

# Remote File Access in GRID

Massimo Donatelli

INFN/CNAF

21/02/2002

# Accesso remoto ai file

- Spesso e' necessario accedere ad una piccola parte di un file senza trasferirlo completamente
- Evito di ricopiare piu' volte lo stesso file
  - Minore occupazione di banda
  - Diminuzione del tempo di trasmissione (I/O Bound)
  - Minore spreco di risorse
  - Non ho problemi di sincronizzazione di versioni diverse dello stesso file
- Voglio poter usare le stesse system call del kernel e vedere il file come se fosse interno al mio sistema ( presente fisicamente )

# Analisi delle varie possibilita'

Dopo un'attenta analisi dei metodi di Remote file access esistenti  
Abbiamo scelto NFSv4, con le seguenti motivazioni:

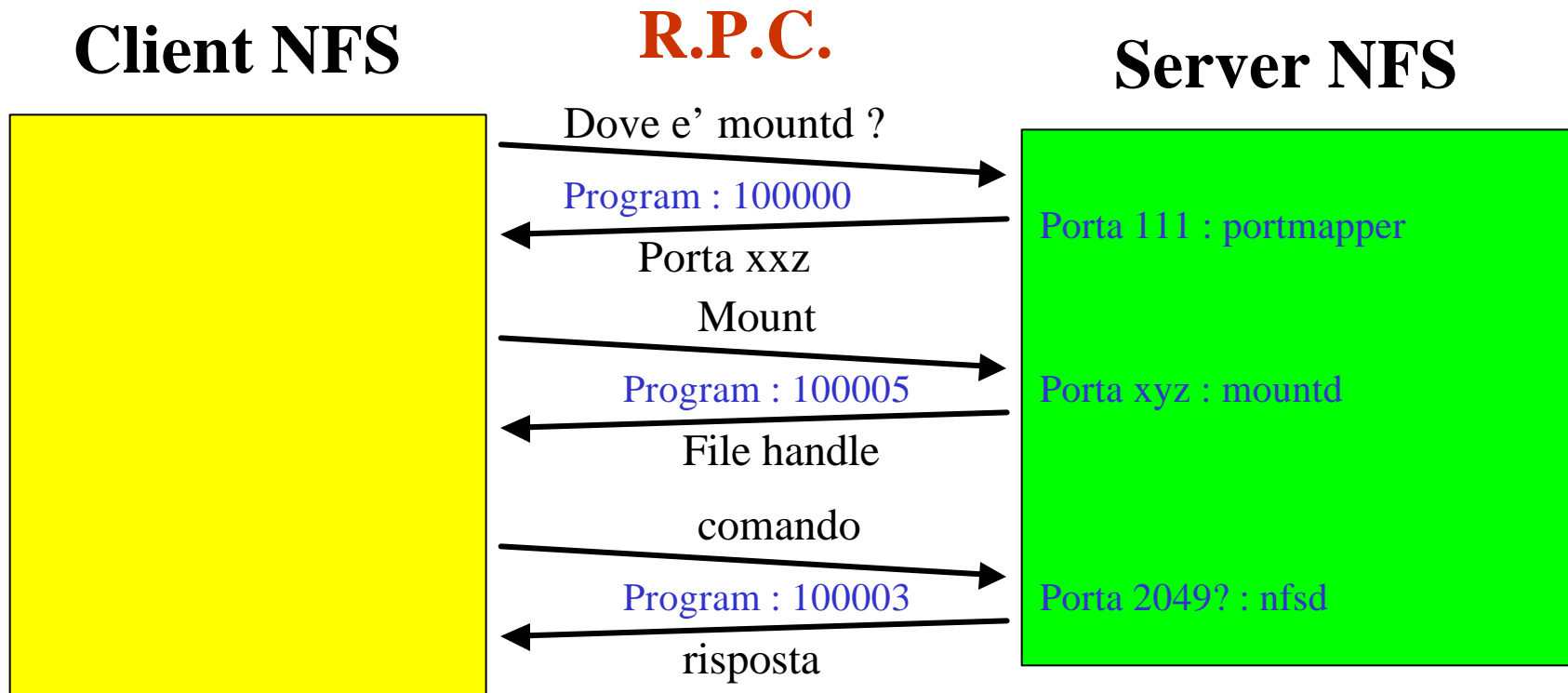
- Protocollo nativo per l'accesso remoto ai file
- Si integra completamente nel VFS permettendo tutte le system call del kernel sui file
- Prevede un meccanismo di autenticazione:
  - Kerberos v.5
  - SPKM (in via di sviluppo)
- Ottimizzato per le reti geografiche (COMPOUND)
- Funzionamento di tipo stateless (OPEN – CLOSE stati provvisori)
- Delega di diverse funzioni al client
- Gestione evoluta del locking dei files

# Continua... NFSv4

Altre caratteristiche di NFSv4:

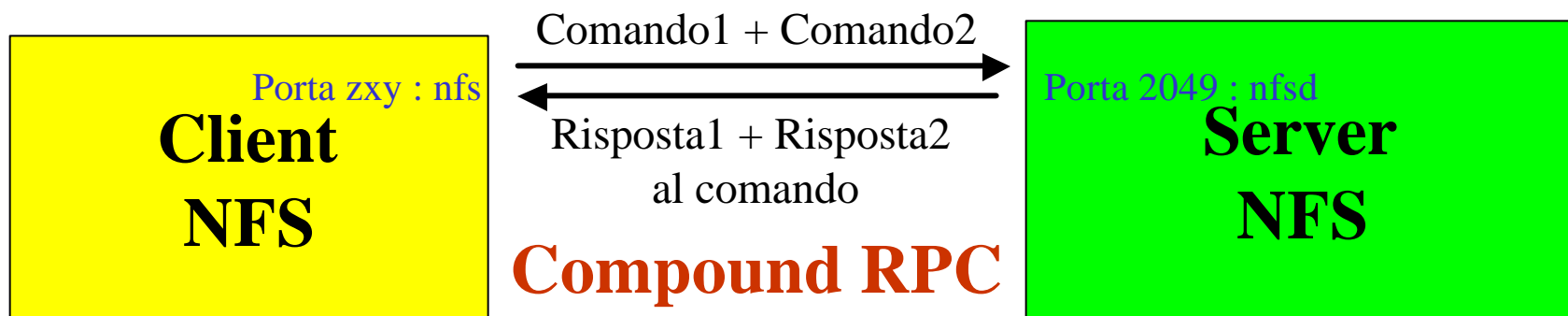
- Gestione avanzata di utenti e gruppi
  - Uso delle ACL
  - Identificatori di utente e gruppo con delle stringhe
  - UserUniversalIdentification (128 bit)
  - Migrazione
  - Replica
- Sicurezza mediante GSS\_API integrata nelle RPC
  - Autenticazione
  - Protezione
  - Integrità

# Il protocollo di mount in NFSv3



# RPC composite in NFSv4

- Tutti i protocolli convergono nella porta 2049 predefinita del server
- In una singola RPC posso gestire diverse system call o comandi
  - La risposta ad ogni comando di una procedura remota puo' essere positiva (comando riuscito correttamente)
  - Oppure la risposta puo' essere un codice di errore specifico per il comando
- Riducono il round trip time
- Aumentano la complessita' nella gestione degli errori



# Come impiegare NFSv4

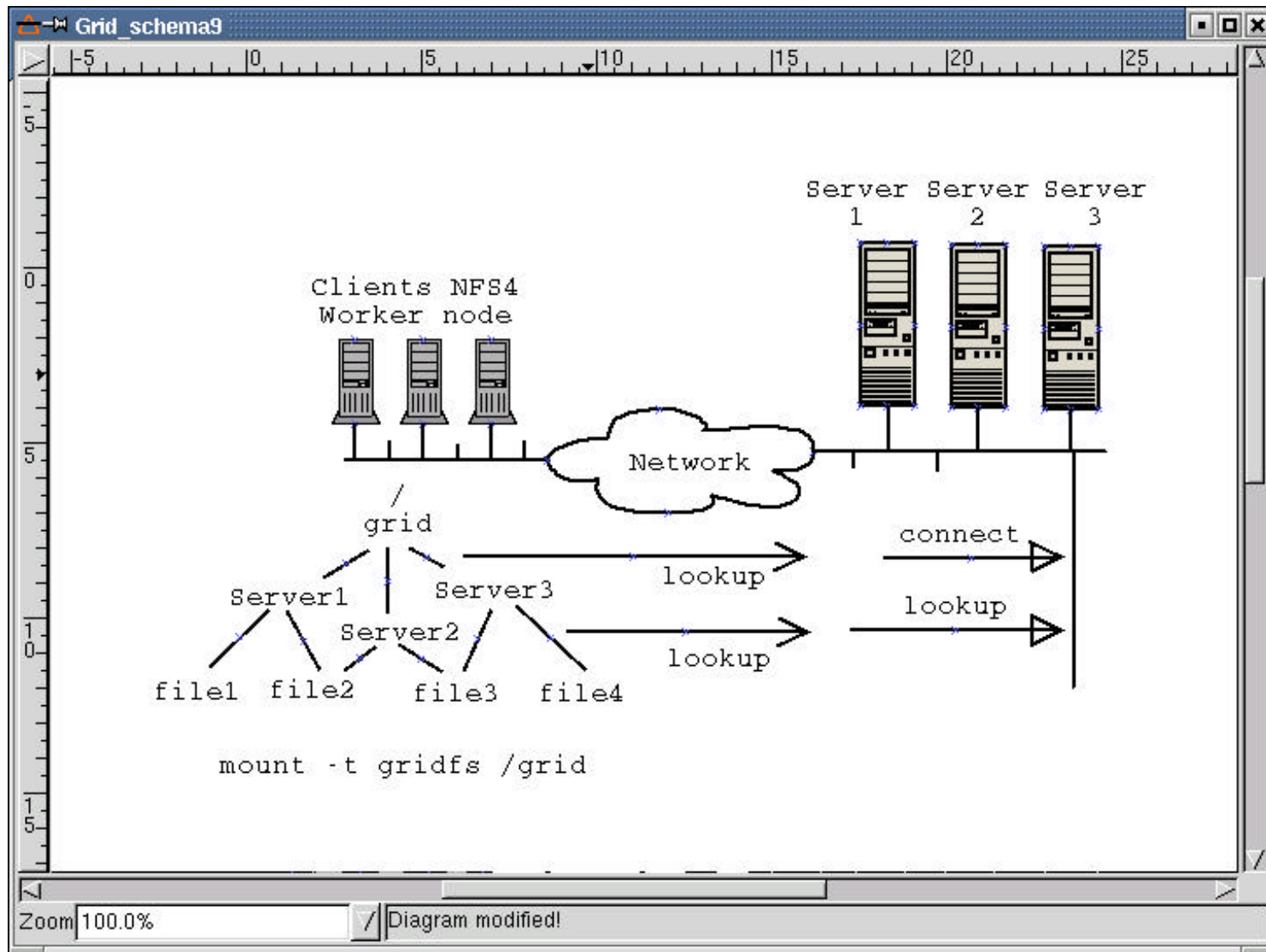
- Introdurre il minimo numero di modifiche
  - Il nuovo protocollo non sara' piu' NFSv4 ma gridfs
- Cosa non vogliamo modificare del protocollo di NFSv4
  - Integrazione nel kernel
  - Compatibilita' con gli standard e le successive versioni
  - Non ci vogliamo preoccupare di tenere aggiornato il protocollo

# Soluzione

- Impiego di un modulo che nasconda l'operazione di mount di NFSv4
  - Il mount e' fatto una volta per tutte nella directory (allo startup della macchina)  
*/grid*
- Il filesystem installato nel client si preoccupera' di aprire e chiudere le connessioni con i vari server
- Il modulo utilizzerà funzionalità (system call) analoghe a quelle presenti nel client NFSv4
- I server saranno di tipo NFSv4 senza alcuna modifica

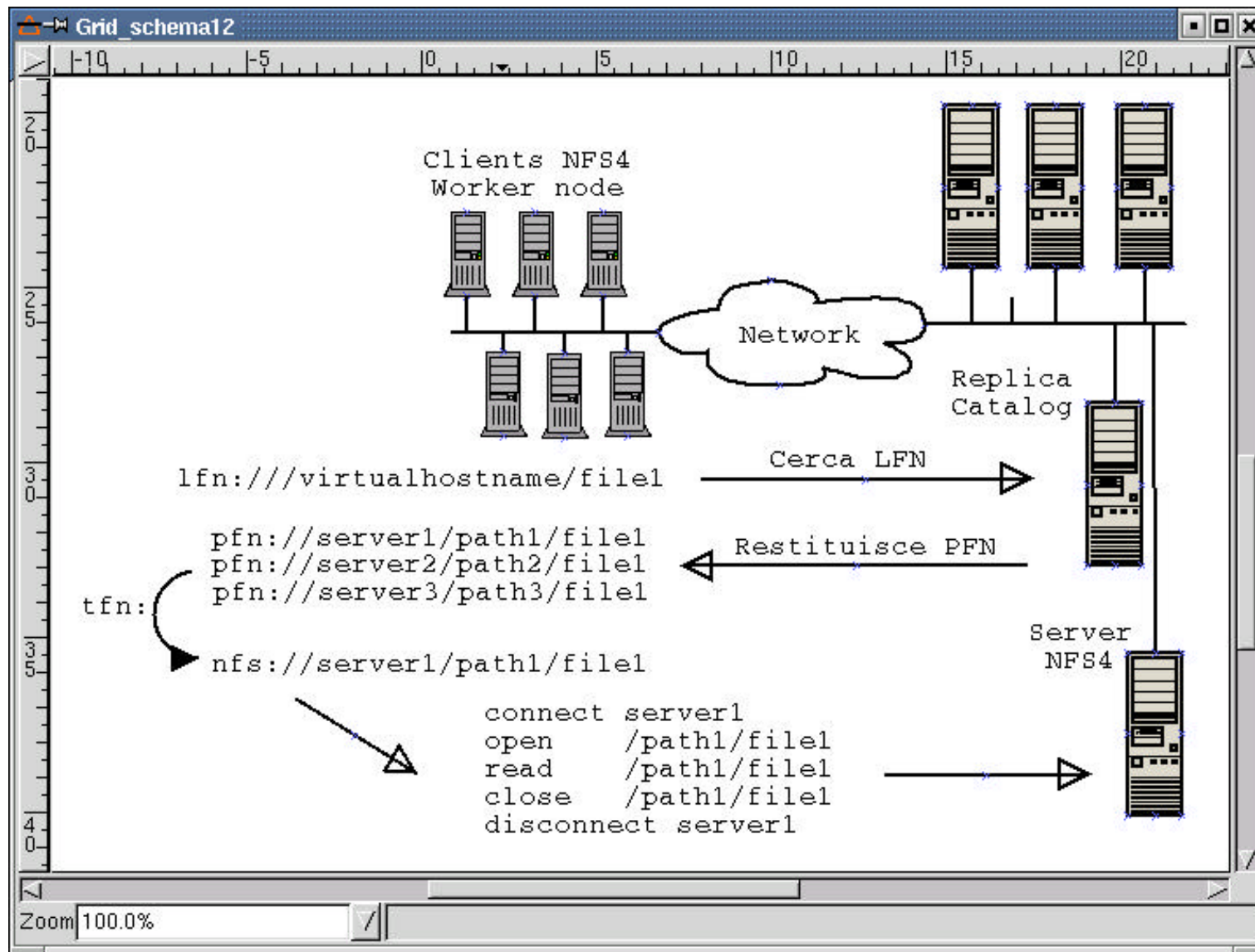


# Sintassi VFS in Unix



# File naming in Datagrid

(Mapping fra la sintassi URL e sintassi VFS)



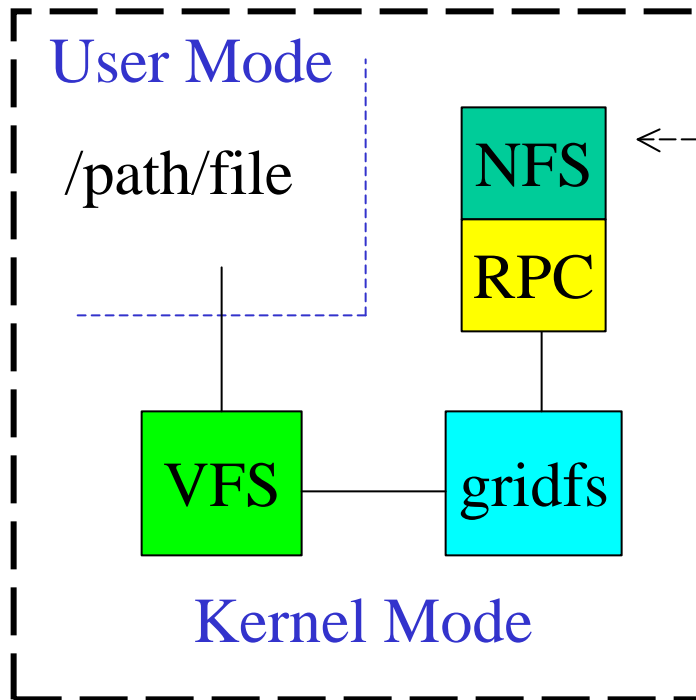
# Possibile implementazione di alcune system call in gridfs

Operazioni:

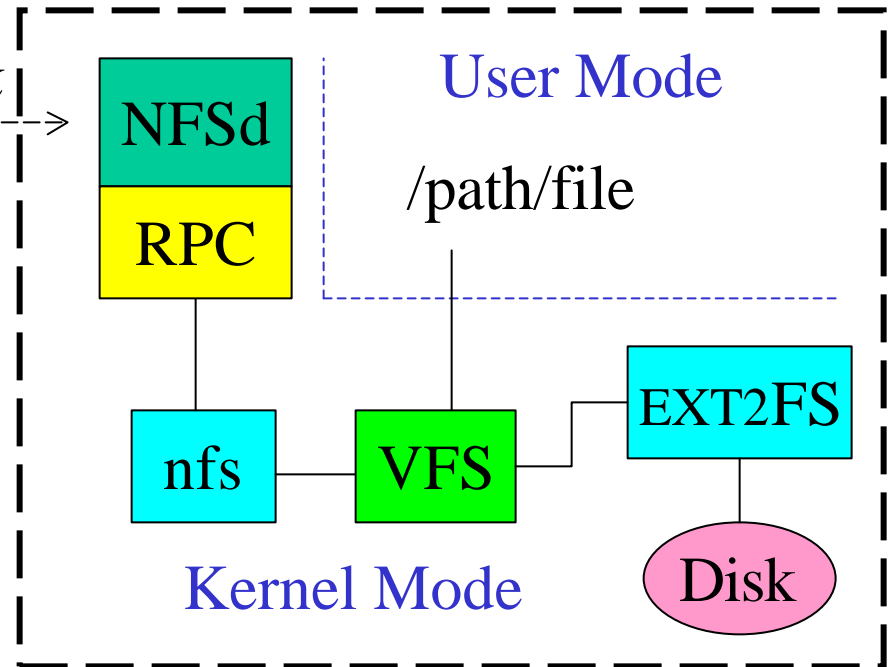
- Lookup /grid/node/path/filename ↗ file handle  
(connect node , lookup /path/filename , disconnect node) ↗ public fh
- Open fh  
(connect node , open fh)
- Close fh  
(close fh , disconnect node)
- Read fh  
(read fh)
- Write fh  
(write fh)

# Schema logico

## Client



## Server



*network*

# Riferimenti utili su NFSv4

Sito ufficiale di NFSv4

[www.nfsv4.org](http://www.nfsv4.org)

Sito di sviluppo di un'implementazione di NFSv4  
dentro il sistema operativo Linux

<http://citi.umich.edu/projects/nfsv4>

Documento divulgativo su NFSv4

<http://www.nluug.nl/events/sane2000/papers/pawlowski.pdf>