Multi-client LAN/WAN Performance Analysis of Ninf: a High-Performance Global Computing System

A. Takefusa, S. Matsuoka, H. Ogawa, H. Nakada, H. Takagi, M. Sato, S. Sekiguchi, U. Nagashima

(from Ochanomizu Univ., TIT, Univ. of Tokyo, ETL, NIT, RWCP)

http://ninf.etl.go.jp/

Towards Global Computing Infrastructure

Rapid increase in speed and availability of network

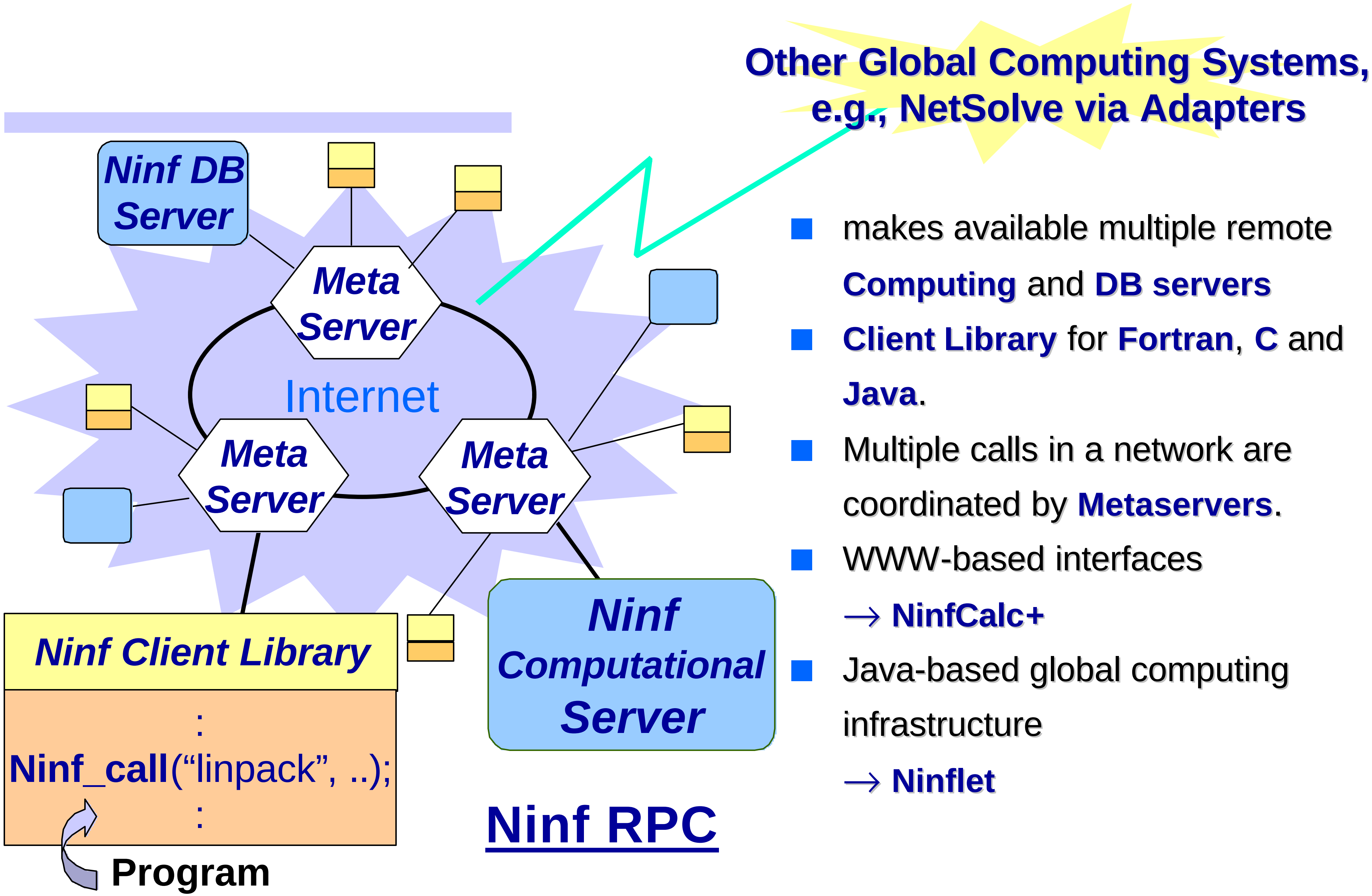
→ **Computational and Data Resources** are collectively employed to solve large-scale problems.

Global Computing (Metacomputing, The "Grid")

Ninf **(Network Infrastructure for Global Computing)**

c.f., ***NetSolve****,* ***Legion****,* ***RCS****,* ***Javelin****,* ***Globus*** *etc.*

Overview of Ninf



Ninf RPC

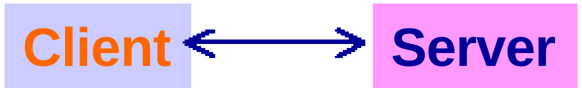
- makes available multiple remote **Computing** and **DB servers**
- **Client Library** for **Fortran**, **C** and **Java**.
- Multiple calls in a network are coordinated by **Metaservers**.
- WWW-based interfaces
  → **NinfCalc+**
- Java-based global computing infrastructure
  → **Ninflet**

Issues of Global Computing systems

- **Communication Performance** (throughput, latency)
  - High latency → deterrent to performance?
- **Computing Server Selection**
  - Criteria to select servers?
  - Can supercomputers/MPPs be effectively shared by multiple remote clients? (Scheduling characteristics)
- **Sharing by Multiple clients**
  - Global computing servers on supercomputer/MPP OSes effectively handle multiple parallel tasks?
- **Remote Library Design and Reuse**
  - Allocate processors in a task-parallel / data-parallel manner ?
  - The right “remote library” design, decomposition, API, etc.

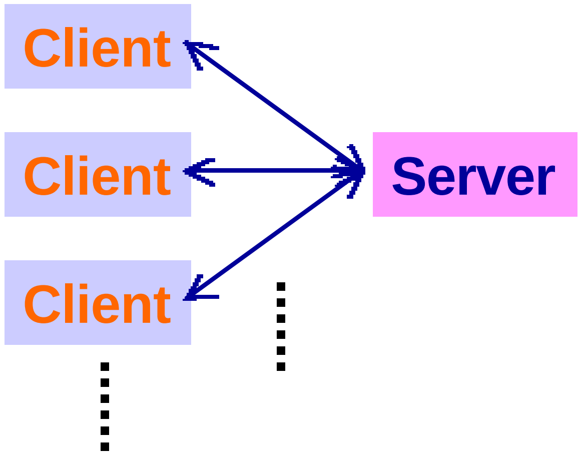
Outline of Benchmarks

- **Single-Client Benchmarks :**



  - **Ninf Baseline Performance** → **LAN, Linpack**

- **Multi-Client Benchmarks :**



  - **Communication Performance**

    → **LAN / WAN (Single-site / Multi-site)**
  - **Remote Library Design and Reuse**

    → **Task Parallel / Data Parallel**
  - **Robustness of Computational Server** (Cray J90)
  - **Ratio of Computation and Communication of Library**

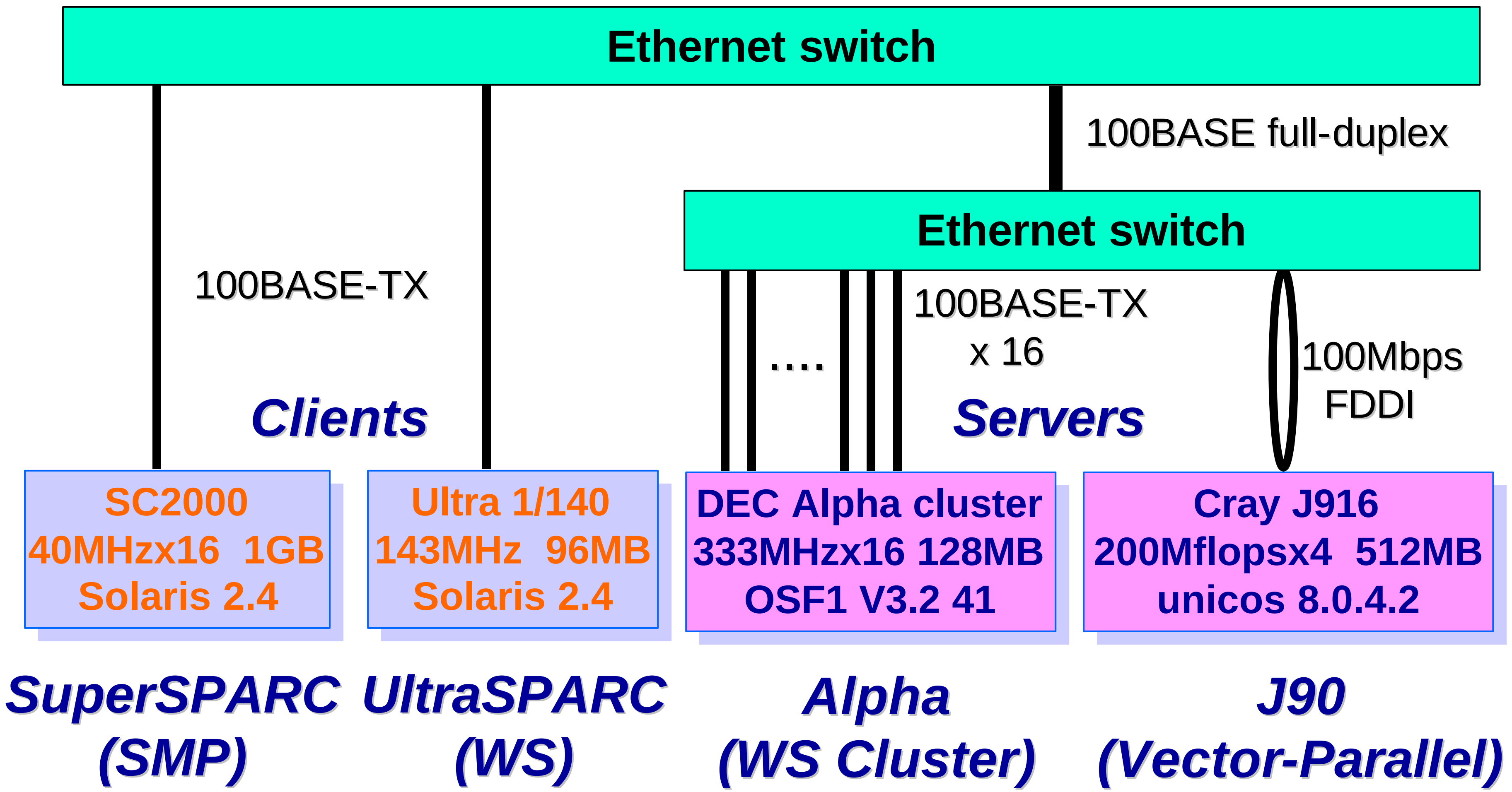
    → **Linpack / EP**

Benchmarks

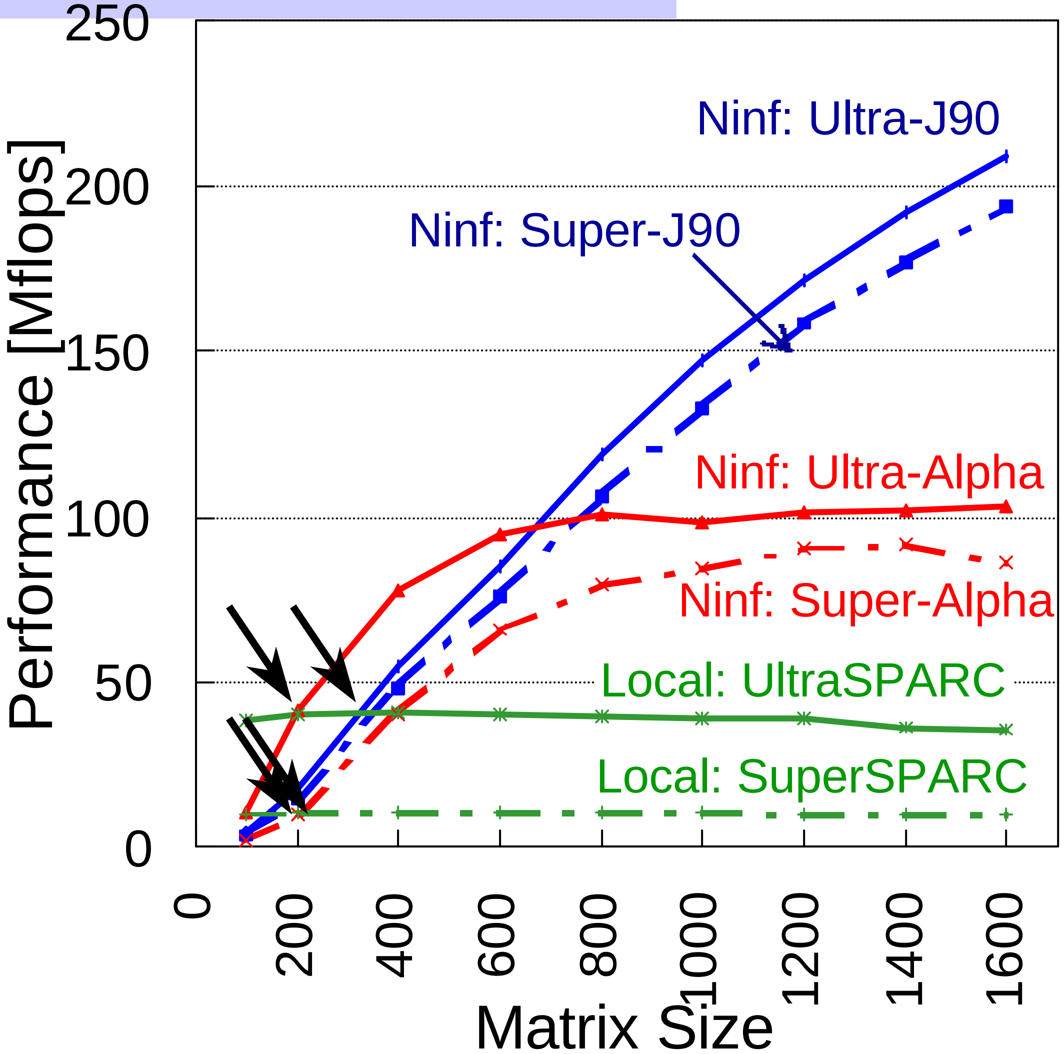
- **Linpack** : Gaussian Elimination
  - Computation: $2/3n^3 + 2n^2$ [flops]
  - Communication: $8n^2 + 20n + O(1)$ [bytes]
  - Performance: $(2/3n^3 + 2n^2)$ / **Elapsed Time** [flops]
- **EP (NASPB Kernel)** : Random Number Generation
  - Computation: 2^{n+1} [ops]
  - Communication: $O(1)$ [bytes]
  - Performance: 2^{n+1} / **Elapsed Time** [ops]

Elapsed Time = Communication + Computation

LAN Single-client Benchmarking Environment (at ETL)



LAN Single Client Linpack Results



- Ninf is faster than Local at *n* = 150〜300
- For Ninf_call to J90,

  Ninf performance is not saturated.

  (J90's Local achieves 600Mflops when *n*=1600)

  → Ninf performance quickly overtakes Local.
- The effects of client machine's performance difference are small.

Multi-client Benchmarks (LAN, WAN)

- **A Model Client Program**

  Linpack and **EP** are both repeatedly called:

  - Each client performs a Ninf_call on the interval of s seconds with probability p. $\rightarrow$ s = 3, p = 1/2 chosen.
  - Number of clients : c , problem size : n.

    $\rightarrow$ c = 1, 2, 4, 8, 16, **Linpack:** n = 600, 1000, 1400

    EP: n = 24(Sample)

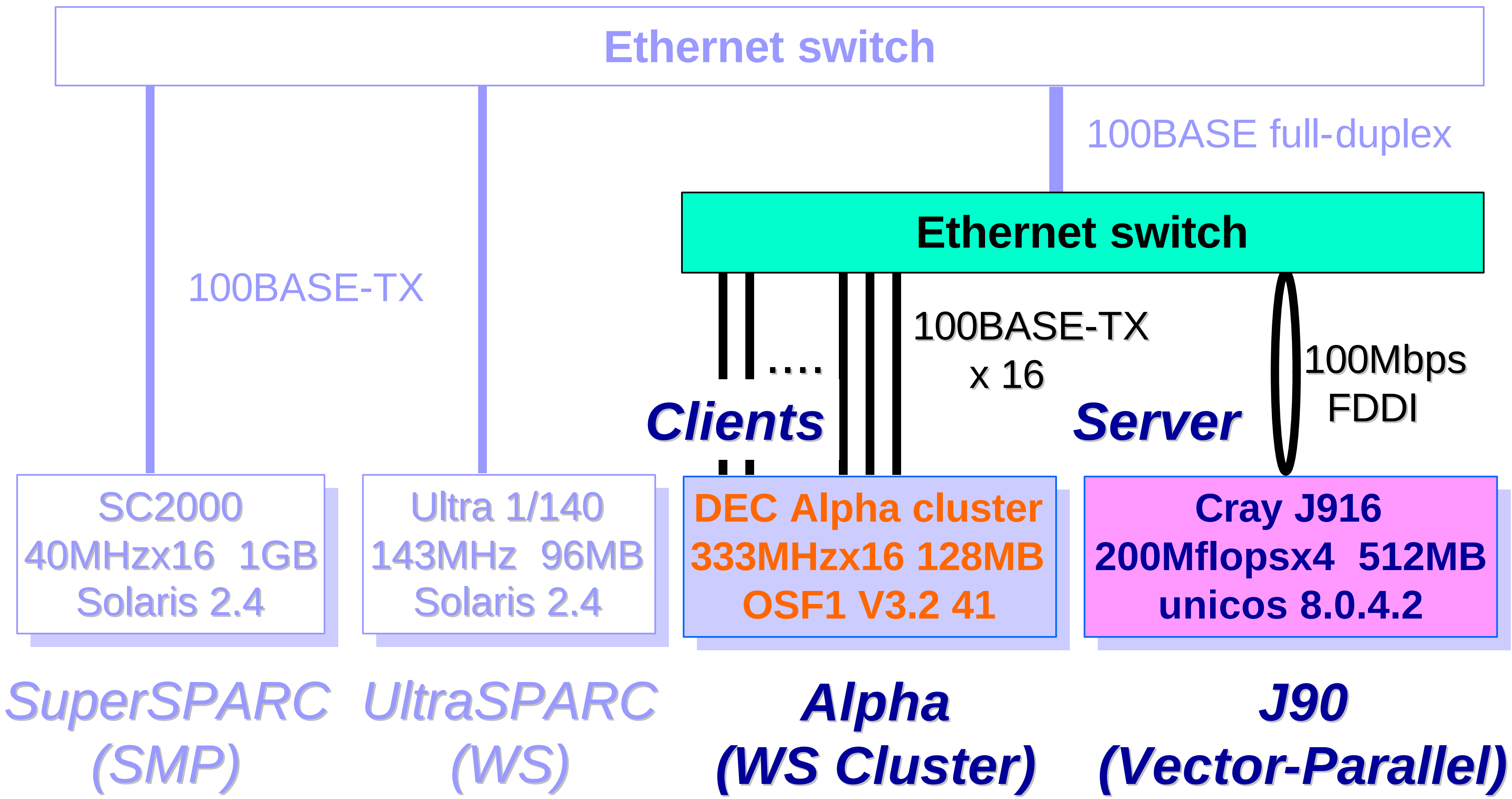
- **Parallel Processing on the server (J90 4PE)**
  - **Linpack**: **1PE ver.** --- Task Parallel

    1PE Execution and Parallel Processing

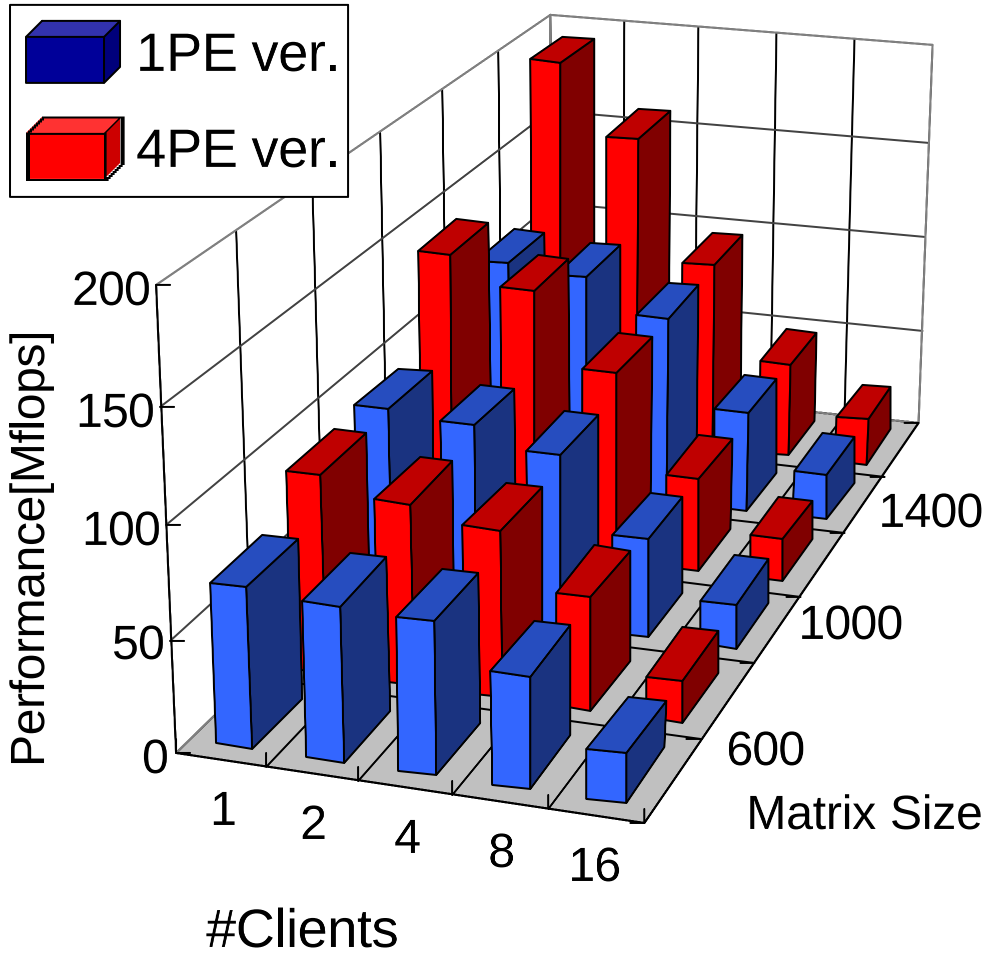
    4PE ver. --- Data Parallel

    4PE Execution and Single Processing
  - **EP**: **1PE ver.** --- Task Parallel

LAN Multi-client Benchmarking Environment (at ETL)

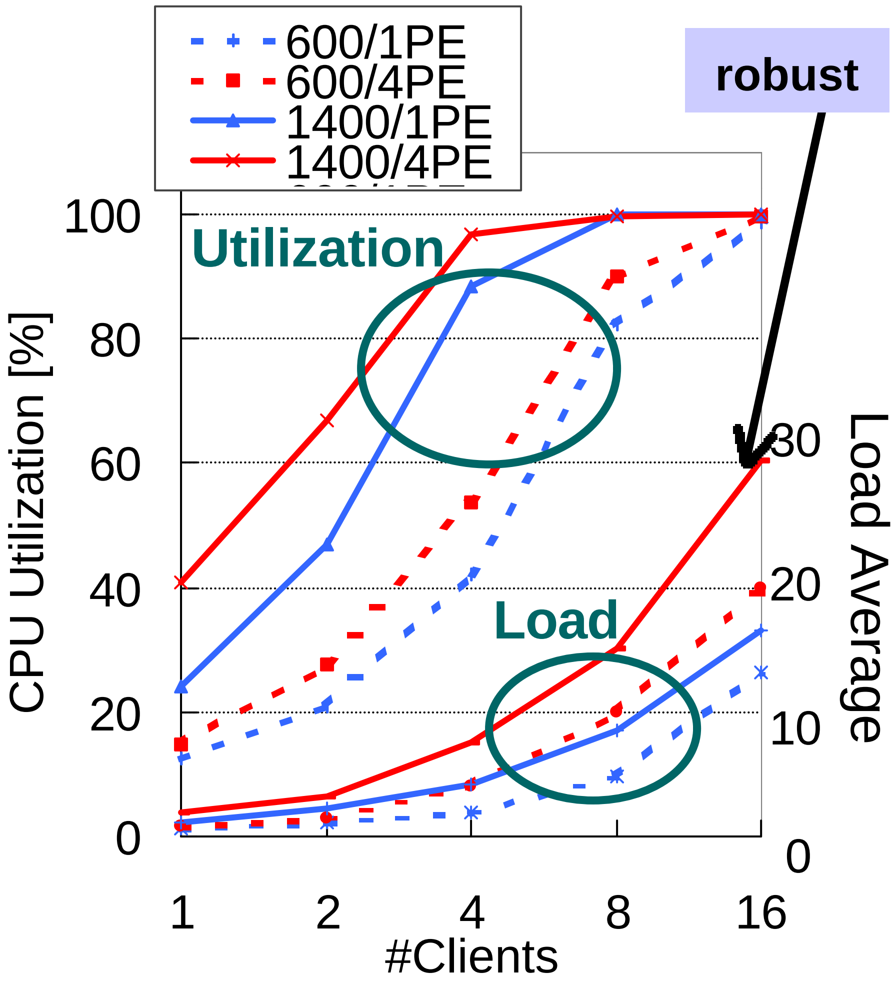


LAN Linpack Benchmark Results
Average Performance



- 4PE ver. exhibits higher performance for a small c,
- while there is little performance edge for the 1PE ver. for a larger c.
  - Numerical core speed
  - Small overhead of switching parallel tasks on J90
  - Overlapped communication
- Average response and waiting time didn't differ depending on n, c, or 1PE / 4PE ver.

LAN Linpack Benchmark Results
CPU Utilization and Load Average



- 4PE ver. exhibits higher Utilization and Load.

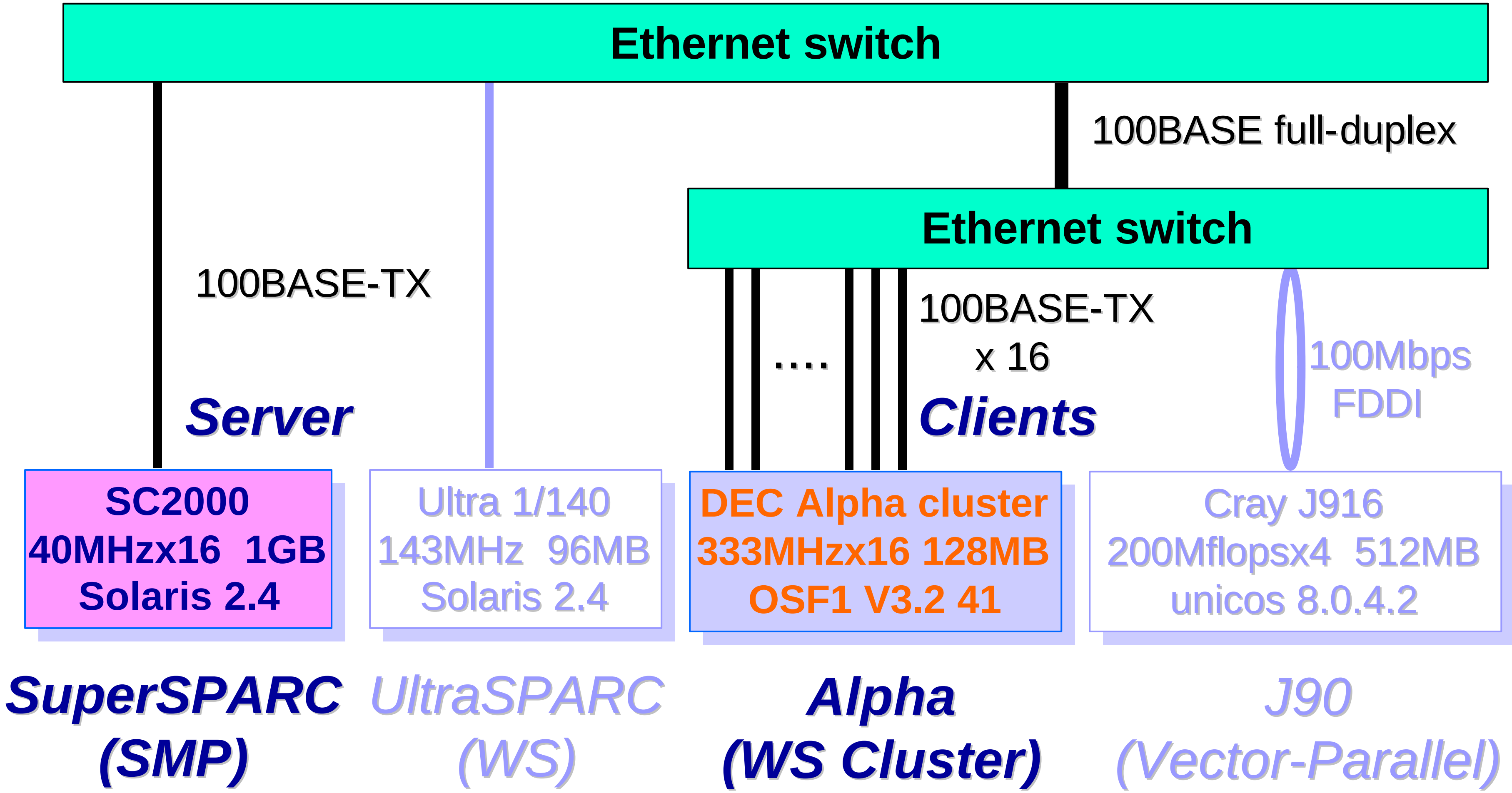
→ The optimized parallel library would be appropriate for J90.

- Utilization and Load are higher as n and c increase.
- The J90 Ninf server continued to work flawlessly.

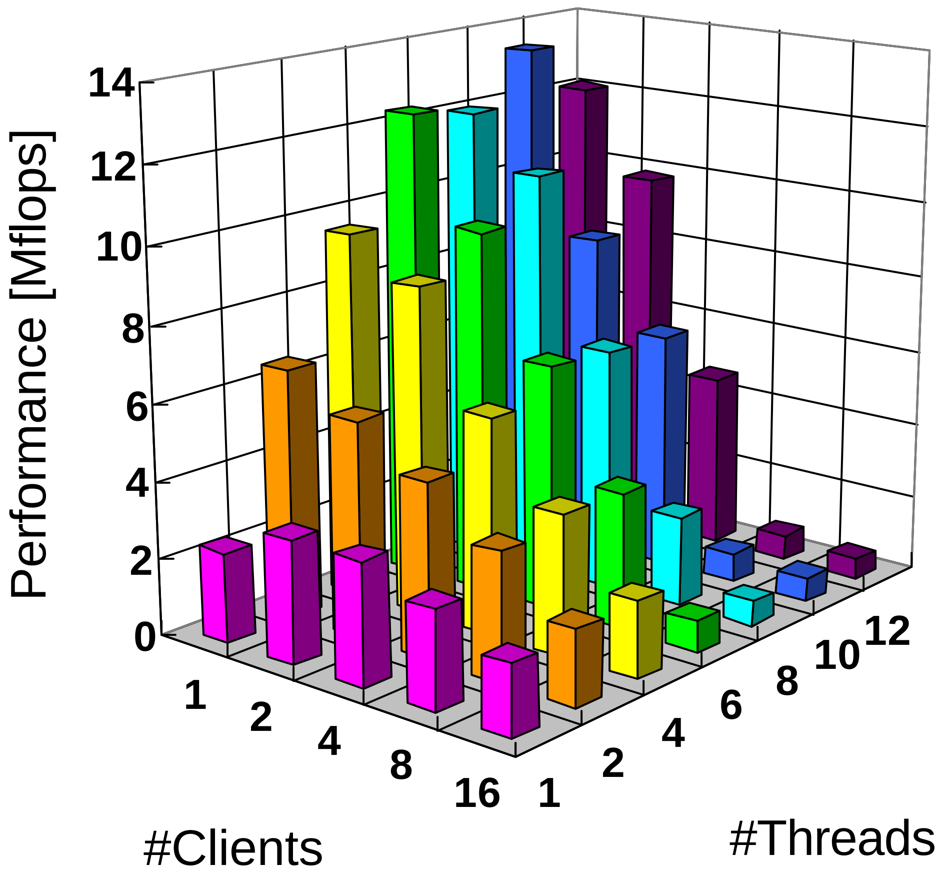
(even for n=1400, 4PE ver.,

max. load average : 30)

LAN SMP Multi-client Benchmarking Environment (at ETL)



Average Performance of Multi-thread LAN Linpack on SPARC SMP



- Highly multi-threaded ver. exhibit notable slowdown.
  - Solaris on SPARC SMP is optimized for handling the requests.
  - do not co-schedule multiple threads.

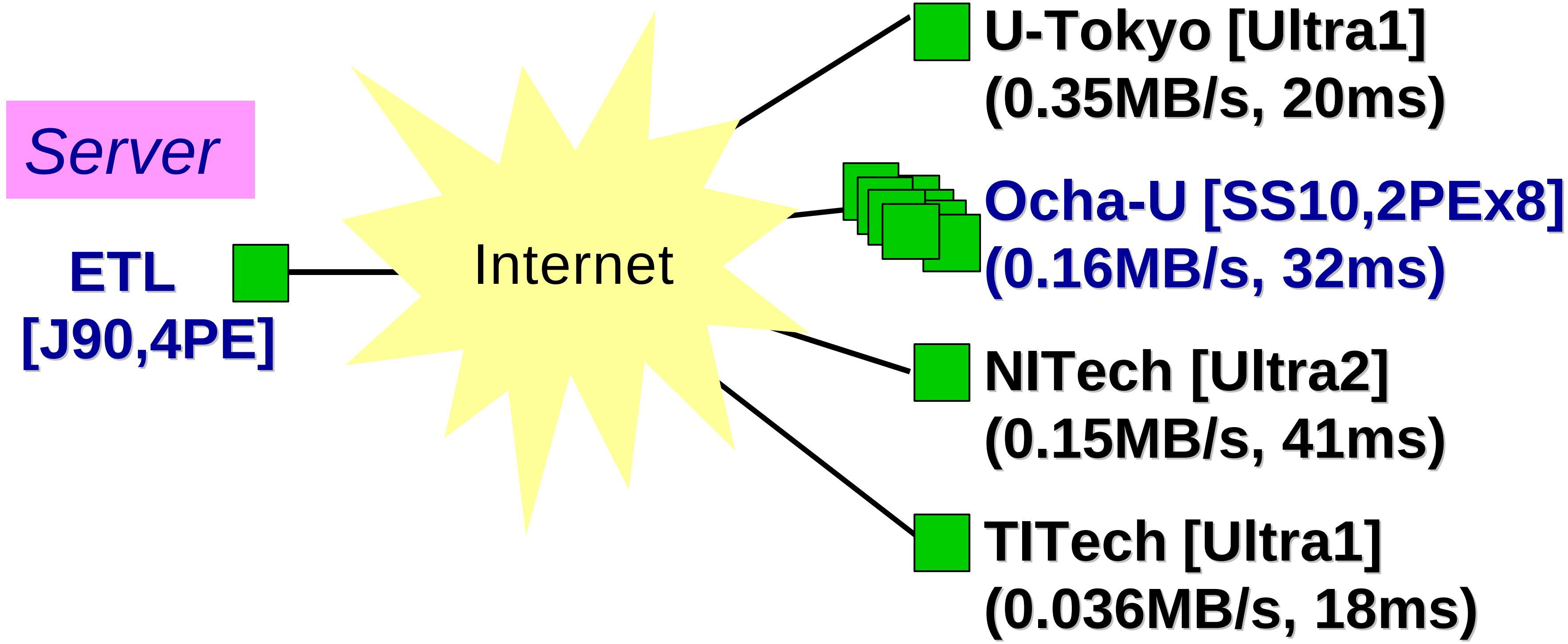
→ thread-switching overhead.

(cache / TLB misses etc.)

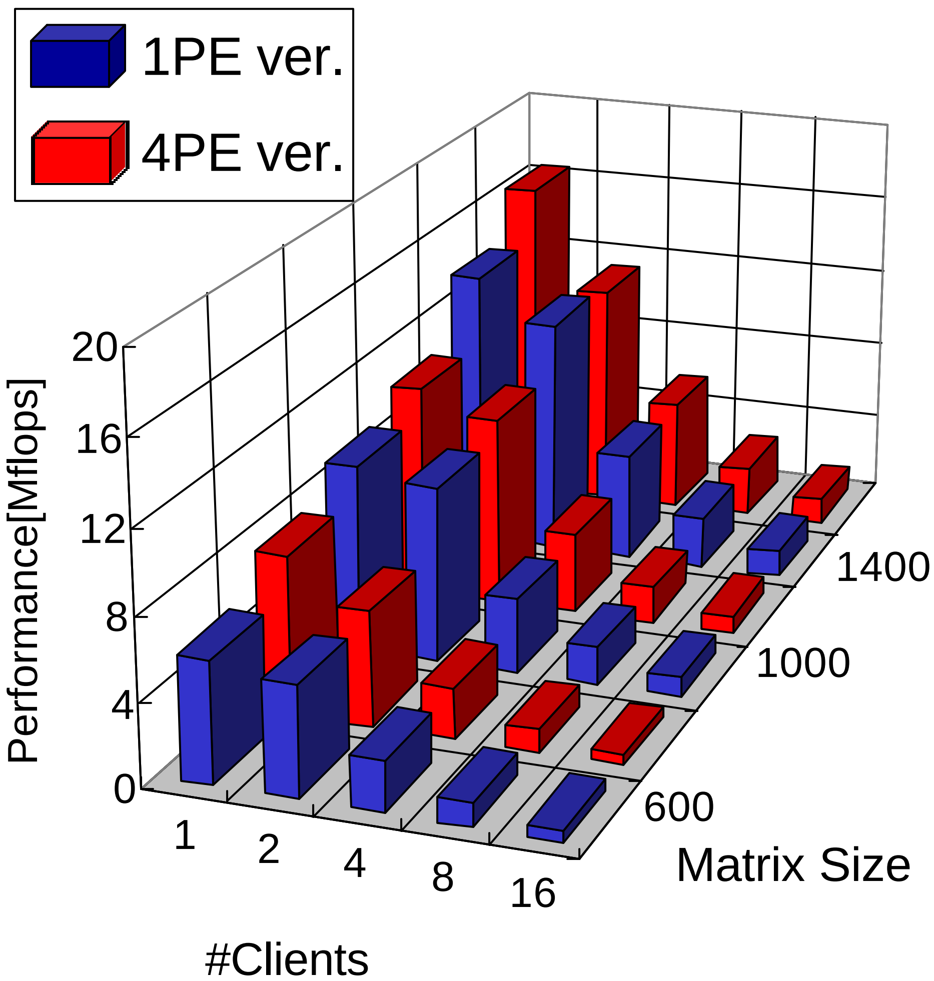
WAN Multi-client Benchmarking Environment

- Single-Site
- Multi-Site

Clients



Single-Site WAN Linpack Benchmark Results
Average Performance



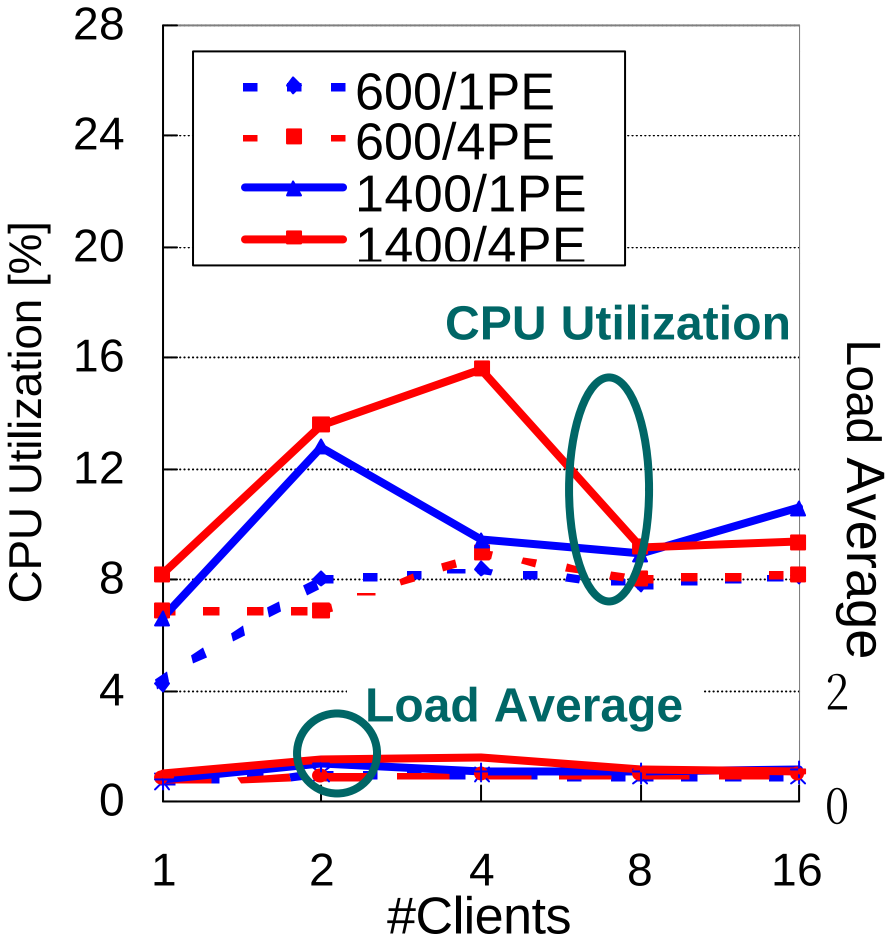
- Ave. performance deteriorates as c increases.

→ The network throughput saturates even for c=1 or 2.

- Exhibited almost the same characteristics as LAN.

→ Better to use the parallel library versions for WAN clients as well for J90.

Single-Site WAN Linpack Benchmark Results
CPU Utilization and Load Average



- Utilization and Load remain low even for c=16.

→ It is difficult for the global computing servers to process numerous requests.

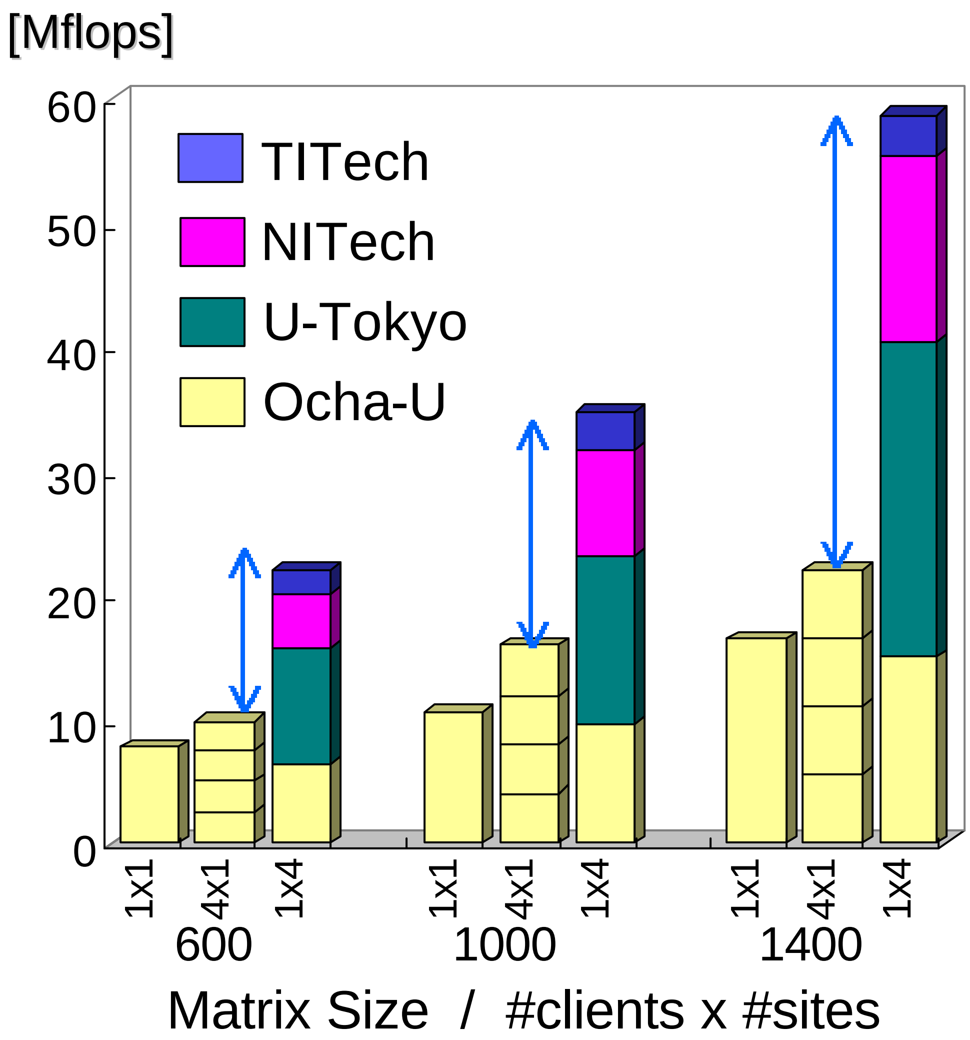
(for communication intensive tasks)

→ A good load balancing algorithm is needed

- server load
- network traffic and topology
- client location etc.

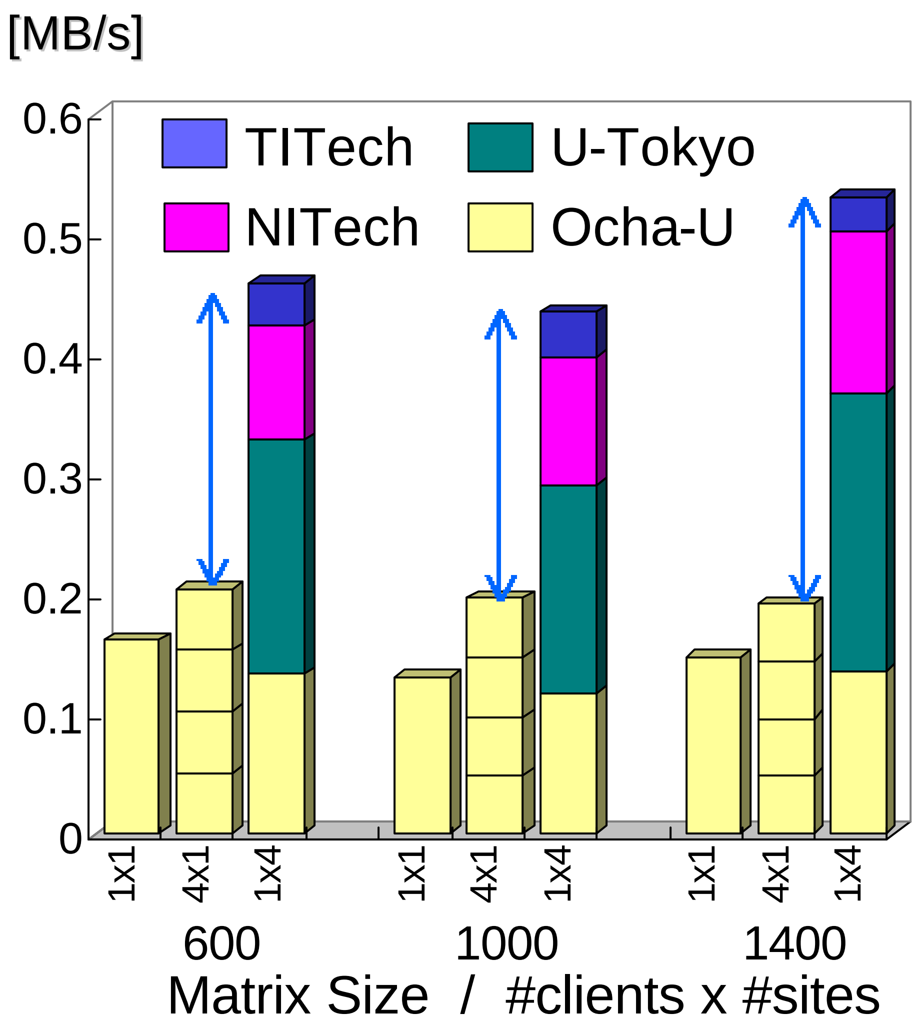
Single/Multi-site WAN Linpack Benchmark Results

Average Performance (c = 4, 4PE ver.)



- Total performance is higher as n increases.
- Aggregate performance from multi-site is substantially higher than from single-site for the same c.

Single/Multi-site WAN Linpack Benchmark Results
Communication Throughput (c = 4, 4PE ver.)

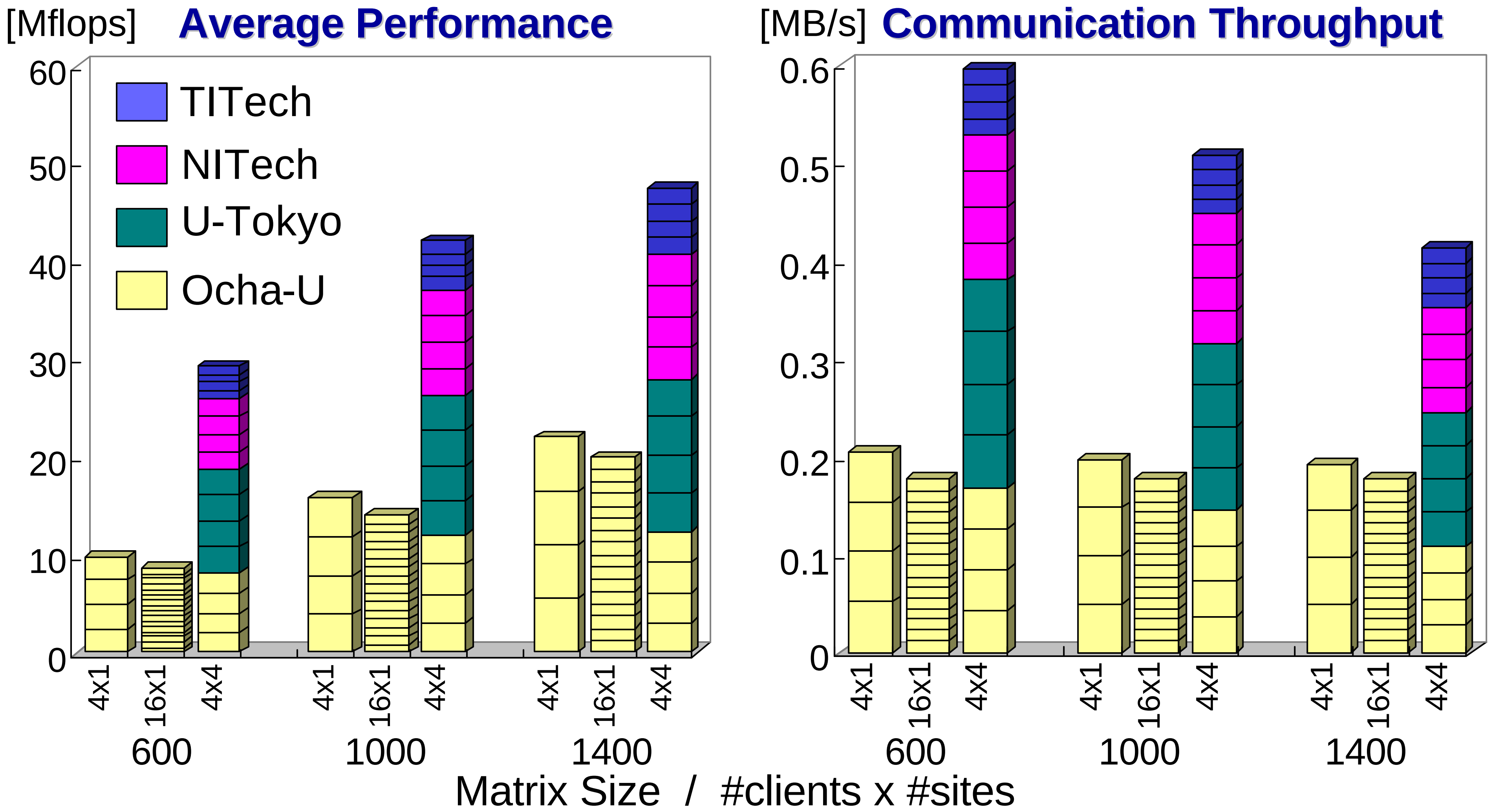


- Total communication throughput is nearly constant for each n.
- Aggregate communication throughput from multi-site is substantially higher than from single-site.

→ **For communication intensive tasks, *point-to-point bandwidth* is the dominant factor in performance. (not latency)**

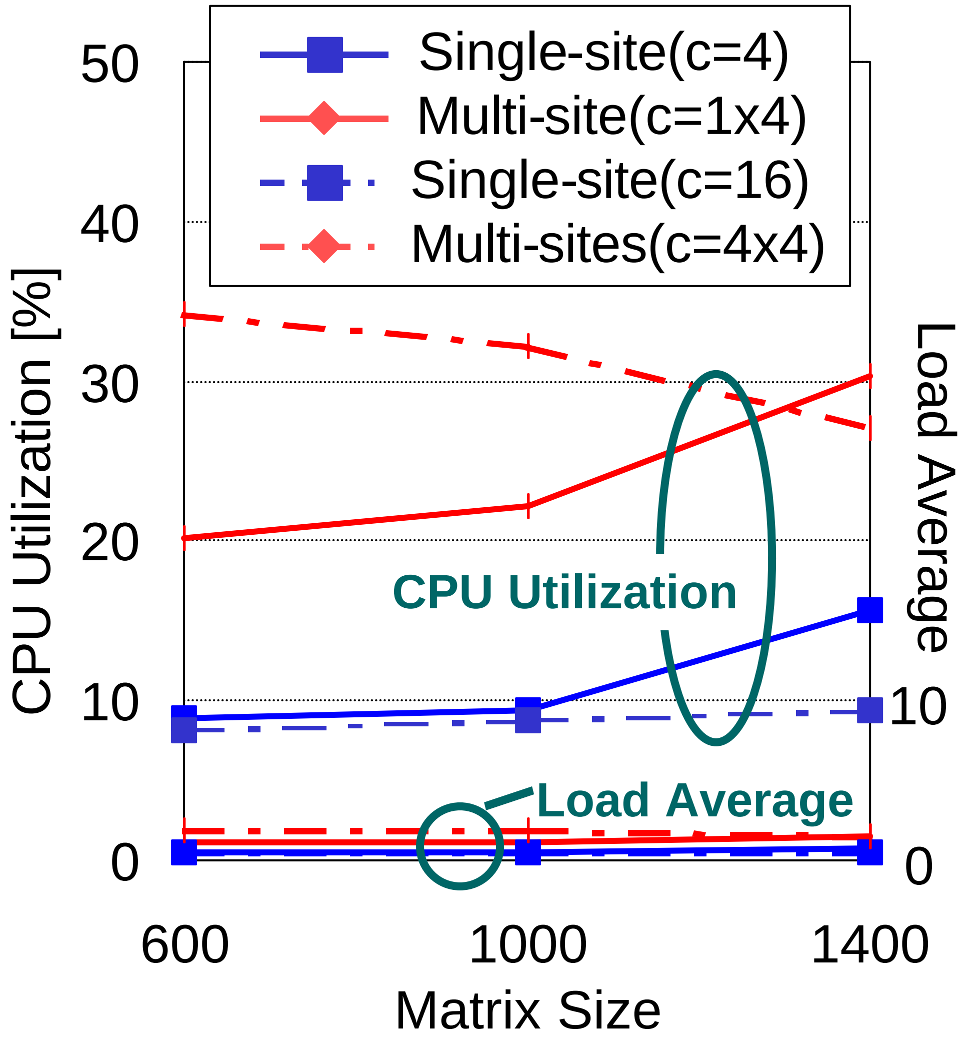
Single/Multi-site WAN Linpack Benchmark Results

Performance and Throughput (c = 16, 4PE ver.)



Single/Multi-site WAN Linpack Benchmark Results

CPU Utilization and Load Average

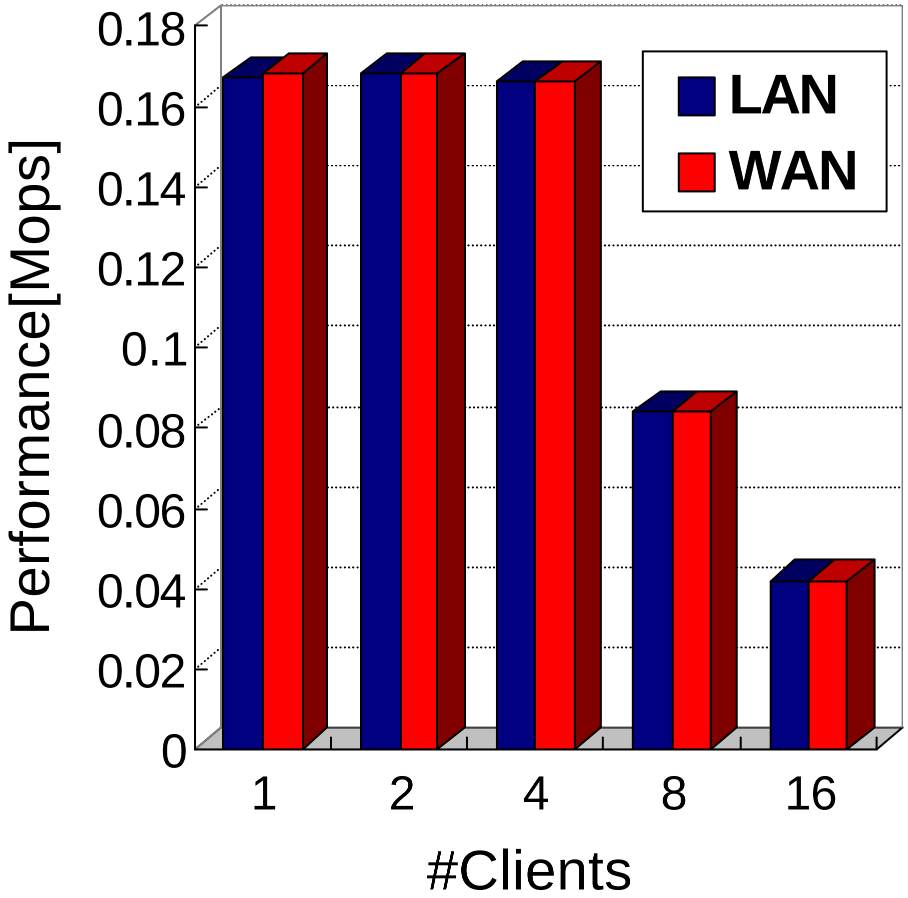


- Utilization and Load are greater for multi-site.

  c.f., single site.
- The J90 server does not saturate for n and c.
  - ***Network bandwidth saturation*** again the cause.

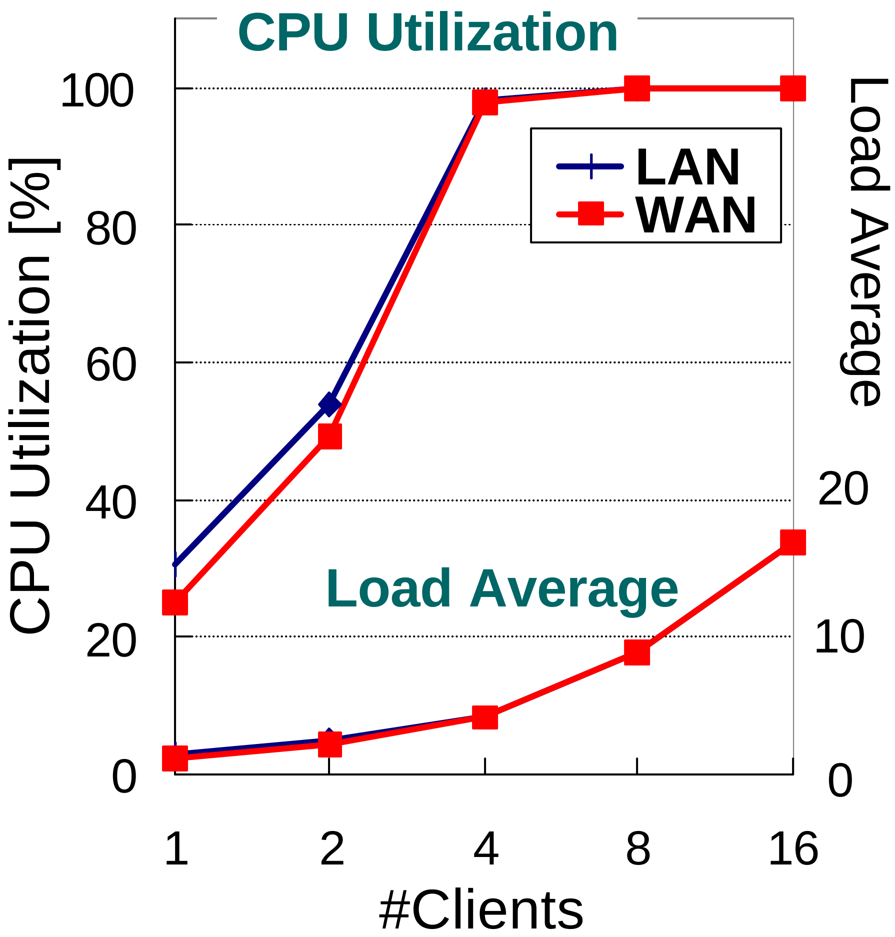
→ **Utilization and Load alone are NOT suitable criteria for load balancing of global computing.**

LAN/WAN EP Benchmark Results Average Performance (1PE ver.)



- Because of task-parallel execution, the performances decline when c=8, 16.
- In both LAN and WAN cases Ninf_call performances are almost equivalent.

LAN/WAN EP Benchmark Results
CPU Utilization and Load Average



- Both LAN and WAN cases are almost equivalent for Utilization and Load.

→ **Global computing can now be considered quite feasible** for this class of applications:

  – parallel rendering / imaging

  – parameter sensitivity analysis

Related Work

- RPC based systems → **use existing programming languages**
  - **NetSolve** [Casanova and Dongarra, Univ. Tennessee]
    - The same basic API as Ninf_call (**now interchangeable**)
    - load-balancing with a daemon process called **Agent**.
  - **RCS** [Arbenz, ETH Zurich]
    - PVM-based
- Systems using parallel distributed language etc.
  - **Legion** [Grimshaw, Univ. Virginia]
    - An user distributes his programs written with the parallel object-oriented language **Mentat**.
  - **Javelin** [Schauser et al., UCSB]
    - High portability due to using **Java** and **WWW**.
- Global scheduling systems - **NWS, AppLeS, DQS**

Conclusion

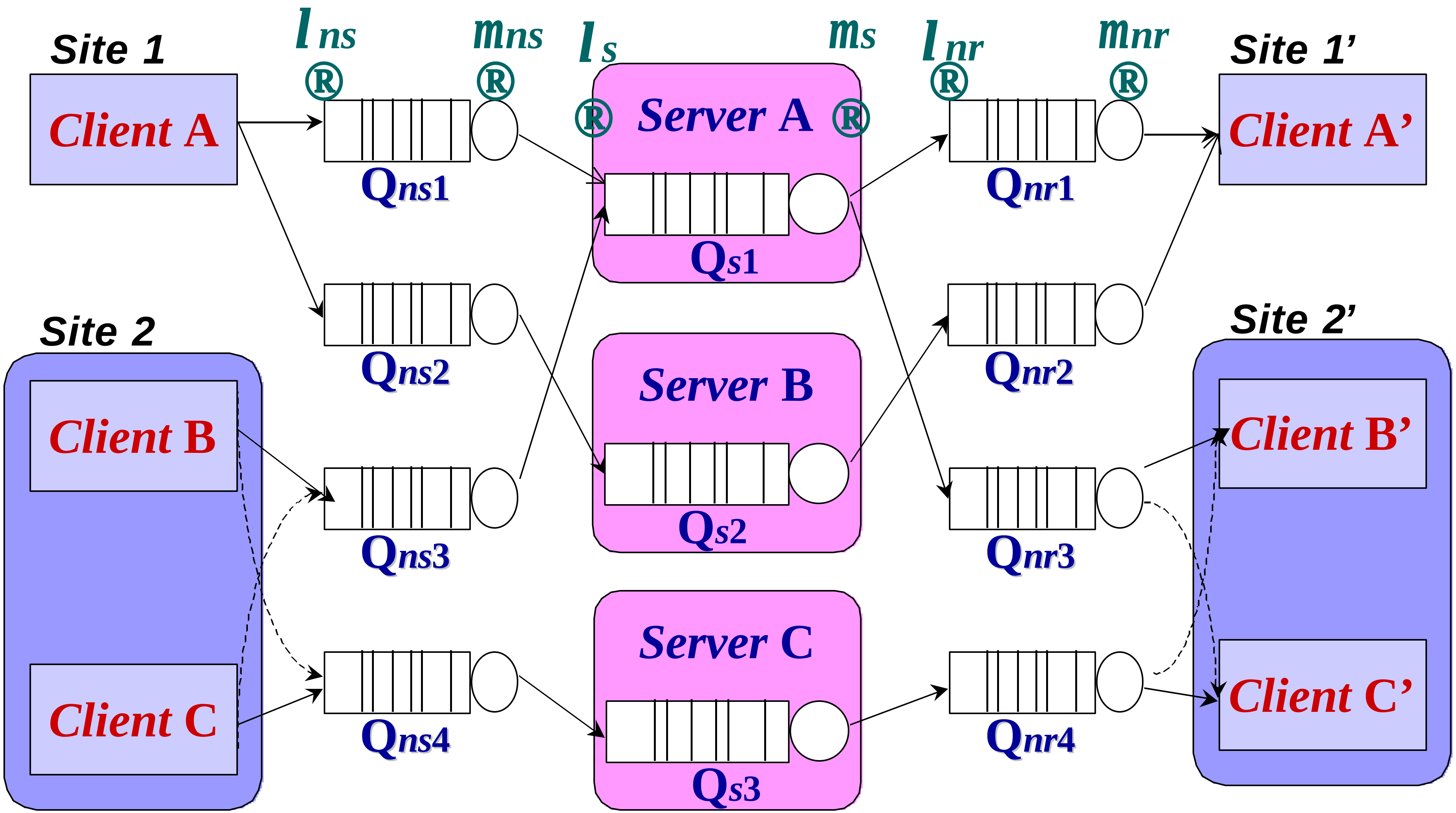
- The current Global Computing systems are likely to work well in both LAN and WAN situations.
- The use of optimized parallel library would be sufficient for vector computing servers.
- In LAN, computing server performance dictates overall performance, while in WAN limitation of communication throughput is more significant. (Esp. communication-intensive applications)
- We expect multiple client requests will be issued from different sites, causing “false” lowering of load ave..

The scheme which properly dispatches communication-/computation-intensive jobs to the servers is important.

Future Work

- Load Balancing for the Global Computing
  - Meta Server (*c.f.* NWS)
  - Develop a theoretical model
    - Performance analysis for more practical situations
    - **Global Computing Simulator**
- Make the Ninf system more powerful
  - Guaranteeing Performance in multi-client global computing
  - Server Job-handling Methodology (FCFS → SJF)
  - Multi-Job Scheduling for MPP servers (FPFS, FPMPFS)
  - Extension of the API for MPP's network services

The Model of Ninf Simulator



Future Work

- Load Balancing for the Global Computing
  - Meta Server (*c.f.* NWS)
  - Develop a theoretical model
    - Performance analysis for more practical situations
    - **Global Computing Simulator**
- Make the Ninf system more powerful
  - Guaranteeing Performance in multi-client global computing
  - Server Job-handling Methodology (FCFS → SJF)
  - Multi-Job Scheduling for MPP servers (FPFS, FPMPFS)
  - Extension of the API for MPP's network services

Exhibition

Electro-
technical
Laboratory

Zone R2