

Data Placement in Widely Distributed Environments

Miron Livny
Computer Sciences Department
University of Wisconsin-Madison
miron@cs.wisc.edu

“ ... Since the early days of mankind the primary motivation for the establishment of *communities* has been the idea that by being part of an organized group the capabilities of an individual are improved. The great progress in the area of *inter-computer communication* led to the development of means by which stand-alone processing sub-systems can be integrated into multi-computer *'communities'*. ... ”

Miron Livny, “ *Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems.*”,
Ph.D thesis, July 1983.

High capacity networks are deployed all over the world and almost everyone is concerned about how to allocate their bandwidth. However, is bandwidth the real issue?

Claims for “benefits” provided by Distributed Processing Systems

- High Availability and Reliability
- High System Performance
- Ease of Modular and Incremental Growth
- Automatic Load and Resource Sharing
- Good Response to Temporary Overloads
- Easy Expansion in Capacity and/or Function

“What is a Distributed Data Processing System?” , P.H. Enslow, Computer, January 1978

Benefits to Science

- > **Democratization of Computing** - "you do not have to be a SUPER person to do SUPER computing." (accessibility)
- > **Speculative Science** - "Since the resources are there, lets run it and see what we get." (unbounded computing power)
- > **Function shipping** - "Find the image that has a red car in this 3 TB collection." (computational mobility)

Function Shipping,
Data Shipping,
or maybe simply
Object Shipping?

Managed Data Placement

Management of storage space and bulk data transfers plays a key role in the end-to-end performance of a distributed application:

- Data Placement (DaP) operations must be treated as “first class” jobs and explicitly expressed in the job flow
- Fabric must provide services to manage storage space
- Data Placement schedulers and matchmakers are needed.
- Data Placement and computing must be coordinated
- Smooth transition of CPU-I/O interleaving across software layers
- Error handling and garbage collection

Customer requests:

Place $y = F(x)$ at L!

System delivers.

Simple plan for $y=F(x)\rightarrow L$

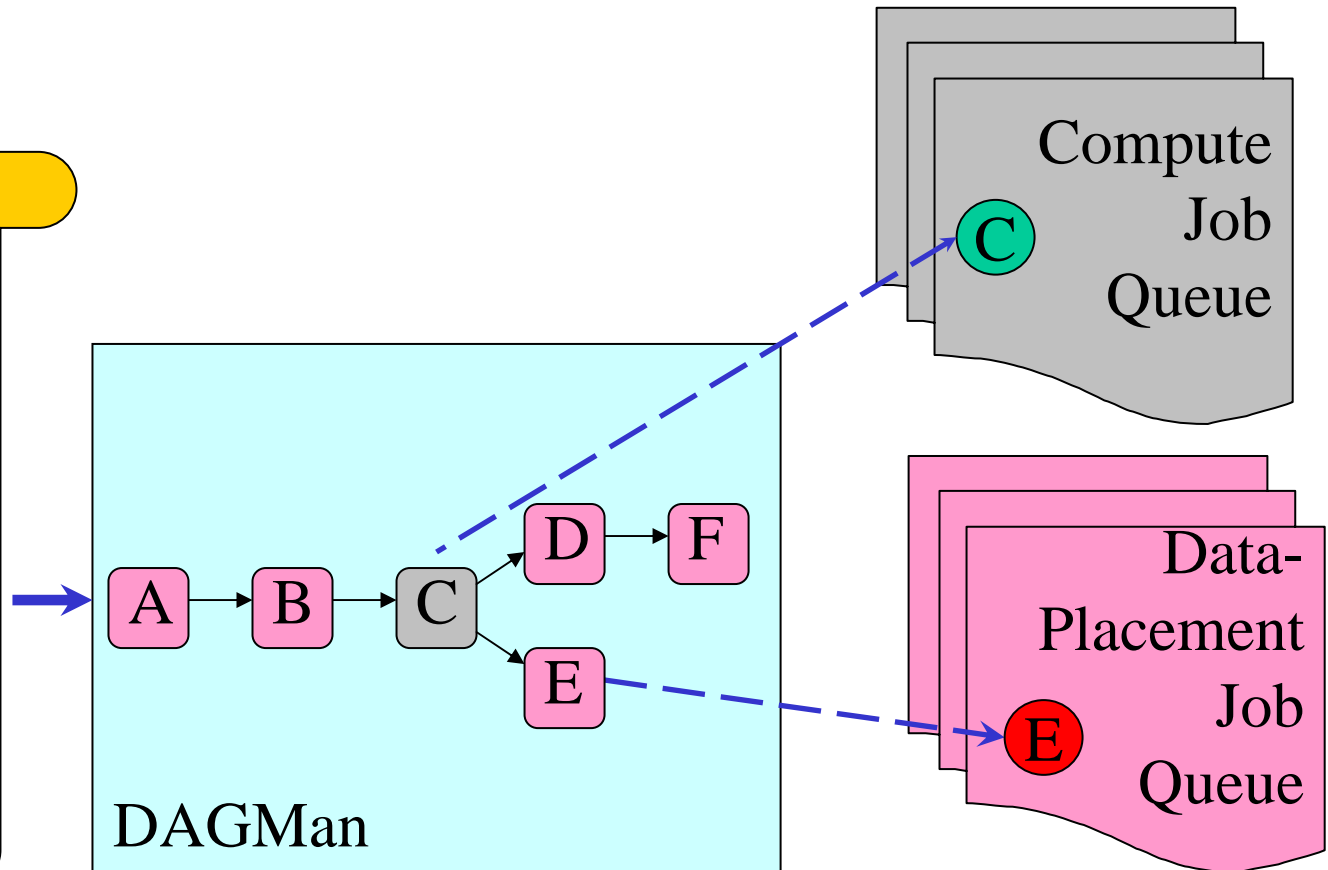
1. Allocate ($\text{size}(x)+\text{size}(y)+\text{size}(F)$) at SE_i
2. Place x from SE_j at SE_i
3. Place F on CE_k
4. Compute $F(x)$ at CE_k
5. Move y from SE_i at L
6. Release allocated space at SE_i

Storage Element (SE); Compute Element (CE)

The Basic Approach*

DAG specification

DaP A A.submit
DaP B B.submit
Job C C.submit
.....
Parent A child B
Parent B child **C**
Parent **C** child D, E
.....

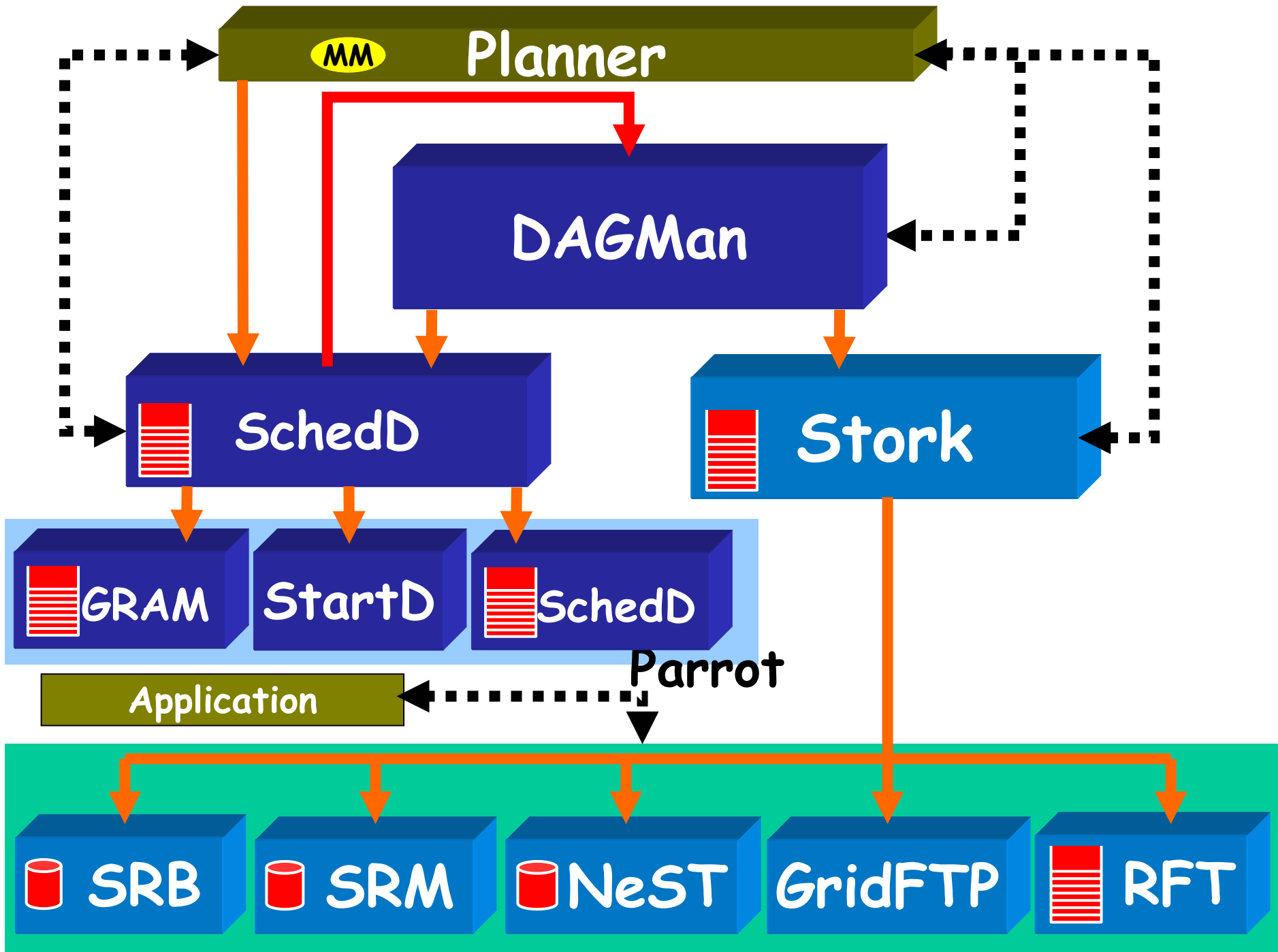


* DAG - Directed Acyclic Graph

Stork

A portable, flexible and extensible Data Placement Scheduler.

- Uses ClassAds to capture jobs and policies (just like Condor)
- Supports matchmaking (just like Condor)
- Provides a suite of DaP jobs that interface with a broad collection of storage systems and protocols and provide end-to-end reliability
- Supports storage allocate/release jobs



Customer requests:

Place *y@S* at L!

System delivers.

Basic step for $y@S \rightarrow L$

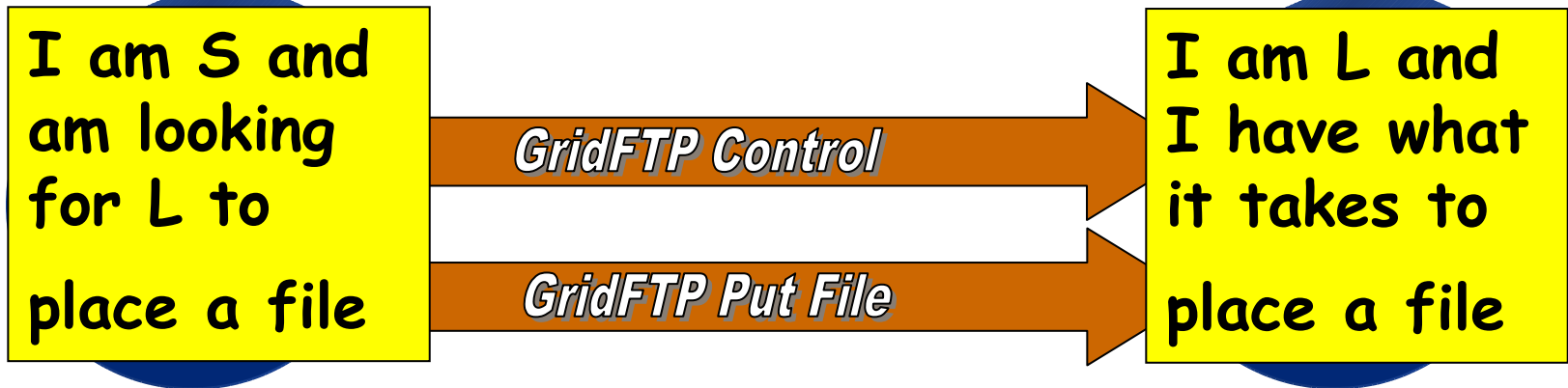
1. Allocate $\text{size}(y)$ at L ,
2. Allocate resources (disk bandwidth, memory, CPU, outgoing network bandwidth) on S
3. Allocate resources (disk bandwidth, memory, CPU, incoming network bandwidth) on L
4. Match S and L

Or in other words,
it takes **TWO** (or more)
to Tango
(or to place data)!

When the
"source" plays "nice"
it "asks" for permission to
place data at "destination"
in advance

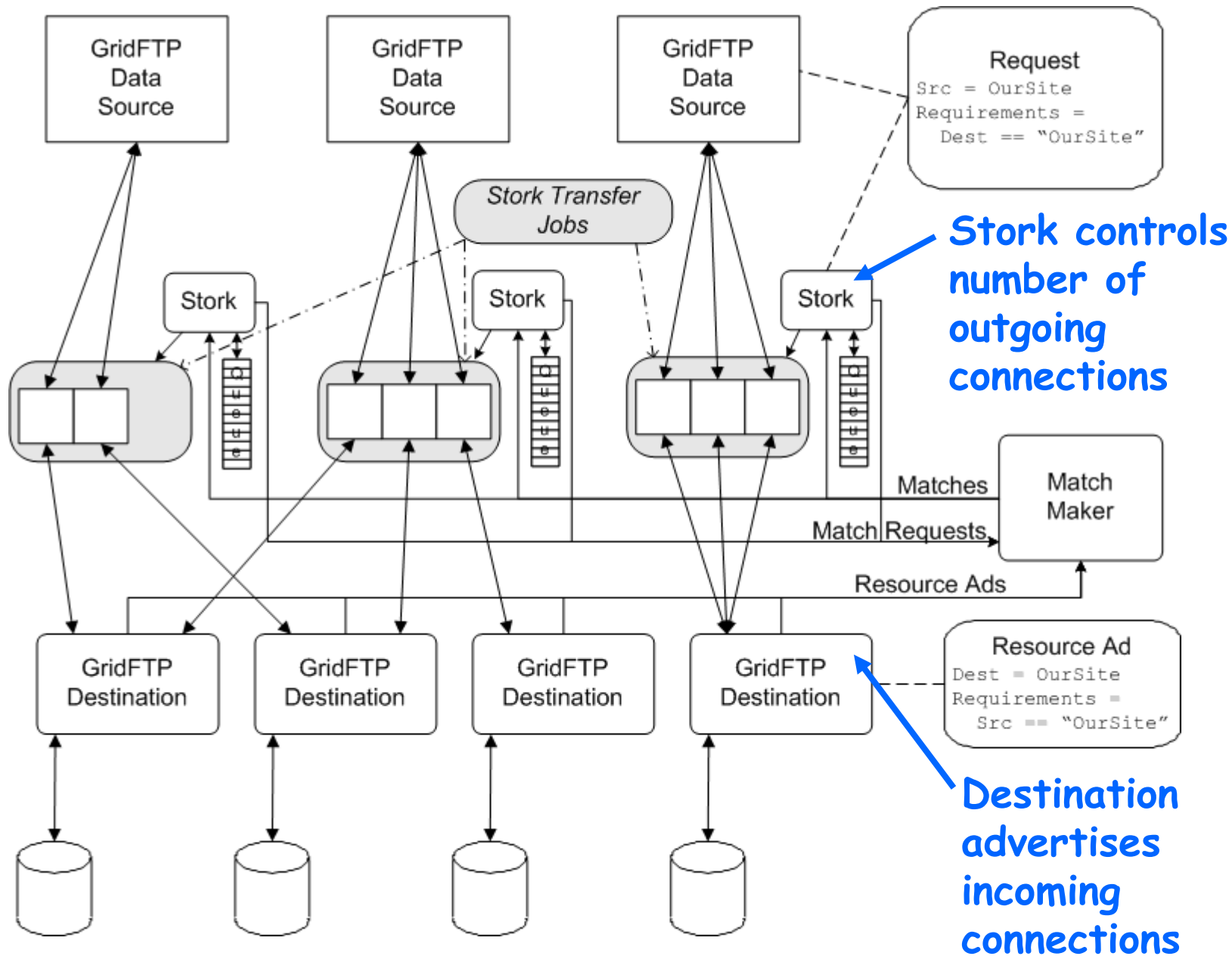
MatchMaker

Match!

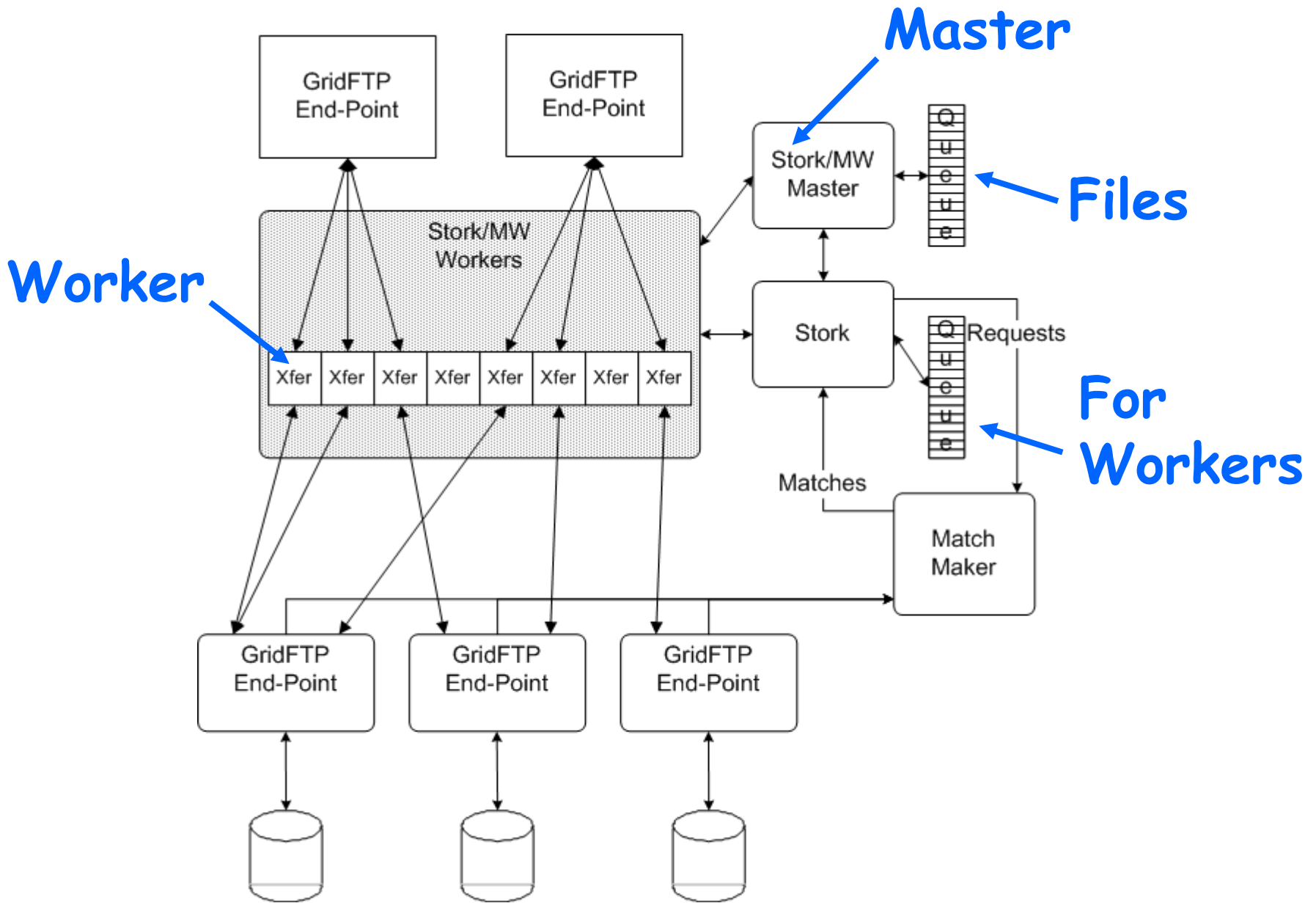


The SC'05 effort

Joint with the
Globus GridFTP team



A Master Worker view of the same effort

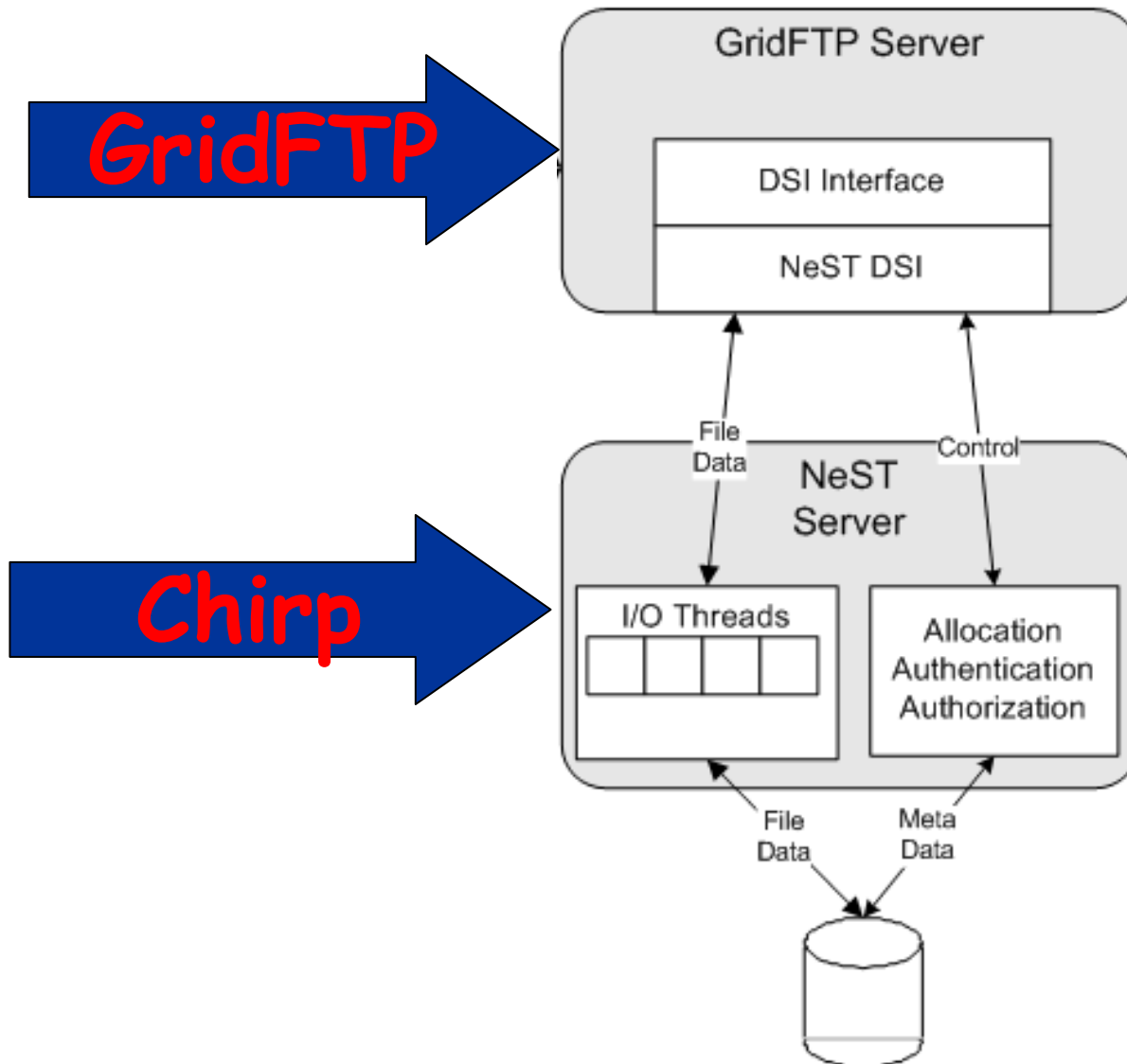


When the
"source" does not
play "nice",
destination must
protect itself

NeST

Manages storage space and connections for a GridFTP server with commands like:

- ADD_NEST_USER
- ADD_USER_TO_LOT
- ATTACH_LOT_TO_FILE
- TERMINATE_LOT



How can we accommodate
an unbounded
amount of data with
an unbounded
amount of storage and
network bandwidth?