



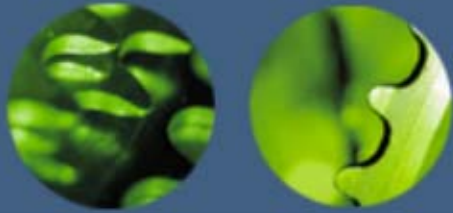
# GridFTP & GridRPC

Yang Wang



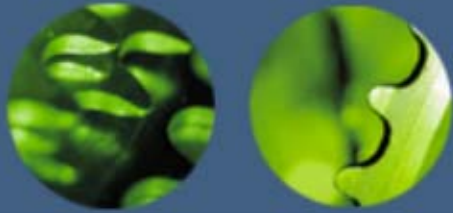
# GridFTP

- GridFTP Overview
  - GridFTP is a transport mechanism for the Grid environment.
  - Requirements for the transport mechanism
    - GSI and Kerberos support
    - 3<sup>rd</sup> party control of data transfer
    - Parallel data transfer
    - Striped data transfer
    - Partial file transfer
    - Support for reliable file transfer
    - Manual control over TCP buffer size
    - Integrated instrumentation



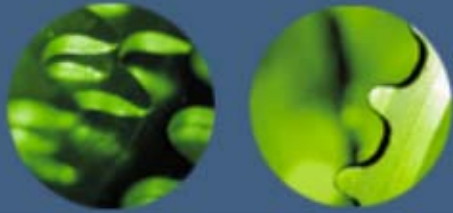
## GridFTP

- Motivation for a new transport mechanism
  - A variety of storage systems: DPSS, HPSS, DFS, SRB, HDF5
  - Each of these storage systems utilize different, incompatible protocols.
  - Need a common level of interoperability.



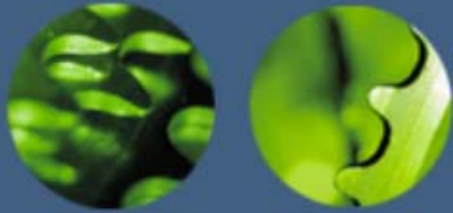
## GridFTP

- Where to place the layer of interoperability?
  - Above the storage system?
    - Advantage: common interface for applications
    - Disadvantage: hides special features
  - Below the storage system?
    - Advantage: future systems automatically compatible
    - Disadvantage: retrofit existing systems.



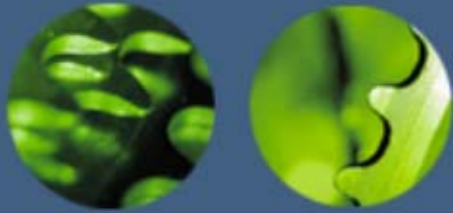
## GridFTP (cont'd)

- Why FTP?
  - Protocols considered: FTP, HTTP, HPSS, DPSS, SRB
  - HPSS, DPSS and SRB all use unpublished/proprietary protocol (more comment on the SRB protocol later)
  - Either FTP or HTTP, but which?



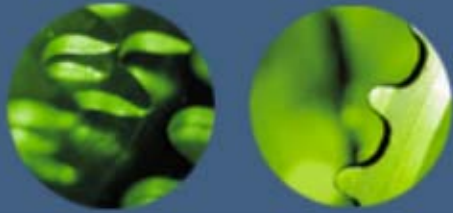
## GridFTP (Cont'd)

- Comparison between standard FTP and HTTP protocols
  - Bulk Transport: FTP, yes. HTTP, no.
  - Open Standard: Yes for both
  - Secure: Both can have GSI integrated
  - Fast: Both can do parallelism, but neither supported striping. Note the separation of control and data channels in FTP
  - Robust: simple restarts.
  - 3rd party control: FTP, yes. HTTP, no.
  - Instrumentation: Neither



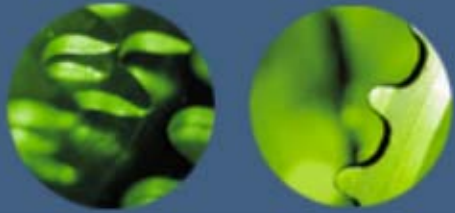
## GridFTP (cont'd)

- The protocol
  - FTP-like communication. Extends RFC959, the original FTP protocol specification with 7 new commands.
    - Striping and parallelism: SPAS, SPOR
    - Server-side processing: ERET, ESTO
    - Buffer control: SBUF, ABUF
    - Channel Authentication: DCAU



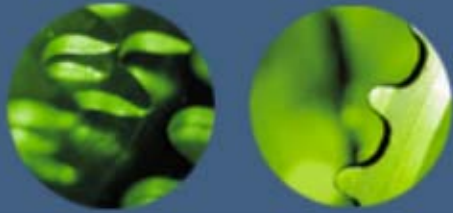
# GridRPC

- What is RPC?
  - Procedure call. the caller places arguments to a procedure, transfers control over, and eventually gains back control.
  - Remote Procedure Call. Calling and called processes need not reside on the same host. (see RFC1050 for protocol)
  - Benefits: transport independent, avoids the details of the underlying network, client-server model.
  - Blocking/non-blocking.



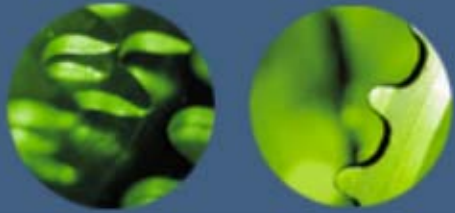
# GridRPC

- Why GridRPC
  - The Grid problem: programming problem. How do we program on the grid in an easy manner
  - GT provides low-level services, such as authentication, job launching, directory service, etc.
  - Gap between these low-level services and the more familiar programming concepts.



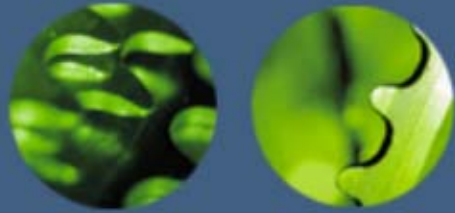
## GridRPC

- High-level view of the GridRPC model
  - Standard RPC plus asynchronous coarse-grained parallel tasking. Hides the dynamicity, insecurity and instability of the Grid from the programmers.
  - Representative GridRPC systems have been built on top of Netsolve and Ninf
  - Can be built on Grid software based on OGSA



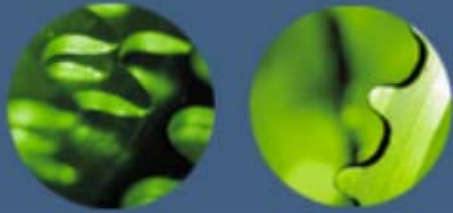
# GridRPC

- Current Directions
  - Related: CORBA, RMI, XML-RPC
  - Trying to define the wire protocol using XML
  - How to pass binary parameters in XML-based RPC protocol?
    - In scientific computing, large arrays are often passed back and forth.
    - Encode/Decode: efficiency and precision concerns.
    - Hybrid protocol?



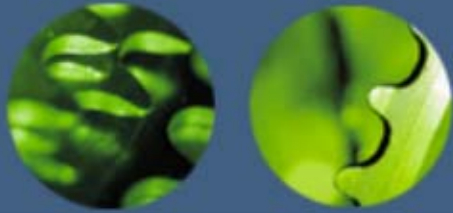
## In practice

- So many Grid-related technologies. In particular, we've seen: GridFTP, SRB, GT
- Let's take an engineering approach to solving real-world problems.
- We will analyze a research project undertaken by UMIACS, which attempts to integrate the Grid technologies into a usable prototype towards a real world problem.



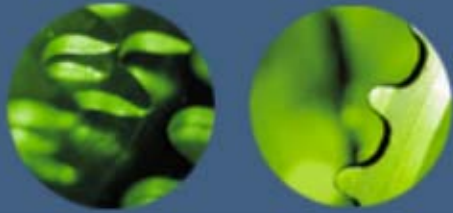
## In practice

- The Project
  - Ingestion and Long-term Preservation of the Electronic Records at the National Archives
  - Institutions Involved:
    - NARA, UMIACS, SDSC
    - A number of other universities in various stages and capacities, UVT, Georgia Tech



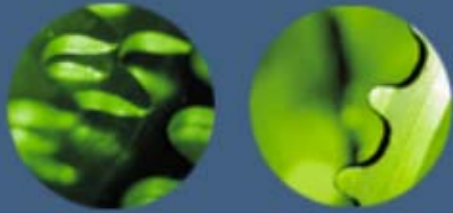
## In practice

- The problem
  - NARA gets a huge amount of documents to be archived a preserved.
  - Federal Law forbids government agencies form discarding records. Everything must be transported to NARA. Only NARA can discard records it deems unnecessary.
  - More and more records come from various government agencies in electronic form.



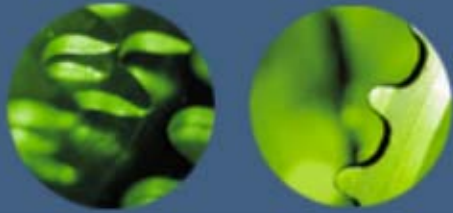
## In practice

- Current practice
  - Ingestion: truck-loads of tape shipped to NARA central receiving
  - Data processing and archiving: read the box label (and tape label if necessary).
  - Long-term preservation: tape archive. (well... not in the most advanced sense)

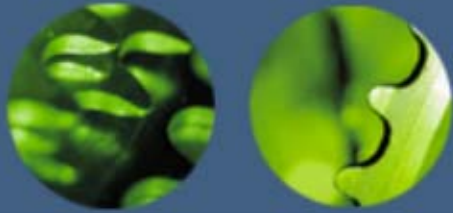


## In practice

- Problem: inefficient, labor-intensive, lack of error detection and correction.
- Takes a long time to process the data. NARA is currently processing data from the 1980-90s. 15 year lag. This is allowed by the law, though.
- With the explosion of IT in the 90s, when NARA processes data from this period of time, it will be overburdened and the lag will be increased.

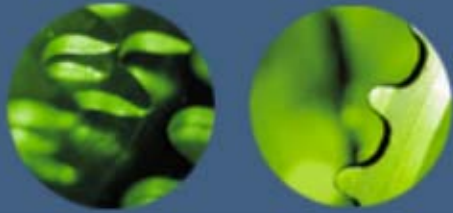


- Using Web services and Grid Technology to solve this problem.
  - Core idea: use the data grid to archive and preserve the data.
  - Existing Grid security infrastructure can be used for client access purposes.



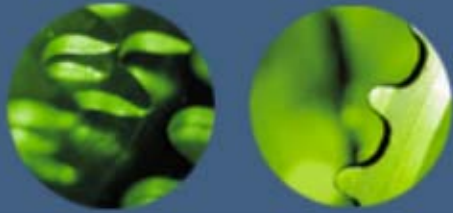
## In practice

- Grid Data storage
  - Use SRB for data archiving
    - self-replication. Can be couple with error discovery. Also, since this is a “one write, multiple reads” situation, data replications can geographically distributed for more efficient client access
    - Metadata query is ideal for accessing archive data. Archivists at NARA is also very accustomed to the idea of the “catalog”, i.e. metadata.

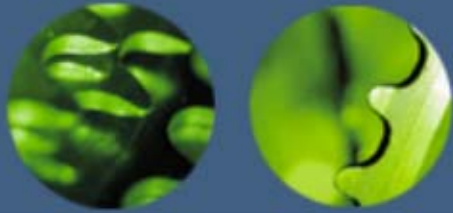


## In practice

- The problem
  - SRB does not work with all databases.
    - UMIACS developed the Informix driver for SRB
  - SRB is a great concept, poorly engineered.
    - The quality of code base is very poor. (for example, until SRB 2.0, the single quotes are not escaped in SQL queries, resulting in malformed queries at best and unauthorized access at worst.)
    - Bad database design. (unnecessary tables, aliases, lack on default indexing, etc)

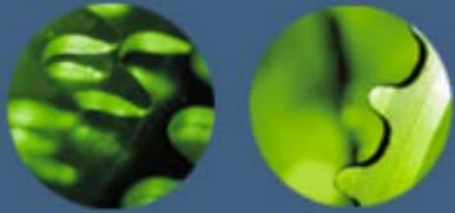


- The biggest problem
  - No documentation on the underlying protocol.
  - Problematic in the ingestion process
  - Java-based client ingestion application, to allow ingestions from heterogeneous client platforms.
  - UMIACS has worked with SDSC on the development of SRB's Java client API, but the lack of documentation on the client-server protocol seriously hinders development
  - Discontinuity of protocol



## In Practice

- The Alternative
  - GridFTP appears better suited for the client ingestion process.
    - Robust parallel transfer
    - Protocol in the public domain
    - Existing API
  - Not a perfect world
    - GT is supposed to be modular, but it is unclear how to separate the GridFTP server module from the rest.
    - UMIACS still working on this issue.



## In Practice

- Conclusion
  - Both SRB and GridFTP are very promising technology and have a wide array of potential applications
  - Implementation issues



## In Practice

- The NARA project model
  - Use research institutions for prototypes
    - prove the concept
    - test the protocol
    - an unbiased selection of technology
  - Use commercial vendors for production quality software
    - More experienced professional engineers mean better code quality and UI design
    - Technical support and documentation