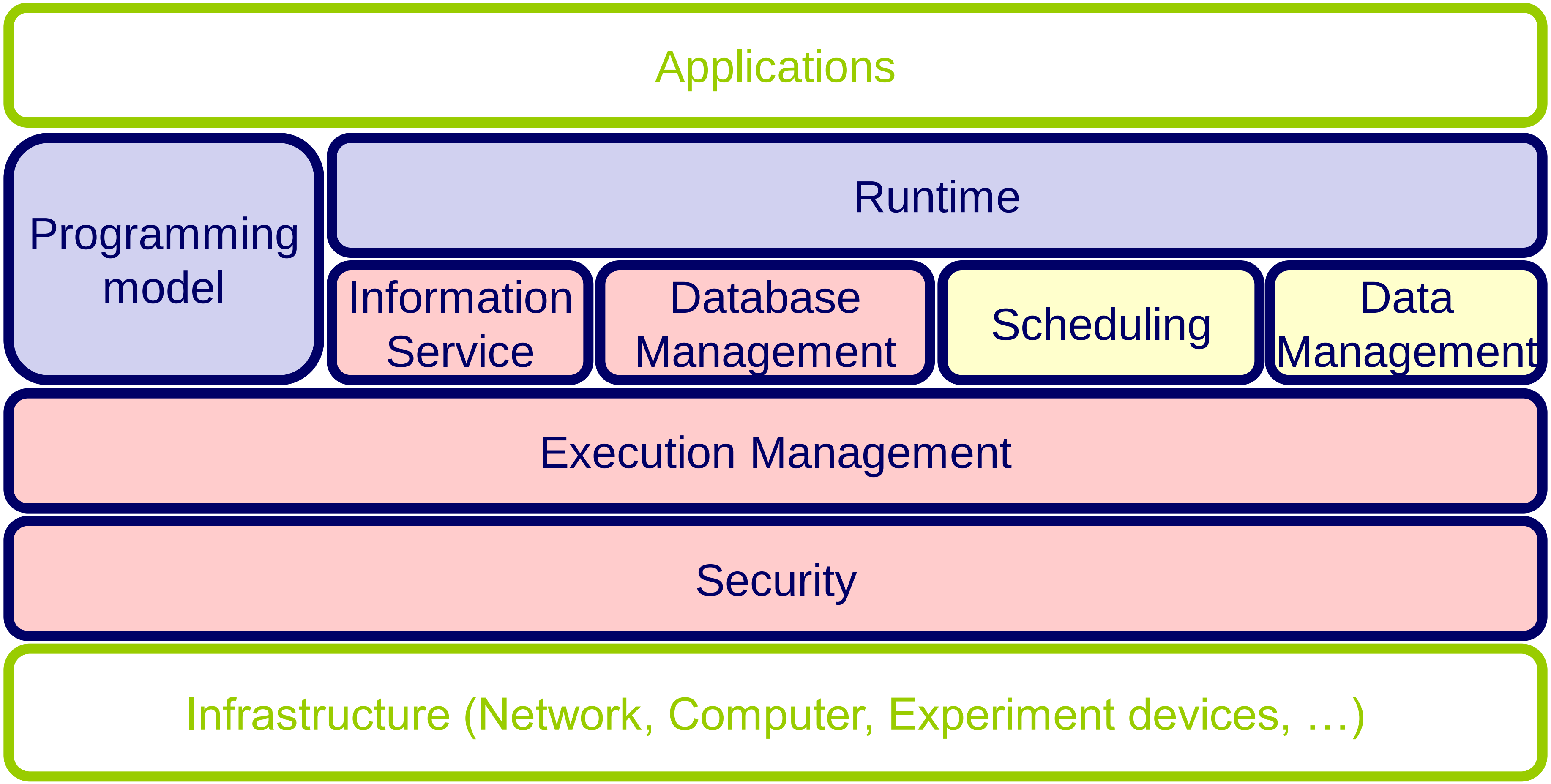
Grid Programming (2)

Osamu Tatebe
University of Tsukuba

Overview

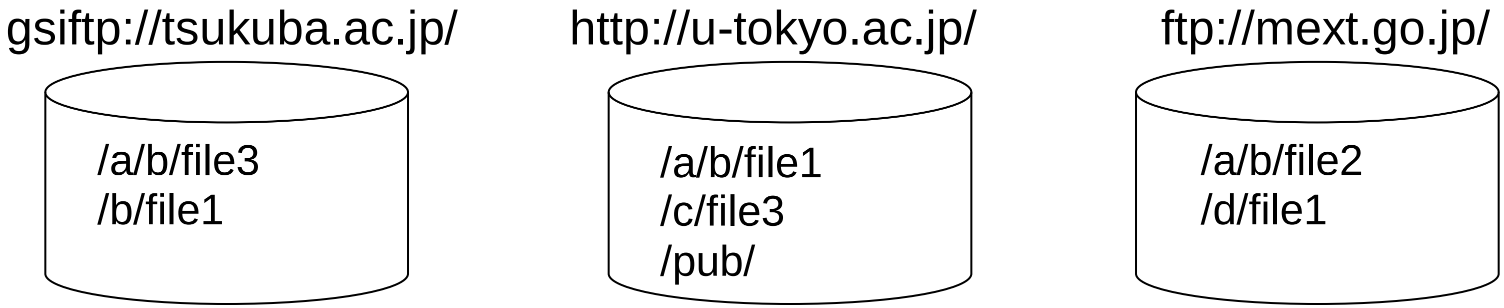
- **Grid Computing**
  - Computational Grid
  - Data Grid
  - Access Grid
- **Grid Technology**
  - Security - Single Sign On
  - Information Service
  - Data management
  - Widearea Data Transfer
  - Resource Management
- **Open Grid Forum (OGF)**
  - http://www.ogf.org/

Grid Technology



Data Management in Grid

- **Network Storage Access requires**
  - Server name
  - Protocol
  - Path name in server
- **A file may be migrated due to some reason**
- **File replicas may be inconsistent**
- **High performance access to large files may be required**



Roles of data management (1)

- **Provides**
  Easy, fast, and stable access
  for necessary files and data in Grid
- **"Easy" access (transparency)**
  - Enable to access data by specifying path name, search expression, and search criteria (location transparency)
    - Not specify the server name and the protocol
  - Requires a mechanism to resolve the location and protocol from the path name and the search expression (resolver)
    - In case of path name, it is called directory management service
    - In case of search expression, it is called metadata management service
  - Indirect management improves the flexibility using dynamic binding of server and protocol
    - To cope with file migration, and enable dynamic file replica selection

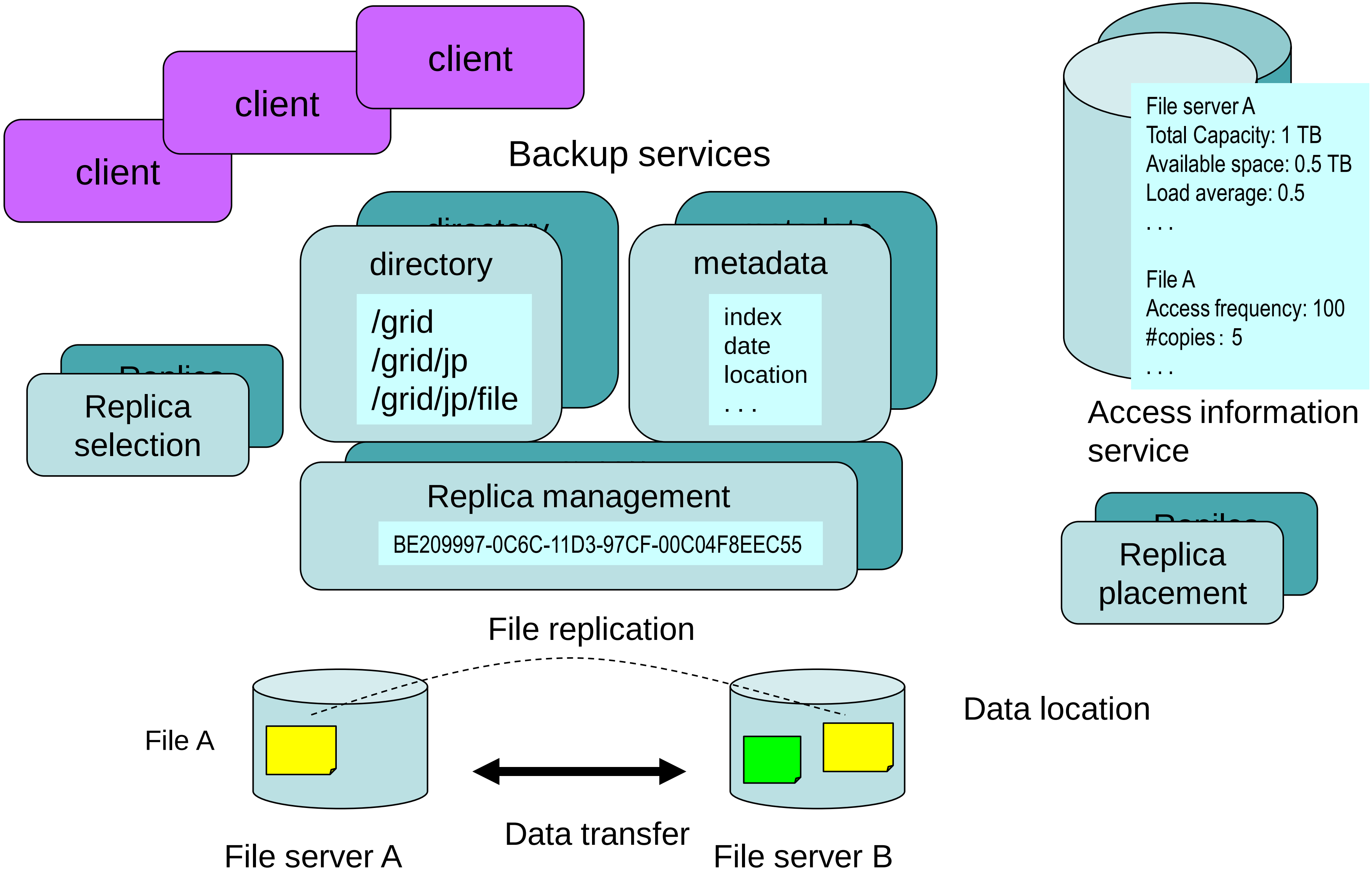
Roles of data management (2)

- **“Fast” access**
  - Increase access bandwidth
    - Fast data transfer technique
    - Access distribution by file replicas at appropriate places
  - Reduce access latency
    - Select a near replica in terms of network latency
    - Create replicas at frequently accessed locations
    - Create replicas of frequently accessed data to avoid access contention

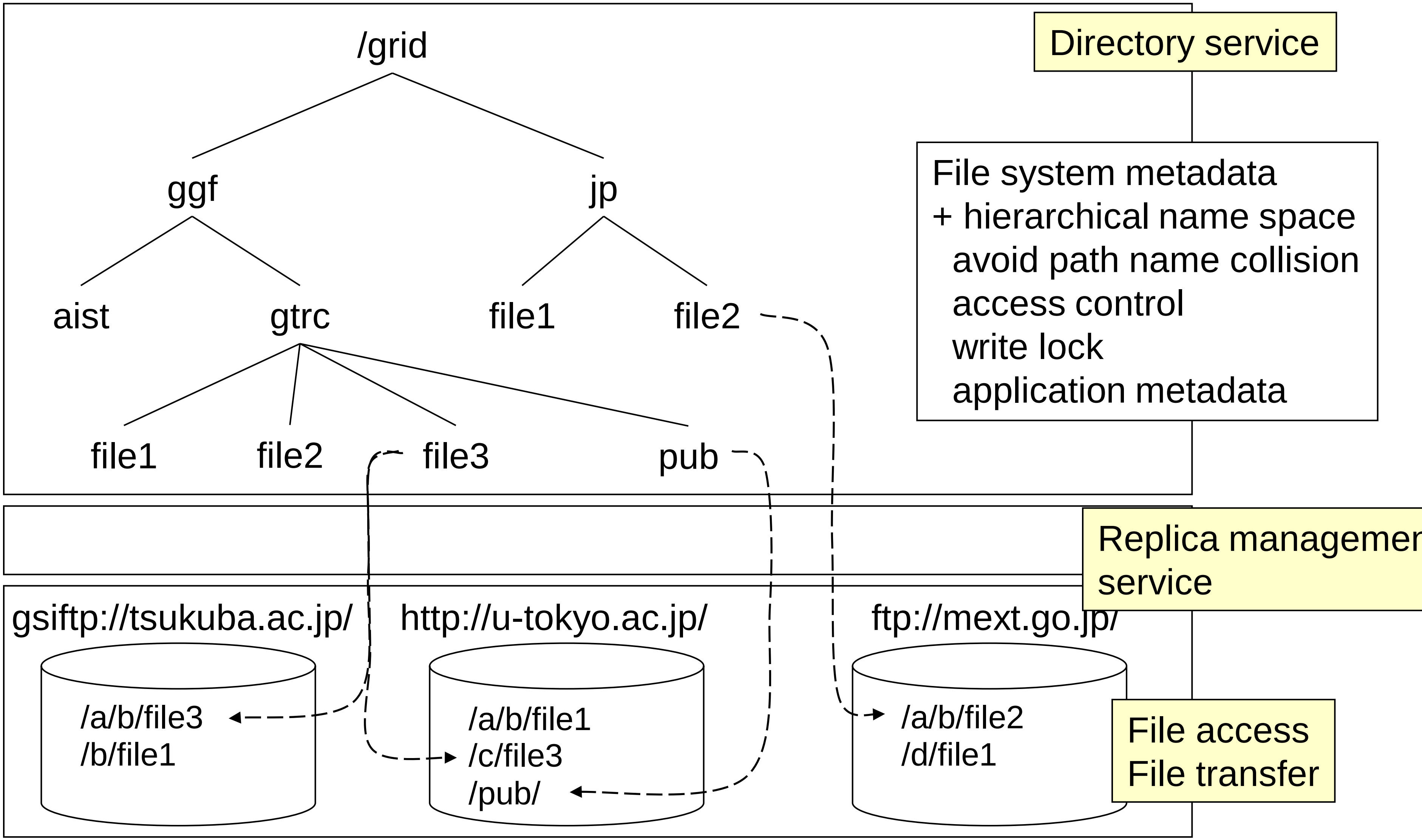
Roles of data management (3)

- **“stable” access**
  - Enable transparent access even if storage and network fail
  - Avoid SPOF (Single Point of Failure)
    - A point that makes the whole system down
  - To avoid data loss, create replicas at different locations
  - Duplicate directory and metadata management service
    - If it is lost, no way to access data even though there is data

Service federation in file management

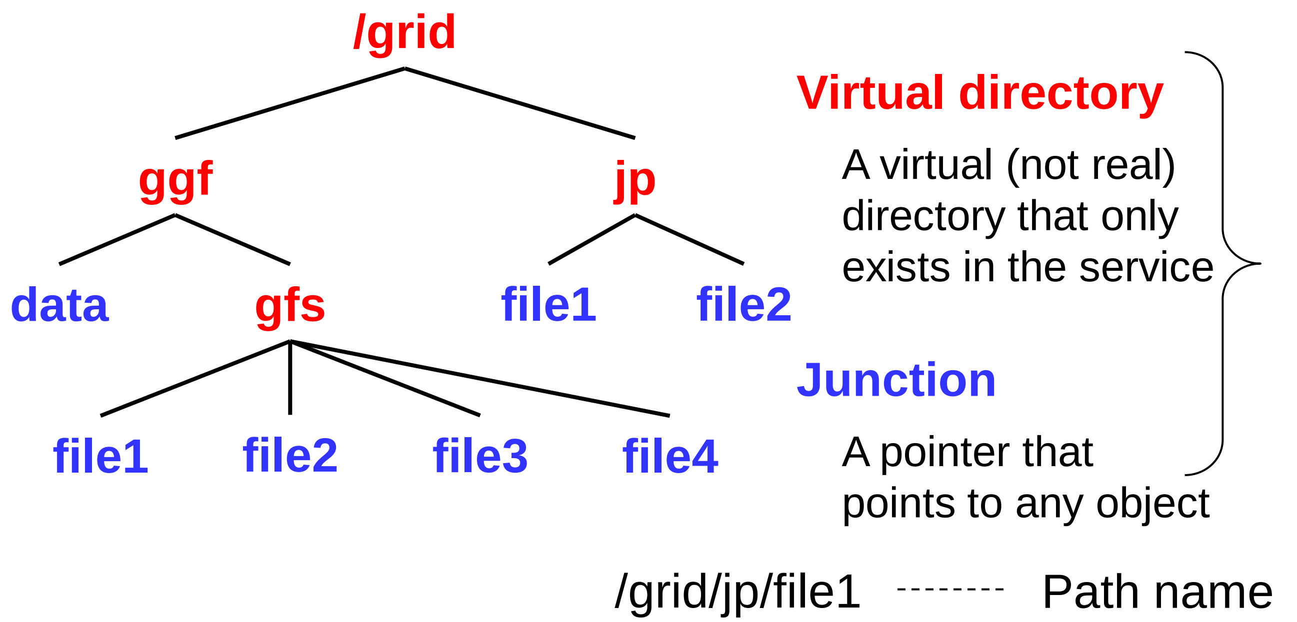


An example of file management

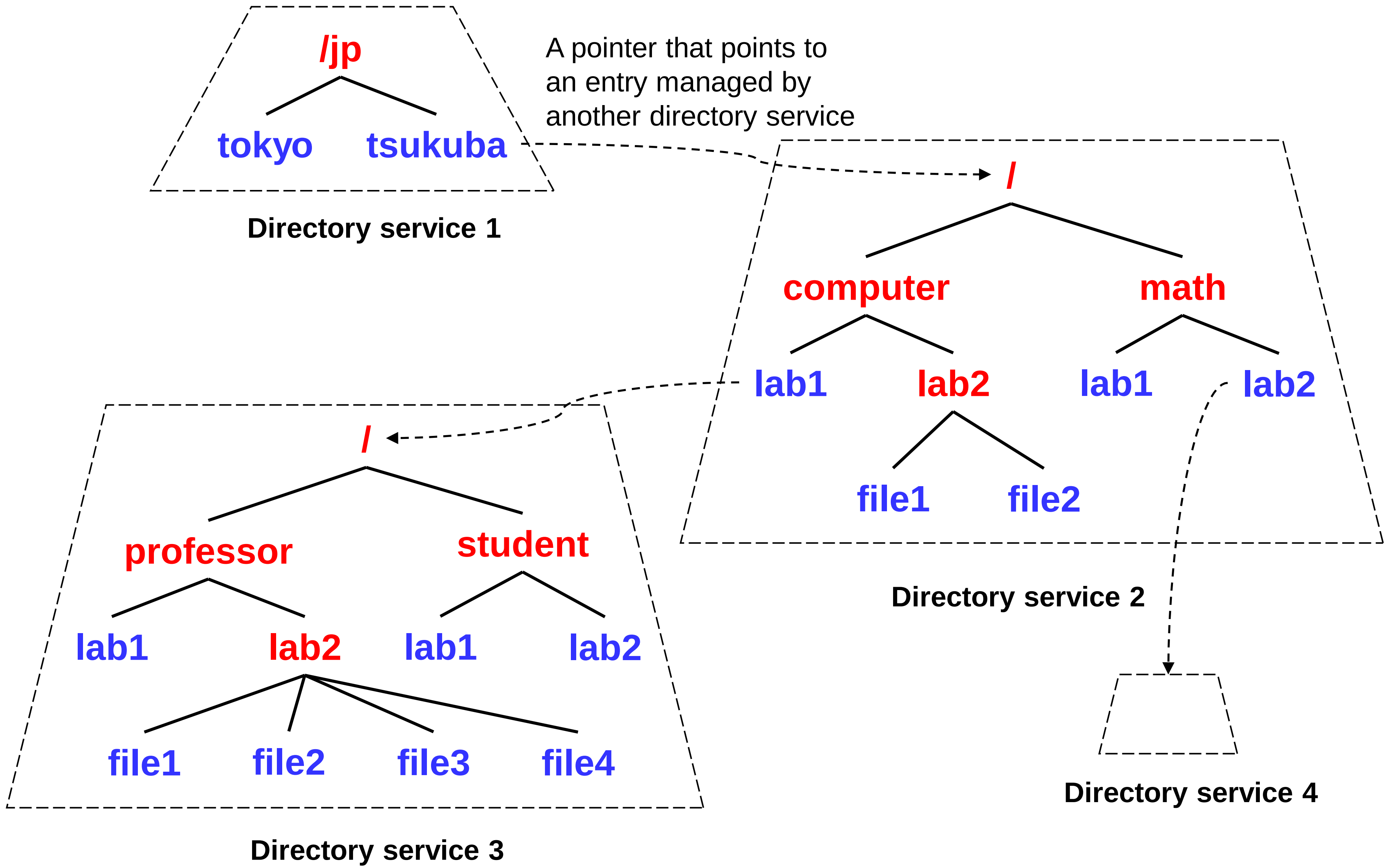


Directory management service

- **Virtual hierarchical namespace**
  - E.g. File system directory tree
- **Pointers that point to a file, data, file system, and database**



Distributed directory management



A standard of directory management - RNS

- **Open Grid Forum**
  - http://www.ogf.org
- **M. Morgan, A. Grimshaw, O. Tatebe, "RNS Specification 1.1", GFD.171, 2010**
- **5 interfaces**
  - Add, lookup, remove, rename, setMetadata
- **Application metadata can be added to each entry**
  - Metadata for Grid file system
  - Application metadata for bio informatics, nano science, and physics

Pseudo-UML for Data Types

RNSEntry
entryName: String [endpoint: EPR] [metadata: XML]

RNSEntryResponse
entryName: String [endpoint: EPR] [metadata: XML] [fault: FaultType]

LookupResponse
[entry: []RNSEntry] [iterator: IteratorRef]

NameMapping
sourceName: String targetName: String

MetadataMapping
entryName: String metadata: XML

RNS 1.1 Port Type Pseudo-UML

RNS Resource Properties
elementCount: unsignedLong createTime: dateTime accessTime: dateTime modificationTime: dateTime readable: boolean writeable: boolean
RNS operations
add(entry: []RNSEntry): []RNSEntryResponse lookup(entryName: []String): LookupResponse remove(entryName: []String): []RNSEntryResponse rename(entry: []NameMapping): []RNSEntryResponse setMetadata(entry: []MetadataMapping): []RNSEntryResponse

Add (1)

- **add request message**

```
<rns:add>
  <rns:entry entry-name="rns:EntryNameType">
    <rns:endpoint>
      wsa:EndpointReferenceType
    </rns:endpoint> ?
    <rns:metadata> {any}* </rns:metadata> ?
  </rns:entry> +
</rns:add>
```

Add (2)

- **addResponse response message**

```
<rns:addResponse>
  <rns:entry-response entry-name="rns:EntryNameType">
    <rns:endppoint>
      wsa:EndpointReferenceType
    </rns:endpoint> ?
    <rns:metadata>
      <rns:supports-rns value="rns:supportType"/>
      {any}*
    </rns:metadata> ?
    <rns:fault> {fault} </rns:fault> ?
  </rns:entry-response> +
</rns:addResponse>
```

Replica management (1)

- **It is effective to improve access performance, access stability**
  - File modification is not often in Grid
- **Replica management service**
  - Manages “logical name”,
  - Translates a “logical name” to a list of pointers to the identical objects
- **“logical name” is an ID that uniquely identifies a data**
  - It is often human unfriendly (machine readable) such as UUID
  - Used with directory management service
- **Three-tier naming scheme**
  - Human readable name -> location independent name -> location dependent address
  - Path name -> logical name -> pointer

Replica management (2)

- **Replica selection**
  - Select the most appropriate file replica
  - In some criteria, e.g. minimum data access time
    - Near replica in terms of network latency when data is small
    - A replica connected by fat network when data is large
- **Replica placement**
  - Select where a file replica is created
    - Reduce file replication time
    - Create a replica at a distant place for disaster recovery
    - Avoid hot spot
      - Identify hot (frequently accessed) files
      - Create file replicas to avoid access concentration as much as possible

A standard of replica management – WS-Naming

- **A resolver from “logical name” to a pointer**
- **A. Grimshaw, D. Snelling, “WS-Naming Specification”, GFD.109, 2007**
- **A pointer is represented by an EPR (WS-Addressing Endpoint Reference)**
- **It extends EPR that includes “logical name” as an endpoint identifier**
- **Identity can be known by the endpoint identifier**
- **Resolver addresses can be included in EPR to update the address (the pointer)**

An example of WS-Naming

```
<wsa:EndpointReference xmlns:wsa="…" xmlns:naming="…">
        <wsa:Address>http://tempuri.org/application</wsa:Address>
        <wsa:Metadata>
                <naming:EndpointIdentifier>
                        urn:guid:B94C4186-0923-4dbb-AD9C-39DFB8B54388
                </naming:EndpointIdentifier>
                <naming:ReferenceResolver>
                        <wsa:Address>
                                http://tempuri.org/resolver
                        </wsa:Address>
                </naming:ReferenceResolver>
        </wsa:Metadata>
</wsa:EndpointReference>
```

Endpoint identifier

Resolver to update the EPR

Service federation in file management

- **File replica creation of hot files**
  - Obtain a list of hot files from access information service
  - Decide the number of replicas depending on the access frequency
  - Decide location to be created by replica placement service
  - Decide a source replica for the file replication by replica selection service
  - Schedule file replica creations
  - Perform data transfer following on the schedule
  - Register replicas to replica management service if they are successfully created

Problems in service federation

- **Error and fault handling**
- **Network fails during data transfer**
  - Incomplete files remain

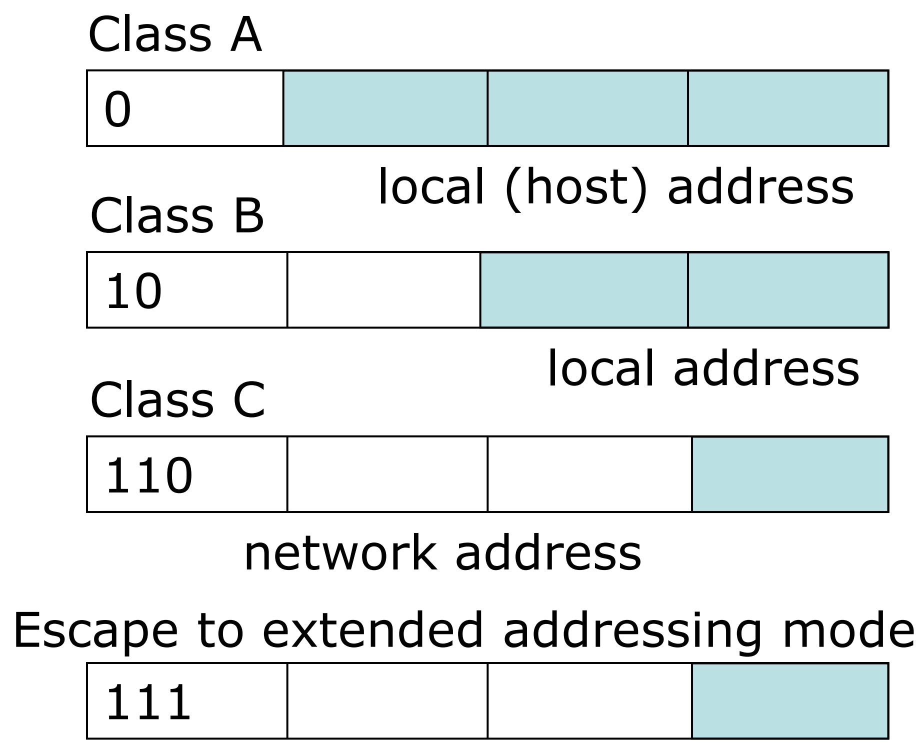
-> ensure a transaction of a series of service federation

  - Introduce monitor service to monitor the progress
  - Keep the checkpoint of the progress
  - Retry the execution at the error
  - Rollback if it fails
  - Consider failure of the monitor service

Widearea fast data transfer

IP (RFC791, 1981)

- **Internet Protocol**
- **Transfer datagram from source to destination specified by Internet address**
- **Long datagram may be fragmented**
  - MTU – Maximum Transmission Unit
  - DF (don't fragment) flag not to be internet fragmented
- **It provides only datagram transfer**
  - not reliable
  - Not include flow control
- **Time to live, checksum**
- **Internet address (IP address)**
  - Version 4
  - 32 bits

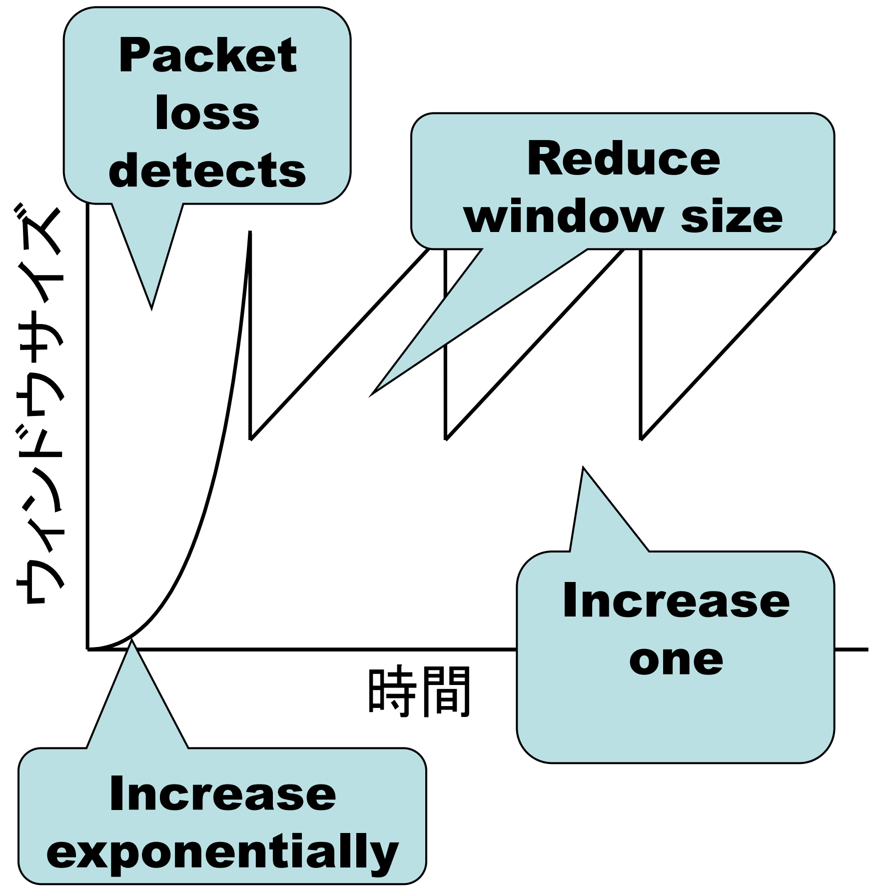


TCP (RFC793, 1981)

- **Transmission Control Protocol**
- **Reliable communication service between processes**
- **Basic data transfer**
  - Bidirectional byte stream data transfer
  - Push function to check the transfer, which sends the data immediately
- **Reliability**
  - TCP recovers from data that is damaged, lost, duplicated, or delivered out of data
  - Sequence number and acknowledgment (ACK)
  - It retransmits data if ACK is not received within a timeout interval (Retransmission timeout; RTO)
  - Receiver detects duplication and out of order by sequence numbers
  - Damage is detected by checksum
- **Flow control**
  - Receiver controls the flow
  - It returns "window" with every ACK (piggy back)
  - "window" indicates an allowed number of octets that sender may transmit before receiving further permission
  - Congestion control
    - Avoid too much traffic than the bottleneck link
- **Multiplexing**
  - A set of ports to allow for many processes within a single host to use TCP
  - A socket is formed by the network address and port. A part of sockets identifies each connection
  - Well-known port number (cf. /etc/services)
- **Connection**
  - connection should be established before data transfer between processes
  - Clock-based sequence numbers to cope with unreliable host and unreliable internet connection

Congestion window control algorithm

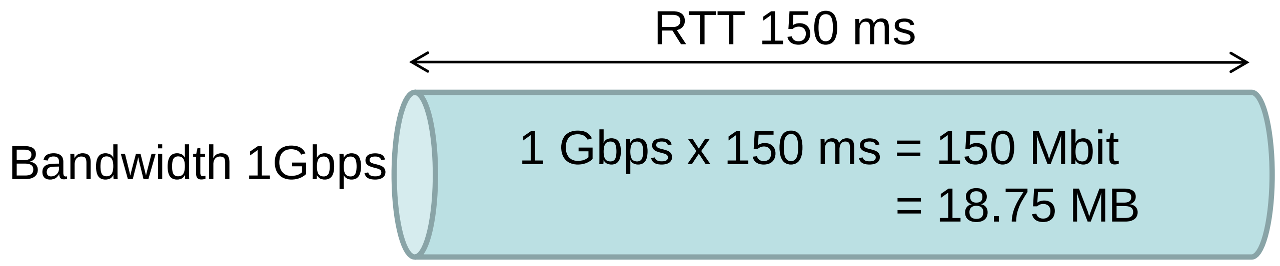
- **Control algorithm of congestion window**
  - It is changed by each Ack, which depends on RTT
- **Reno, Newreno, Vegas, Sack, Fack**
- **Reno is often used. It is stronger than others**
- **Congestion window change in Reno**
  - Slow start phase
    - At the beginning, when the window size is minimum
    - Increase window size exponentially
  - Congestion avoidance phase
    - Increase window size one by one
    - Reduce window size by half when packet loss detected



- **-> only 75% of peak available**

Data transfer in long fat pipe

- **Poor TCP performance in LFN (elephan(t), Long, Fat Network)**
- **RFC1323 TCP Extensions for High Performance (1992)**
  - TCP performance depends on not network speed by bandwidth delay product
  - Bandwidth delay product is data size in flight. To transmit data in the maximum bandwidth, the sender should send the amount



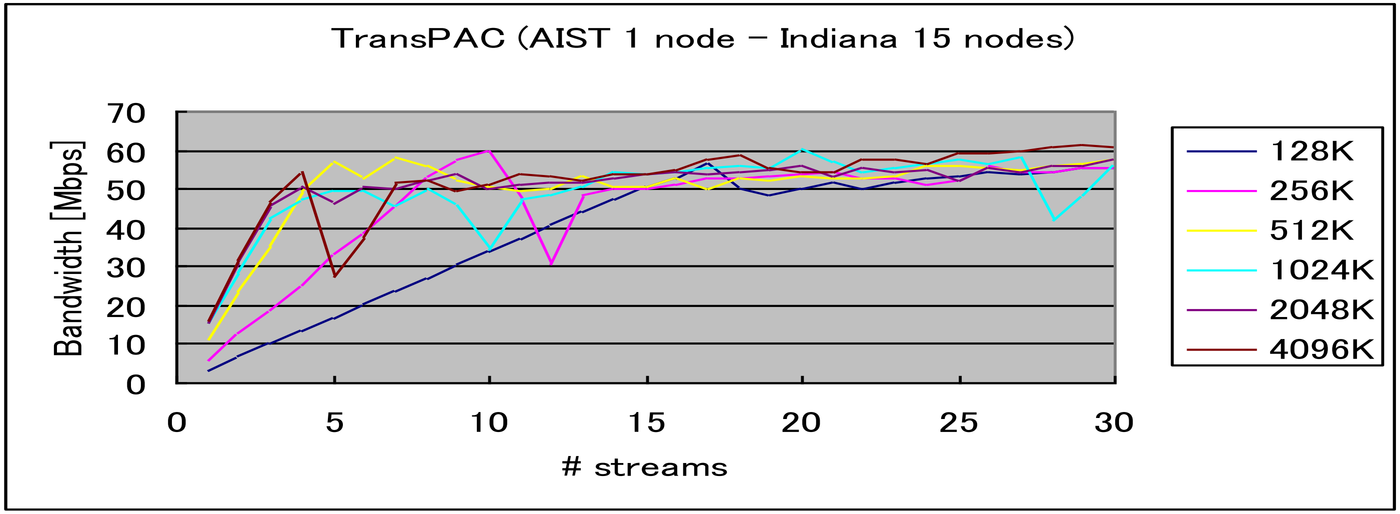
TCP performance problem over LFN

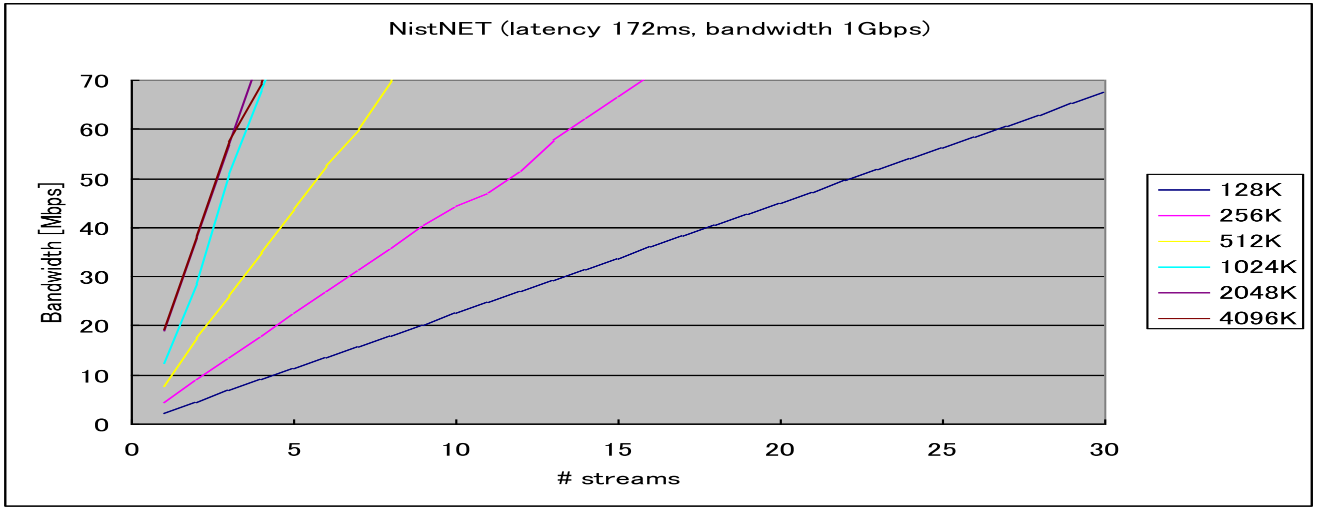
- **Limitation of window size**
  - Window size is specified by 16bit field in TCP header
    - Maximum window size = 64KB
    - Maximum band width = 64KB/RTT
  - Introduce Window Scale TCP option (RFC1323)
    - 16bit -> 30bit = 1GB (limitation of 31bit sequence number)
- **Recovery from packet loss**
  - Packet loss in LFN (large window size) is terrible
  - Data pipeline should be flushed and recovery by slow start
- **RTT measurement**
  - Dynamic measurement of RTO is essential for TCP performance
  - RTO is decided by average and dispersion of RTT (round-trip time)
  - Introduce Timestamps TCP option (RFC1323)
- **It takes long time to increase window size in case of large RTT**
  - Increase MTU (jumbo frame)
  - HighSpeed TCP (RFC3649), Scalable TCP, CUBIC (Linux default), Compound TCP (Windows Vista default)

Network striping

- Root privilege required to specify large socket buffer size
- Network striping is data transfer using multiple streams in application level
- The same effect to specify the default buffer size times # of streams

Network performance (TransPAC (Tokyo – Seattle) and NistNET)





GridFTP (GFD20, 2003)

- **GridFTP: extended version of popular FTP protocol for Grid data access and transfer**
- **Secure, efficient, reliable, flexible, extensible, parallel, concurrent, e.g.:**
  - Third-party data transfers, partial file transfers
  - Parallelism, network striping, striping server (e.g., on PVFS)
  - Automatic and manual TCP tuning
  - Reliable, recoverable data transfers, data channel authentication
- **Reference implementations**
  - gridftp-server, globus-url-copy, uberftp
  - Flexible, extensible libraries in Globus Toolkit

Extension of GridFTP

- **Protocol extension**

SPAS	Striped Passive	Return array of Host/port
SPOR	Striped Port	Return array of Host/port
ERET	Extended Retrieve	Transfer a part of a file
ESTO	Extended Store	Store a part of a file
SBUF	Set TCP Buffer Size	Specify TCP buffer size
ABUF	Auto-negotiate TCP Buffer Size	Decide TCP buffer size automatically
DCAU	Data Channel Authentication	RFC2228 introduces GSS auth for control channel, but not for data channel

- **Mode extension**
  - EBLOCK (Extended block) mode
  - Transfer data in block in parallel
  - 8bit flag, 64bit data size, 64bit offset, data

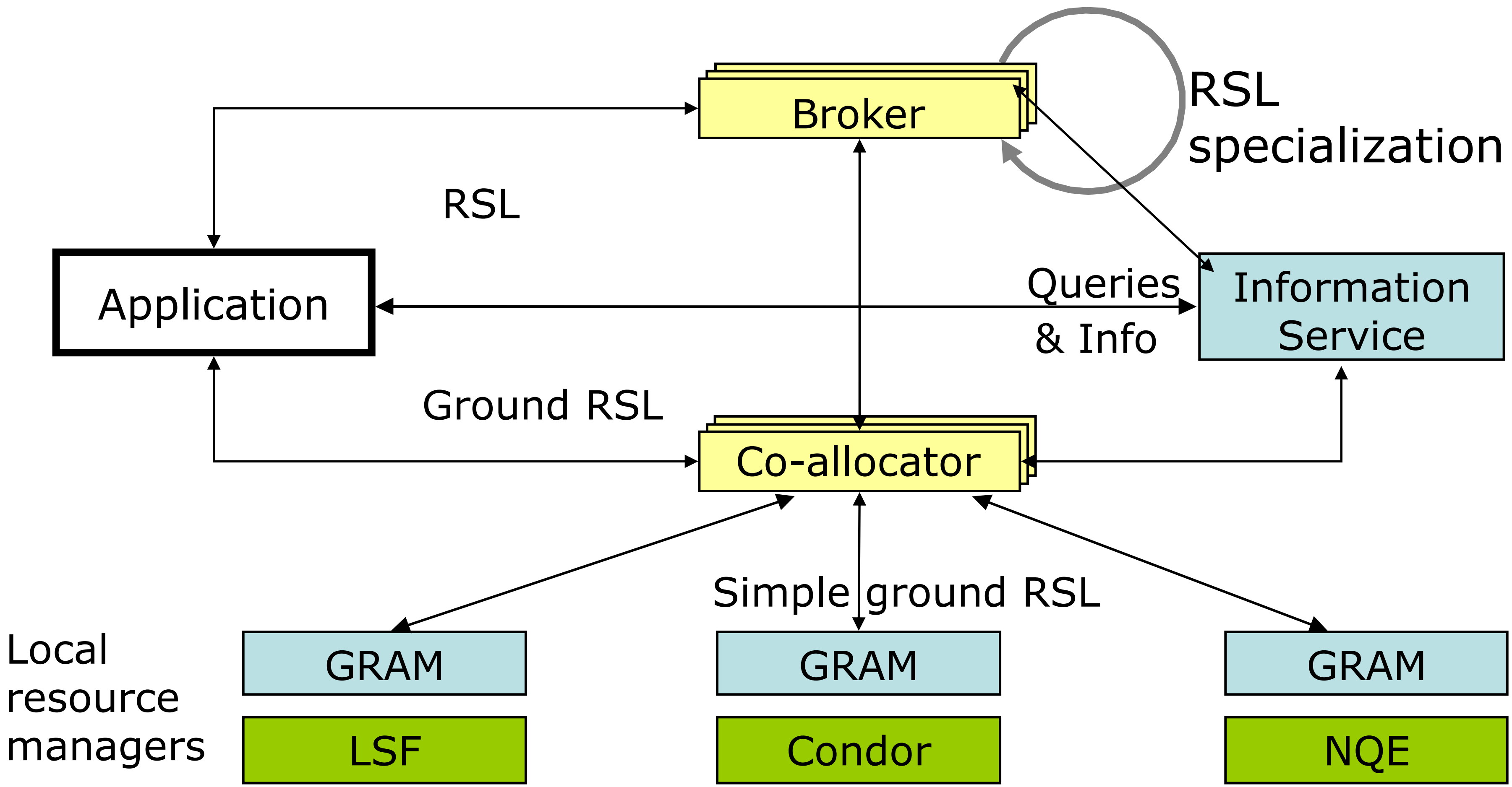
Papers: Widearea fast data transfer

- H. Sivakumar, S. Bailey, R. L. Grossman. PSockets: The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks, Proc. SC2000 http://www.sc2000.org/techpapr/papers/pap.pap240.pdf
- Thomas Dunigan, Matt Mathis, Brian Tierney. A TCP Tuning Daemon, Proc. SC2002 http://www.sc2002.org/paperpdfs/pap.pap151.pdf
- Thomas J. Hacker, Brian D. Noble, Brian D. Athey. The Effects of Systemic Packet Loss on Aggregate TCP Flows, Proc. SC2002 http://www.sc2002.org/paperpdfs/pap.pap270.pdf
- B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke. Data Management and Transfer in High Performance Computational Grid Environments. Parallel Computing Journal, Vol. 28 (5), May 2002, pp. 749-771. http://www.globus.org/research/papers/dataMgmt.pdf
- W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, S. Meder, S. Tuecke. GridFTP Protocol Specification. GGF GridFTP Working Group Document, September 2002. http://www.globus.org/research/papers/GridftpSpec02.doc

Papers: replica management

- A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunst, M. Ripenu, B, Schwartzkopf, H, Stockinger, K. Stockinger, B. Tierney. Giggle: A Framework for Constructing Scalable Replica Location Services. Proc. SC2002 http://www.sc2002.org/paperpdfs/pap.pap239.pdf

Resource Management



Papers: resource management

- Rajesh Raman, Miron Livny, and Marvin Solomon, Resource Management through Multilateral Matchmaking, Proc. Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9), August 2000, pp 290-291. http://www.cs.wisc.edu/condor/doc/gangmatching.ps
- Fabio Kon, Roy Campbell, M. Dennis Mickunas, Klara Nahrstedt, and Francisco J. Ballesteros. 2K: A Distributed Operating System for Dynamic Heterogeneous Environments. Proc. Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9), August 2000. http://choices.cs.uiuc.edu/2k/papers/hpdc2000.pdf