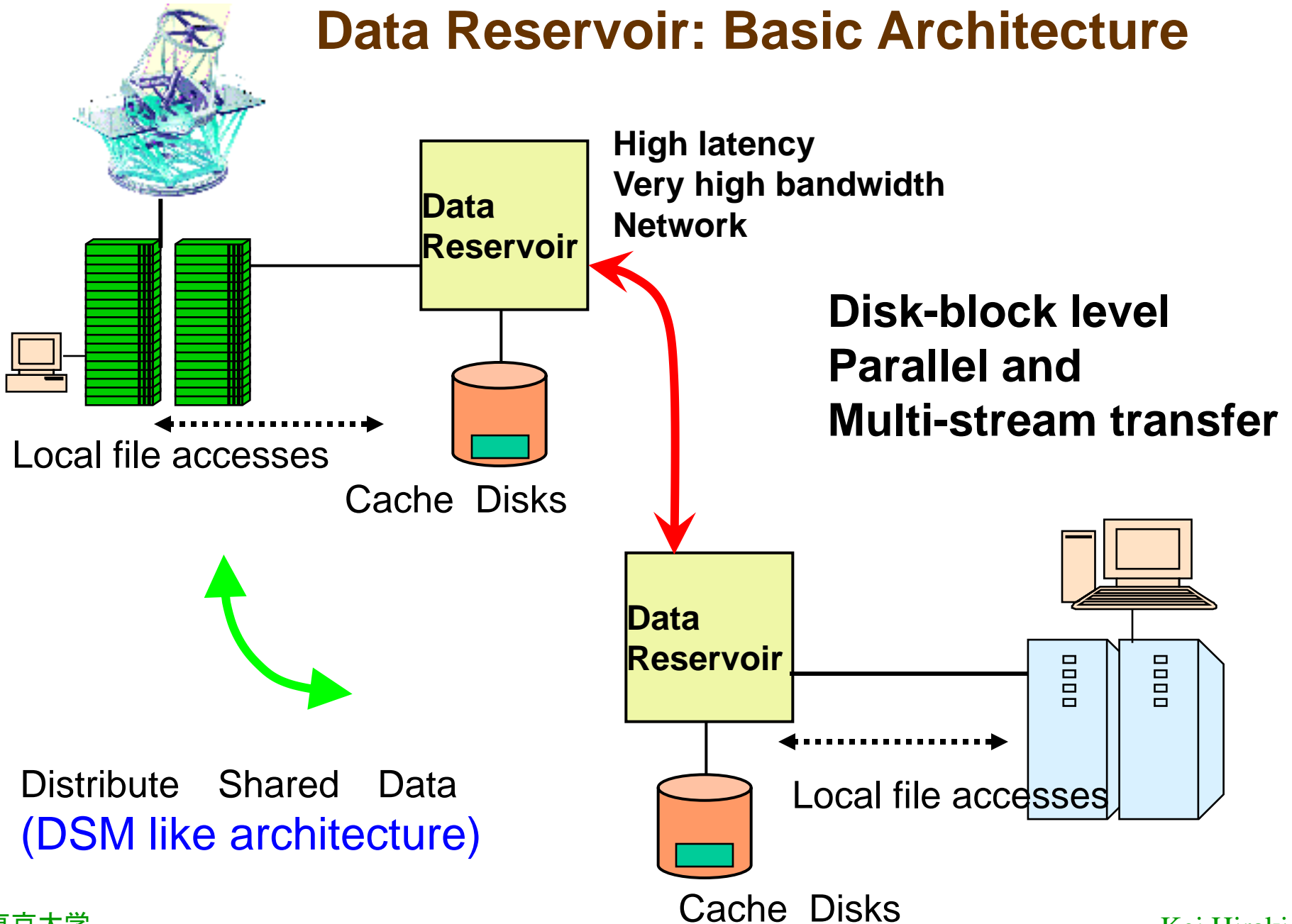
# Data intensive scientific computation through global networks



東京大学 Data analysis at University of Tokyo

Kei Hiraki

# Data Reservoir: Basic Architecture

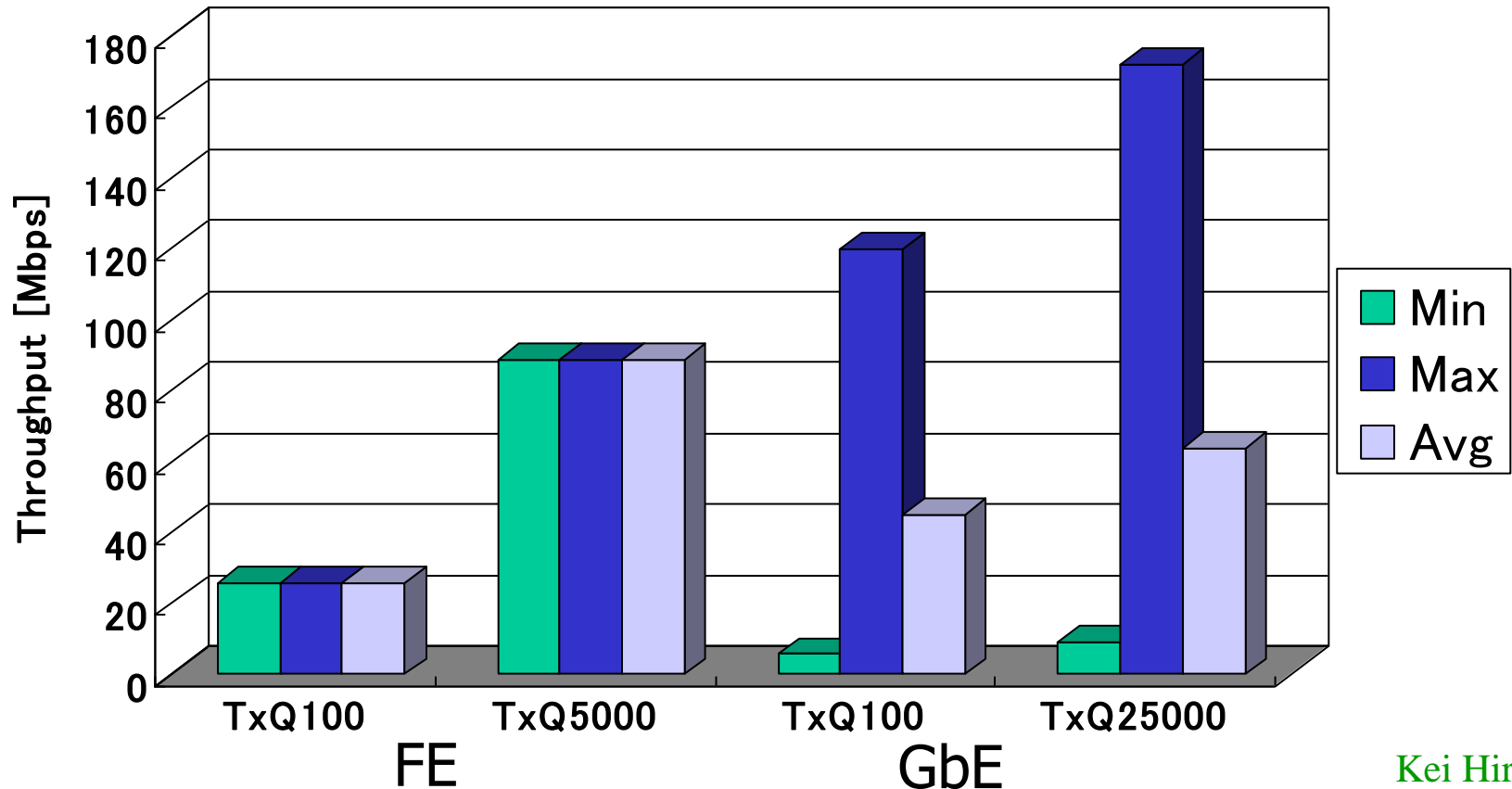


# Difficulties

- Artificial packet losses
  - Losses under Average BW < Maximum BW
- Bursty behavior of TCP traffic
  - Improvement in TCP or TCP-like protocol is not effective
  - Pacing works in some situation
- Too small buffer size at switches
  - Merging TCP streams cause artificial packet losses
  - Minimum buffer size  $> \frac{1}{4} \text{RTT} * \text{BW}$  is necessary

# Our starting point (1)

- **Fast Ethernet vs. GbE**
  - Iperf in 30 seconds
  - Min/Avg: Fast Ethernet > GbE



# Our starting point (2)

- **Delay emulator v.s. actual network**

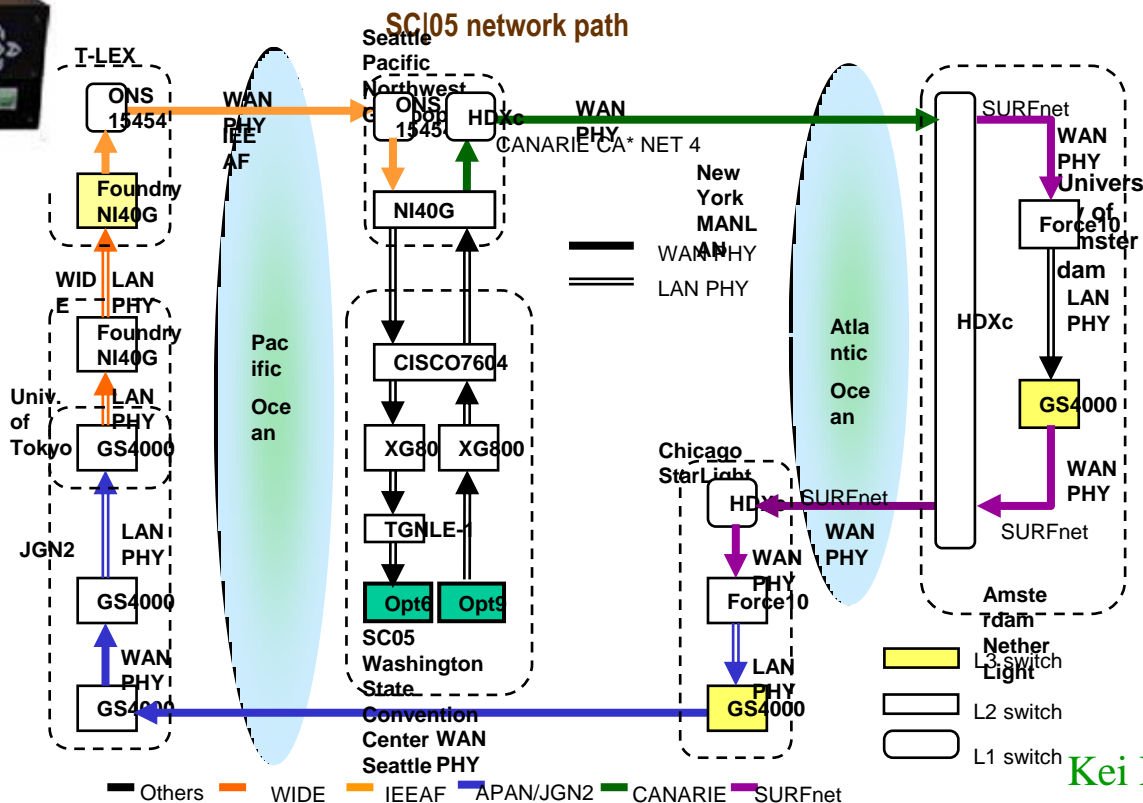
- Performance on a delay emulator is much better than actual network

Real network  
500ms RTT



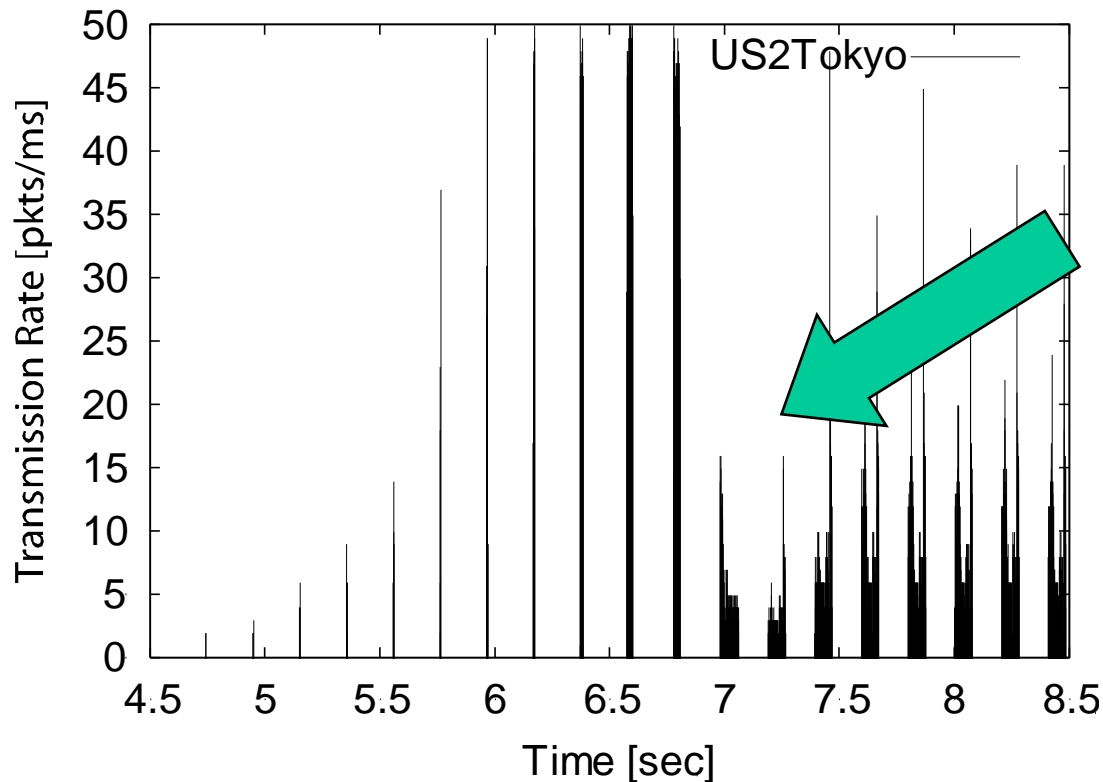
Delay emulator

Clock level accuracy



# Packet Transmission Rate

- Bursty behavior
  - Transmission in 20ms against RTT 200ms
  - Idle in rest 180ms

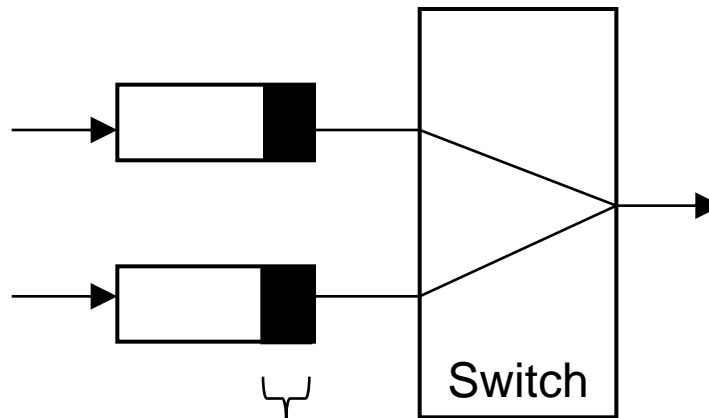
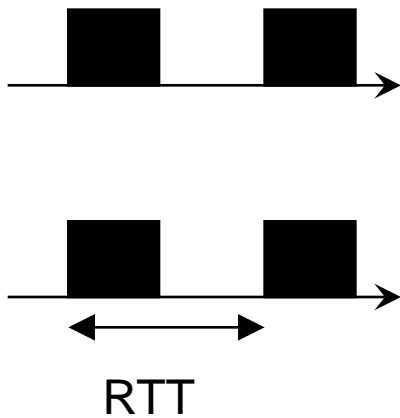


Packet loss occurred

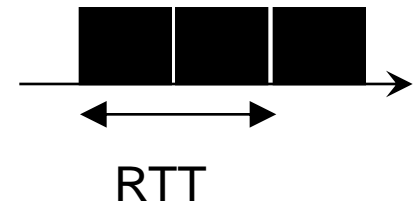
# Merging TCP streams

- Worst case is 2 TCP streams -> 1 TCP stream
- Bursty traffic just after slow start is the worst case

Traffic =  $\frac{1}{2}$  BW



Traffic = BW



Buffer size =  $\frac{1}{4}$  BW \* RTT \* Network Utilization

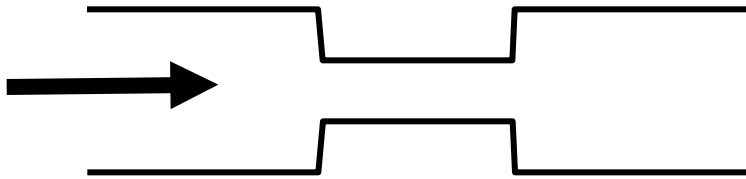
10 Gbps, 200msRTT  $\Rightarrow$  62.5 MB \* Utilization factor

Many switches do not have large buffer

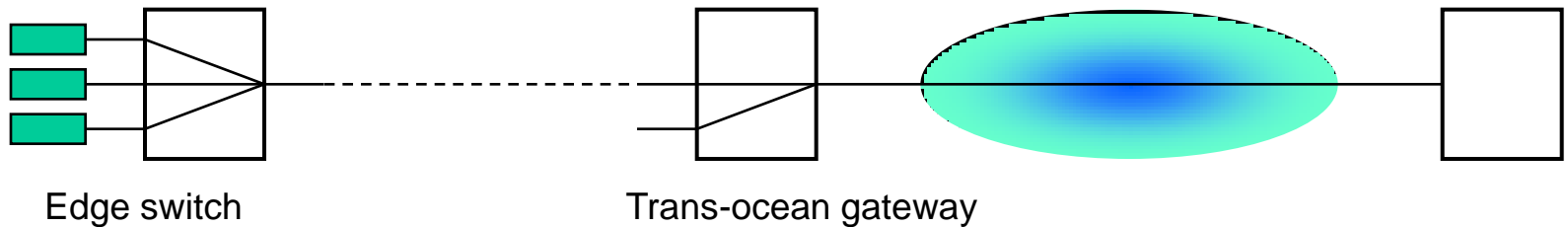
Flow control cannot apply at intermediate switches

# Sender side pacing

- BW bottleneck in the middle of network



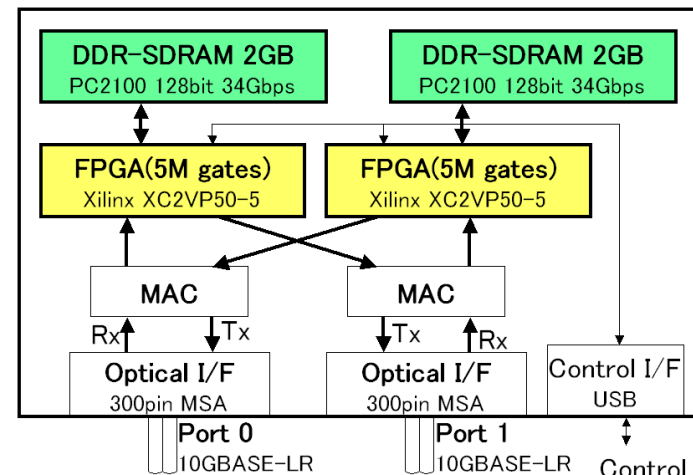
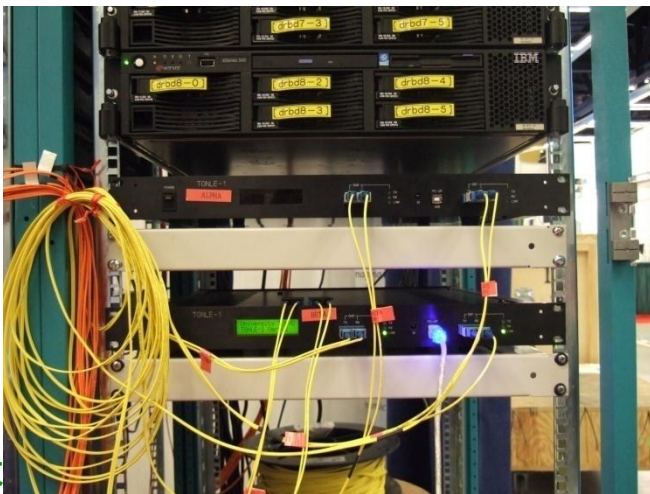
- Too small buffer size of edge switch and trans-ocean gateway switch



- Coordination between MAC and TCP layer (variable packet pace)
  - Avoid unnecessary packet loss at slow start phase

# Receiver side pacing

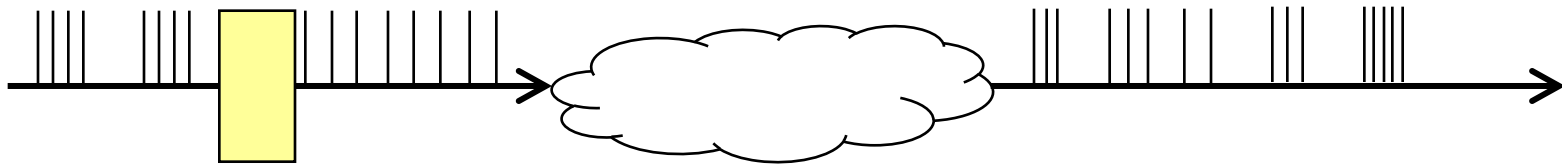
- Objective
  - Reduce packet losses due to receiver bottleneck
    - Act as large receiving buffer at receiving Network Interface Card
    - Set to the maximum receiving speed of the system
- Implementation
  - FPGA is used to implement fine-grain pacing mechanism
  - 2GB buffer memory (DDR memory)



# Pacing

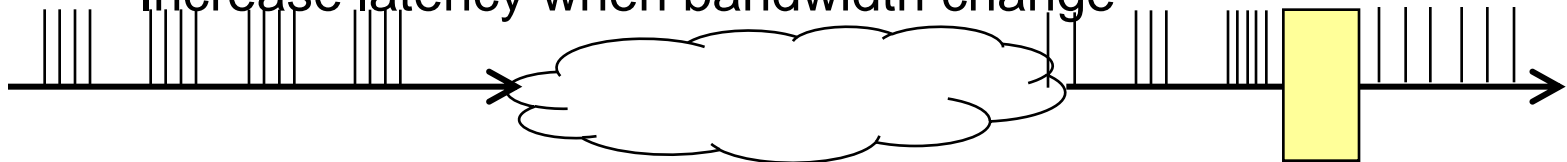
- Sender side pacing

- Effective for the bottleneck middle of the network
  - Complex hardware to cooperate with TCP protocol
  - Not effective to some 10G switches

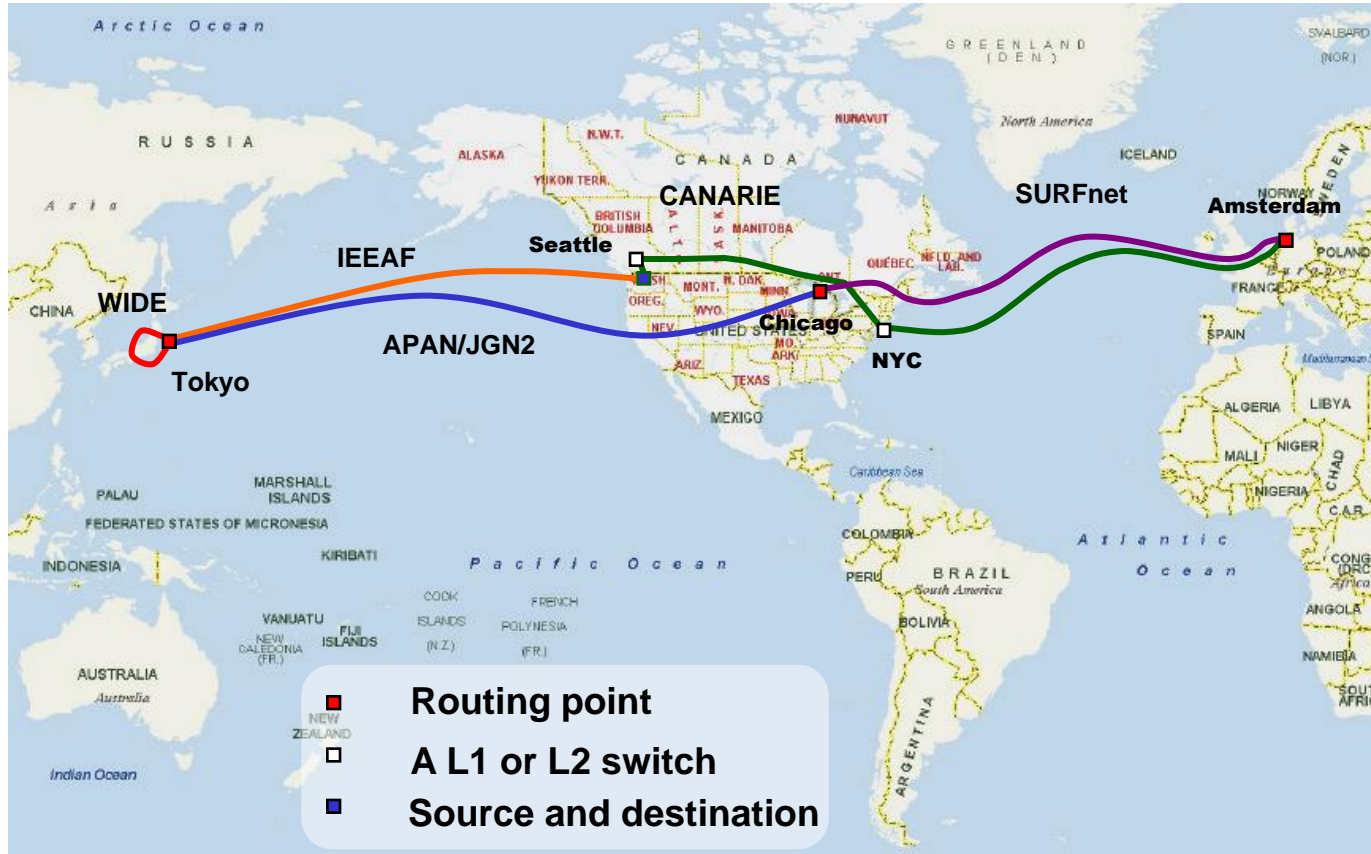


- Receiver side pacing

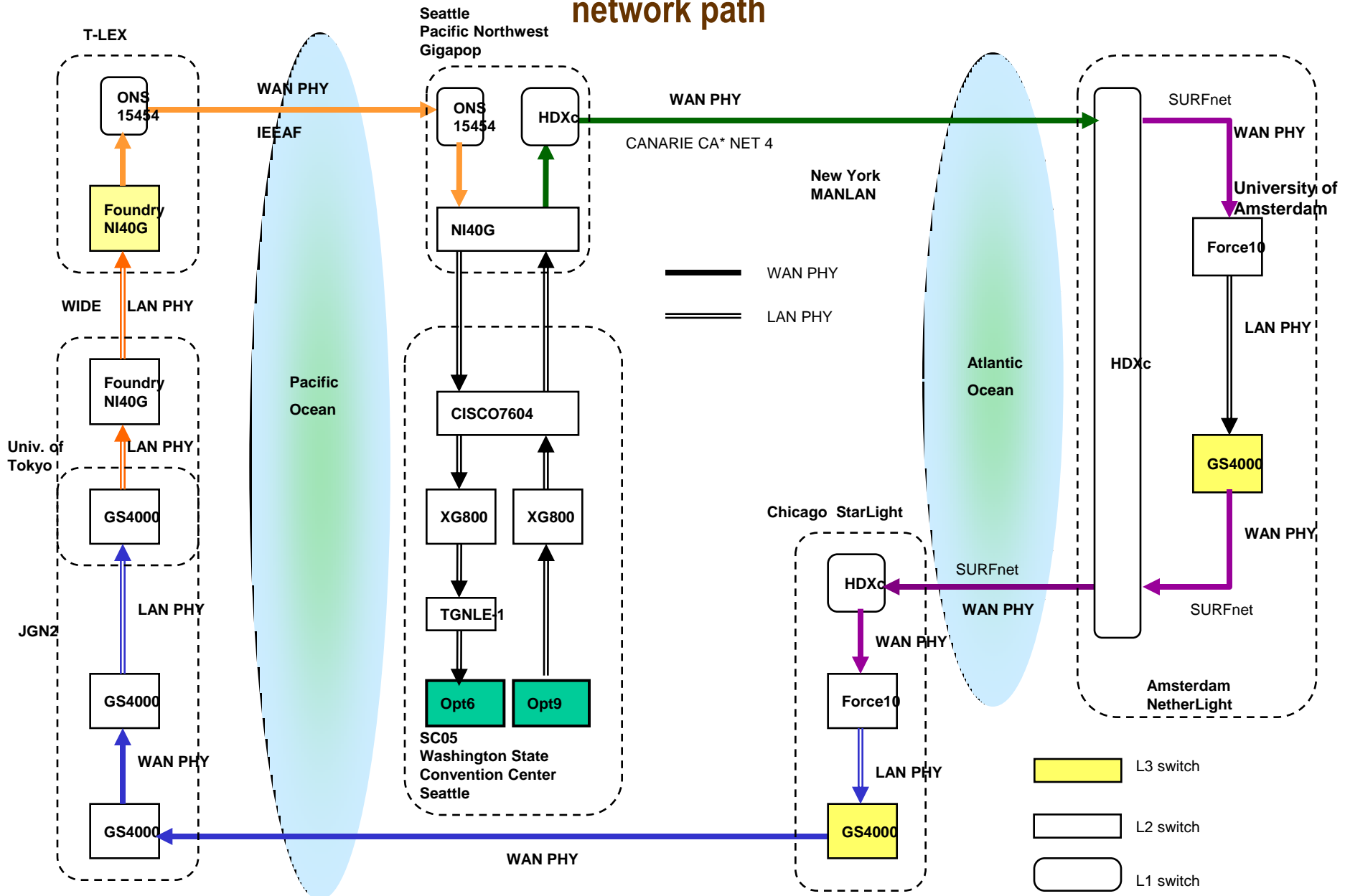
- Effective to NIC whose maximum bandwidth is less than 10G
- Simple hardware. It can be implemented in NIC
- Increase latency when bandwidth change



# SC|05 network path map

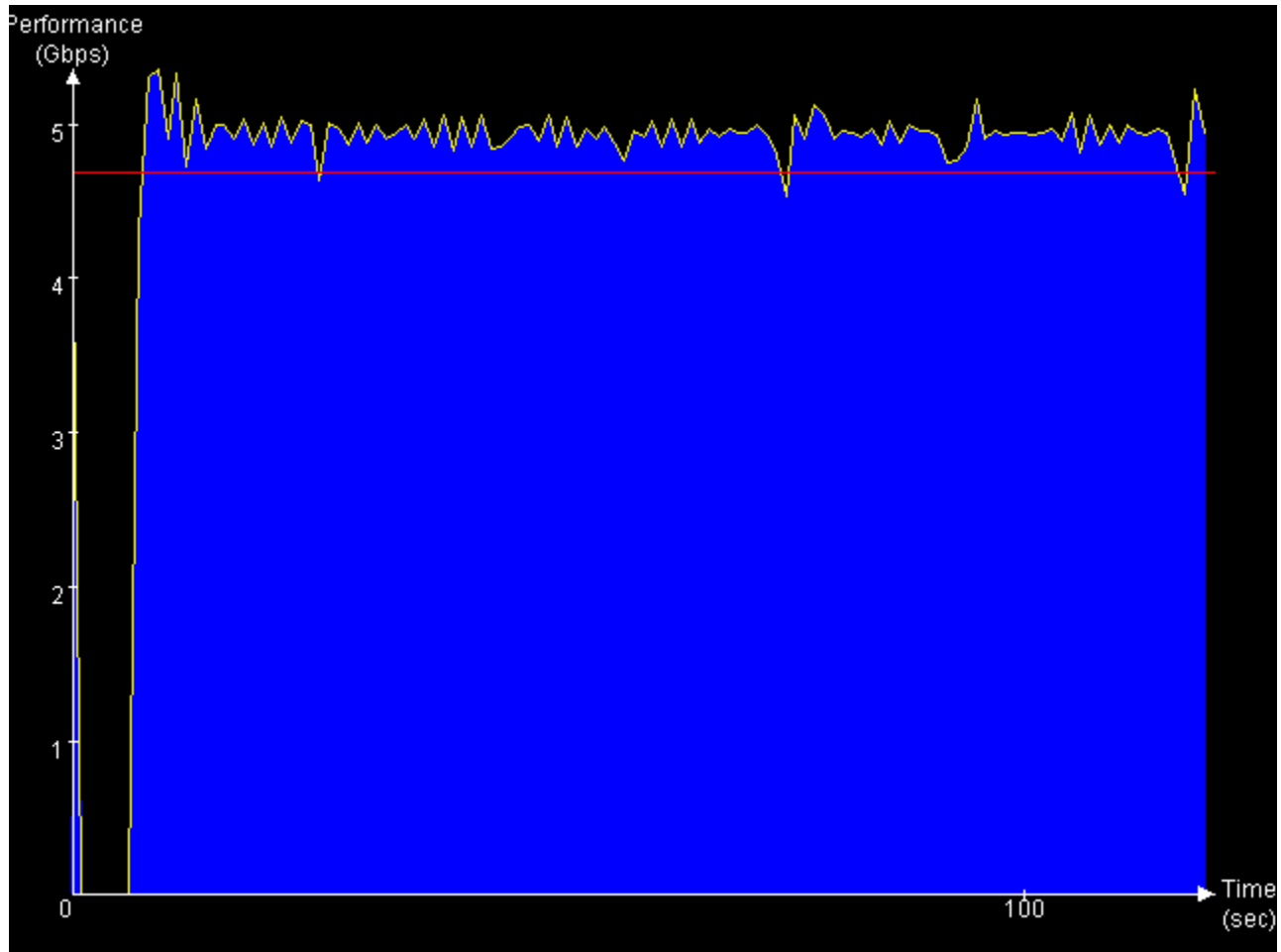


# network path



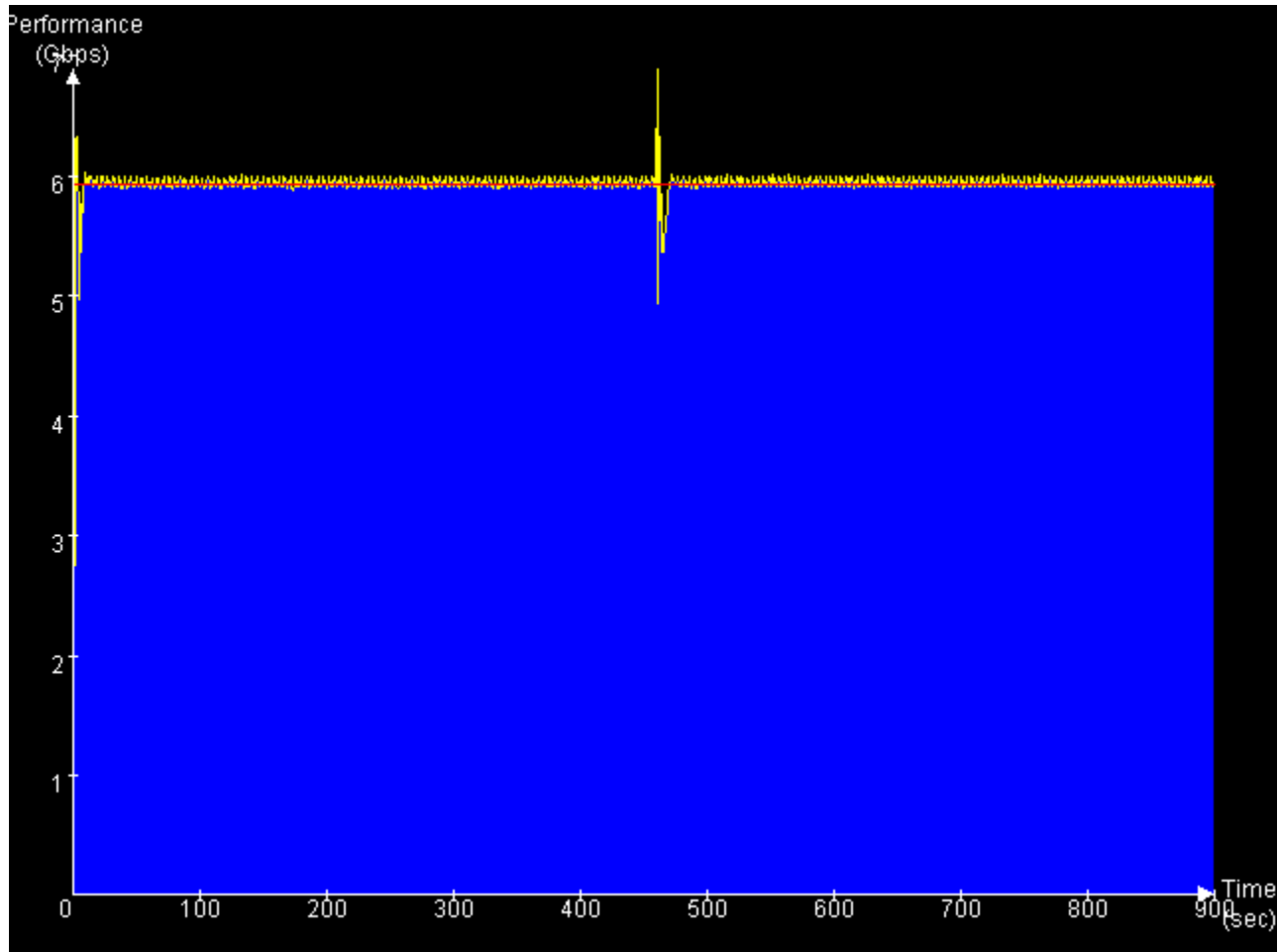
# Preliminary Results

- Without receiver side pacing



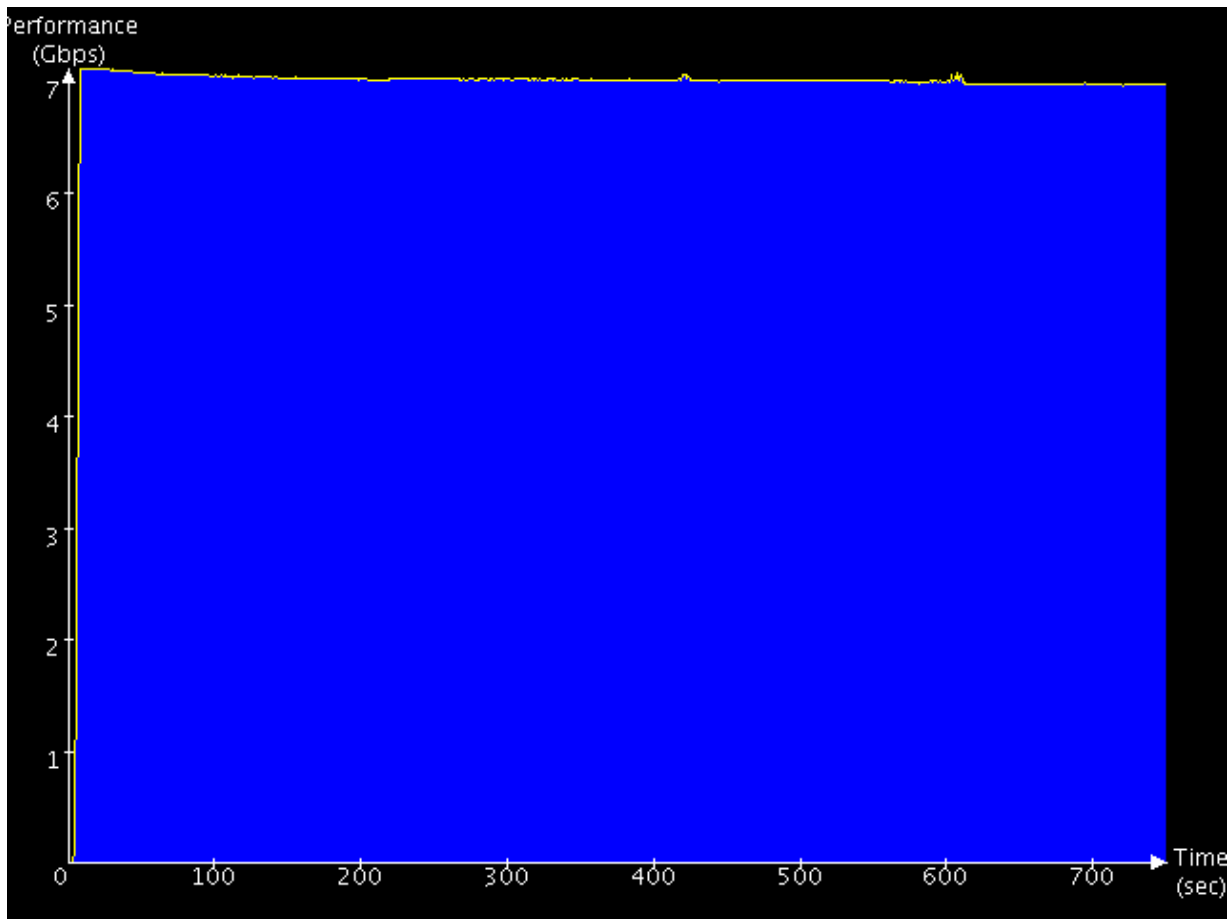
# Preliminary Results

- Without receiver side pacing

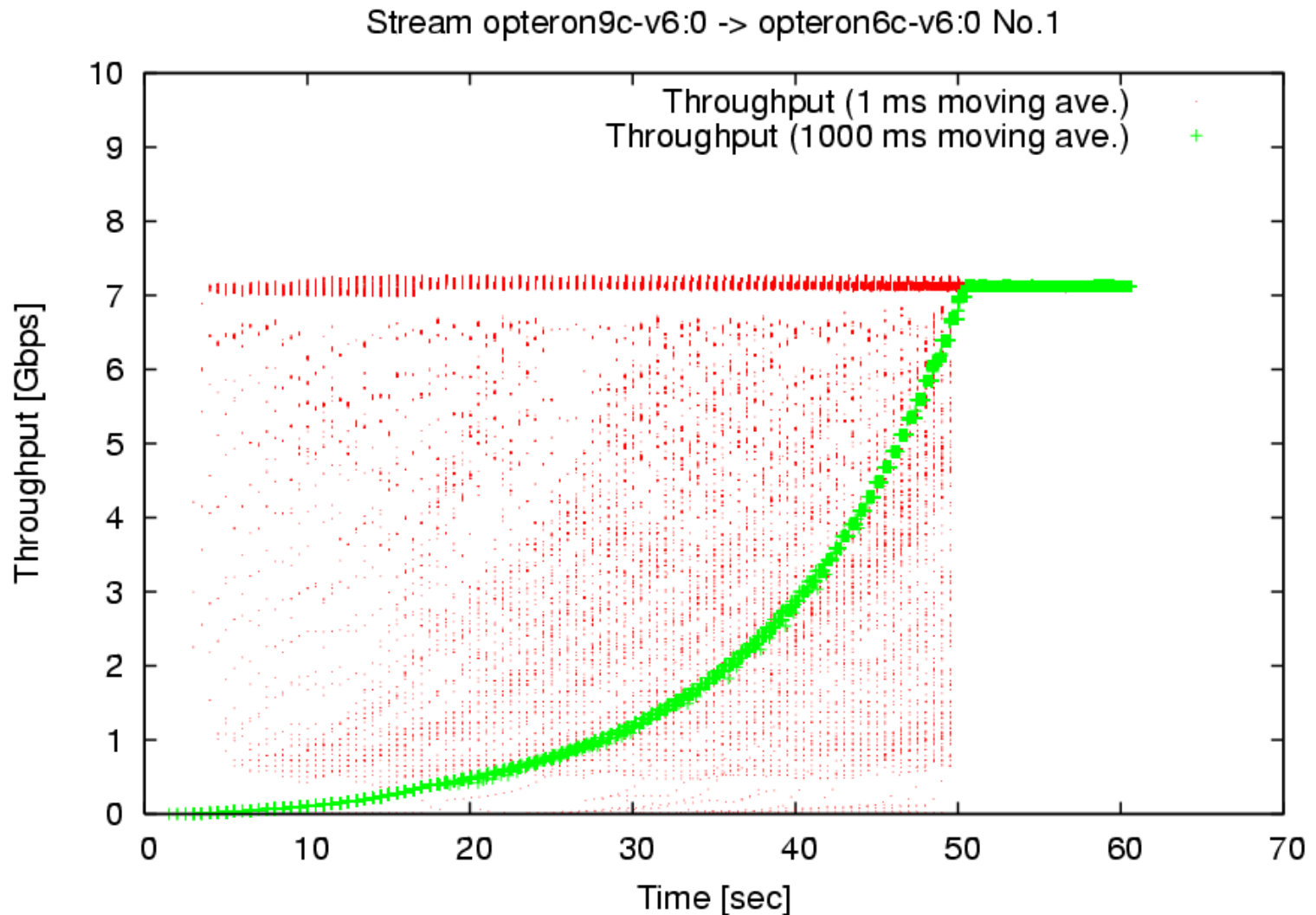


# Preliminary Results

- IPv6 TCP single stream
  - 6.96Gbps (use of “receiver side pacing”)
  - 5.58Gbps (without “receiver side pacing”)



# Linux 2.6.16 IPv6 Optron Performance



# Internet2 Land Speed Record

- (1) IPv4 single and multiple streams  
30000 Km, 8.8Gbps**
- (2) IPv6 single and multiple streams  
30000 Km, 9.06Gbps**

**Final records for 10Gbps Era.**



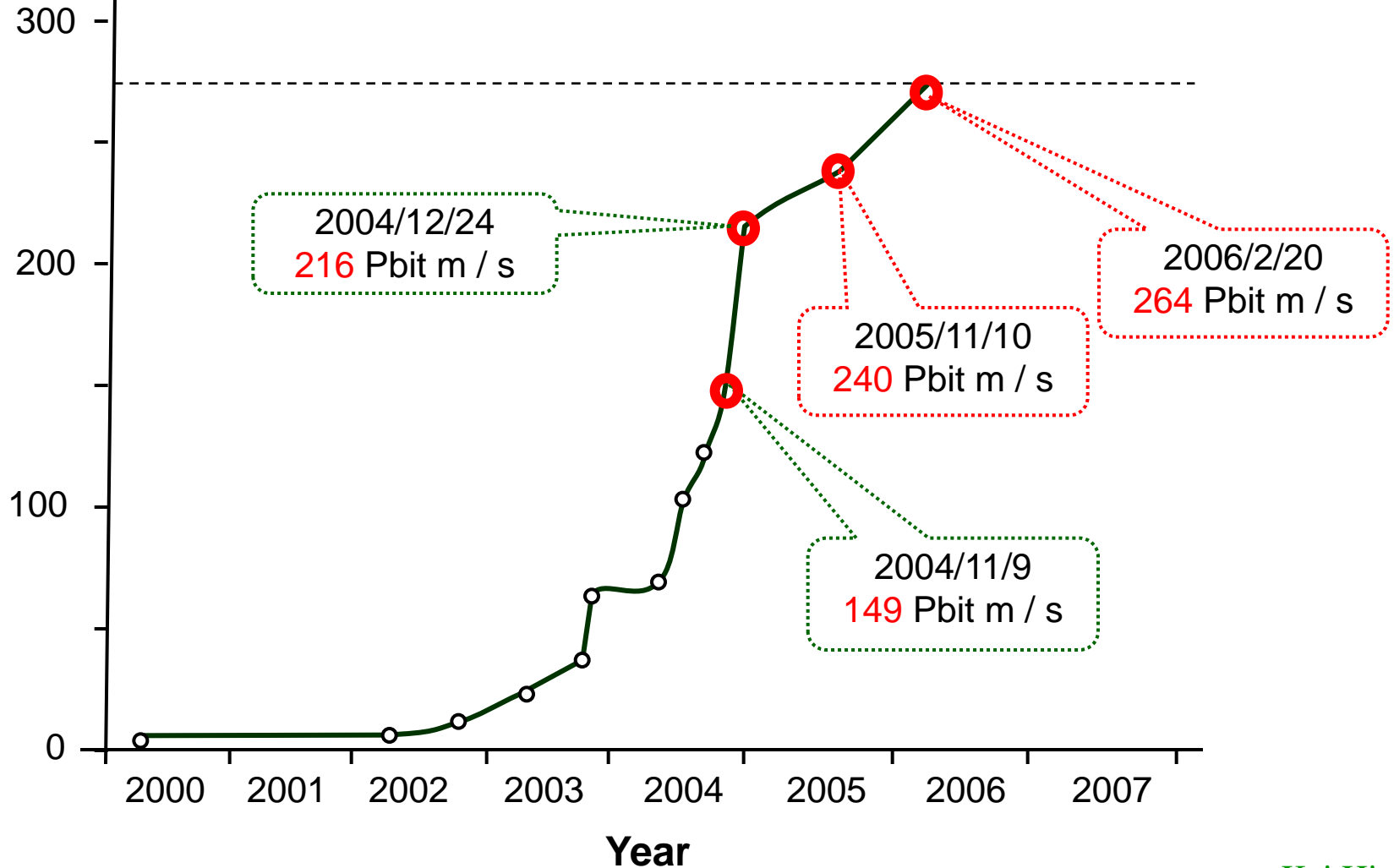
# History of IPv4 TCP speed records

Bandwidth Distance

products

Pbit m / s

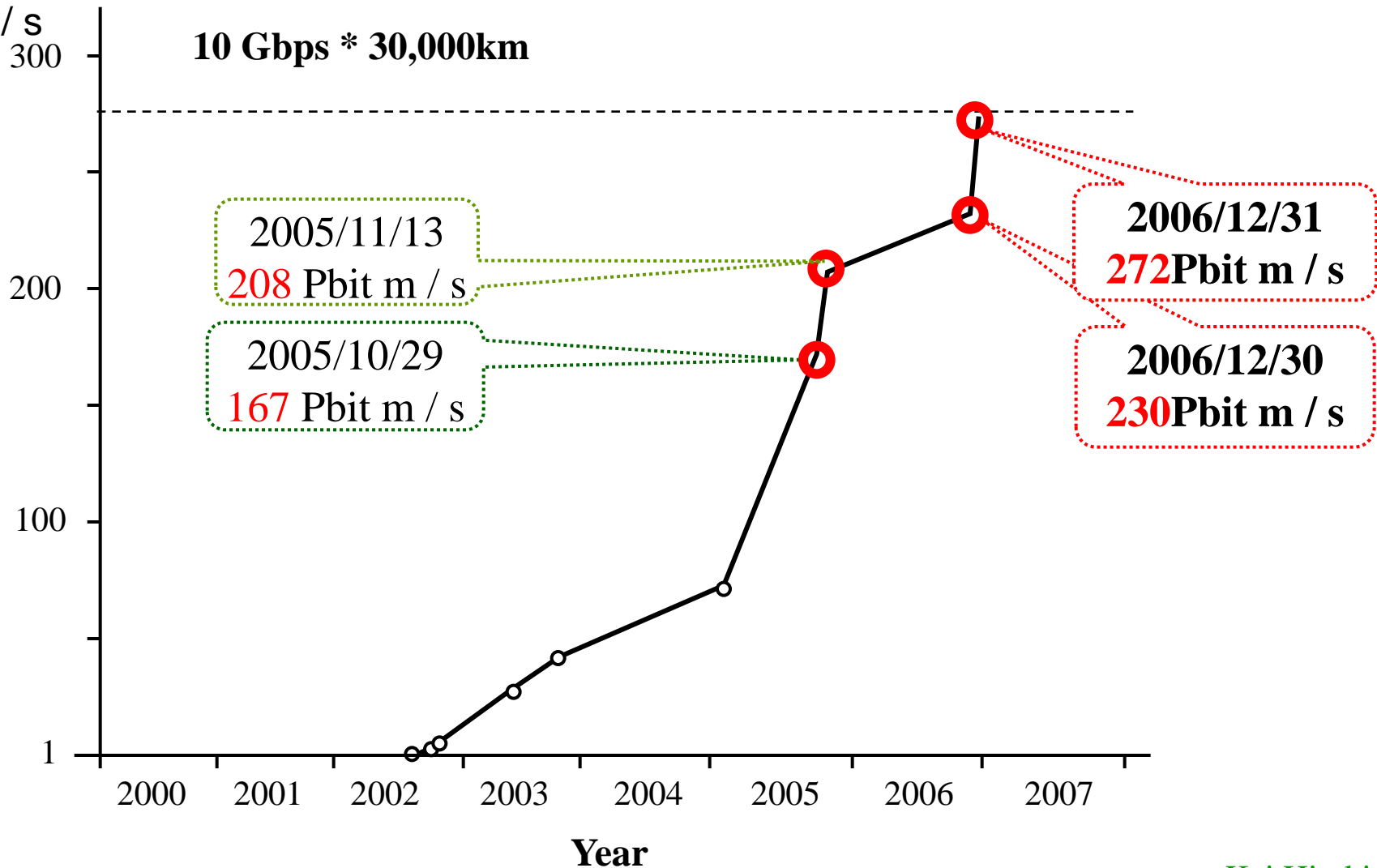
**10 Gbps \* 30,000km**



# History of IPv4 TCP speed records

Bandwidth Distance  
products

Pbit m / s



# Lessons learnt from experiments

- Difficult combination -- WAN PHY, IPv6, Jumbo frame, L3 switching
  - There is no trouble-free switch
  - Unexpected packet losses by many reason
  - Further investigation on “Flow Control” is essential
- Unnecessary packet losses by packet clustering
  - Max receiving speed is normally less than wire speed
  - Behavior is different from switch to switch
  - Receiving side pacing is quite useful.
- Unnecessary packet losses by bursty TCP traffic
  - Sending side pacing is effective if the number of intermediate switch is small
  - Intermediate switches and routers erase effect of pacing
  - Difference between WAN PHY and LAN PHY may make trouble

# Lessons learnt from experiments

- Use of wide-area L2 network
  - Spanning Tree algorithm may make unstableness
  - MAC address learning may cause packet losses
  - Difficulty in debugging
  - Switches for trans-ocean (trans-pacific, etc.) should have very large buffers and pacing capability
- 10G network interface
  - Pacing capability is essential
  - Large input buffer ( $2 \cdot \text{RTT} \cdot \text{BW}$ ) or receiving side pacing is useful
  - Proper setting of window size, buffer size and queue length is essential

# Conclusion

(1) We thank all the people who support our experiments

Next target is 9Gbps through WAN PHY network

(2) Current 10Gbps devices and software technology is still far from satisfaction

Buffer size, burstness control, large-scale L2 network

(3) Reasonable Pflops system can be constructed using GRAPE-DR processors

- Merging two technologies

- Possible cooperation with IBM