

Introduzione al Grid Computing e al Globus Toolkit™

D. Talia - UNICAL

Griglie e Sistemi di Elaborazione Ubiqui

Sommario

- Introduzione al Grid Computing
- Alcune Definizioni
- La Filosofia della Architettura di Griglia
- Il Globus Toolkit (GT2)
 - Introduzione, Security, Resource Management, Information Services, Data Management
- Open Grid Services Architecture (GT4)

D. Talia - UNICAL

Griglie e Sistemi di Elaborazione Ubiqui

2

Il Problema della Griglia

- **Condivisione flessibile, sicura, coordinata** condivisione di risorse tra gruppi dinamici di individui, istituzioni e sistemi.

Da "The Anatomy of the Grid: Enabling Scalable Virtual Organizations"

- **Permettere a comunità reali o virtuali** ("virtual organizations") con obiettivi comuni di condividere risorse distribuite geograficamente - *assumendo l'assenza di...*
 - sito centrale,
 - controllo centrale,
 - completa conoscenza dello stato del sistema,
 - l'esistenza di relazioni affidabili.

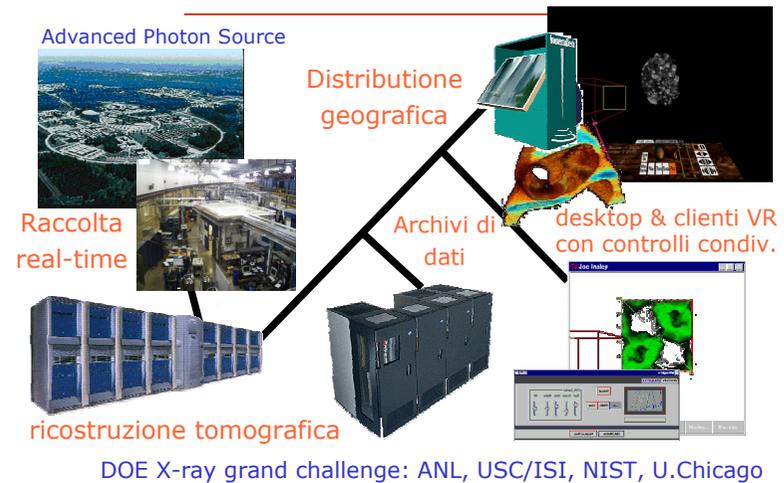
Elementi del Problema

- **Condivisione di risorse**
 - Computer, memorie, sensori, reti, ...
 - Condivisione condizionale: problemi di fiducia, politiche, negoziazione, pagamento, ...
- **Coordinated problem solving**
 - Oltre il client-server: analisi distribuita di dati, elaborazione distribuita, collaborazione, ...
- **Organizzazioni dinamiche, multi-istituzionali e virtuali**
 - Comunità sovrapposte su strutture org. Classiche.
 - Grandi o piccole, statiche o dinamiche.

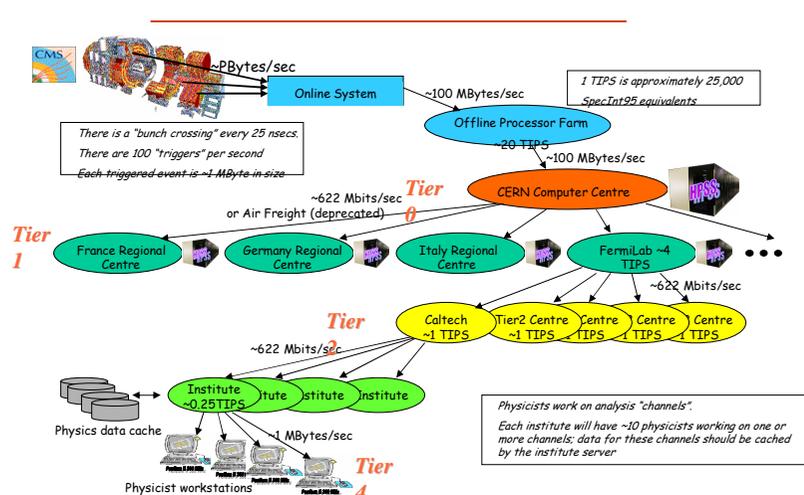
Perché usare le Griglie?

- Un biologo usa 5.000 computer per analizzare 100.000 composti in qualche ora.
- Un migliaio di fisici nel mondo usano in maniera integrata alcuni petabytes di dati.
- Ingegnerici civili collaborano per progettare, realizzare e analizzare esperimenti di terremoti.
- Scienziati del clima visualizzano, annotano, e analizzano terabyte di dati di simulazioni.
- Un team di gestione di emergenze integra dati real time, modelli di previsione del tempo e dati sulla popolazione.
- Una banca effettua analisi finanziarie sui propri conti correnti e sul mercato azionario.

Accesso Online a Strumenti Scientifici



Data Grid per la Fisica delle Alte Energie



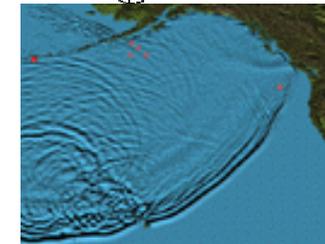
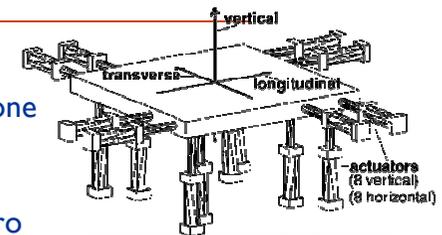
D. Talia - UNICAL

Griglie e Sistemi di Elaborazione Ubiqui

7

Rete per la Simulazione di Terremoti

- NEESgrid: infrastruttura nazionale per la collaborazione tra ingegneri civili e l'uso di dispositivi sperimentali, databases, computers, & altro
- Accesso on-demand ad esperimenti, dati, calcolo, archivi, e collaborazioni.



NEESgrid: Argonne, Michigan, NCSA, UIUC, USC

D. Talia - UNICAL

Griglie e Sistemi di Elaborazione Ubiqui

8

Home Computers Per l'Analisi di Farmaci per l'AIDS

- Comunità=
 - migliaia di utenti di PC
 - Philanthropic computing vendor (Entropia)
 - Gruppo di Ricerca (Scripps)
- Obiettivo Comune= avanzamento nella Ricerca nello studio dell'AIDS

Contesto più Generale

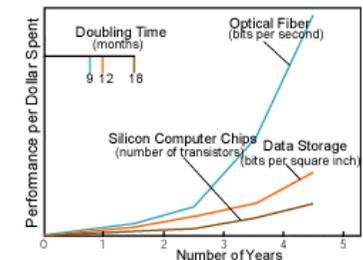
- Il “Grid Computing” ha molto in comune con i maggiori trend industriali
 - Business-to-business, Peer-to-peer, Application Service Providers, Storage Service Providers, Distributed Computing, Internet Computing...
- Problemi comuni non adeguatamente affrontati dalle tecnologie esistenti
 - Requisiti Complicati: “eseguire il programma X sul sito Y conforme alla politica di Comunità P, fornendo l'accesso ai dati in Z secondo la politica Q”
 - High performance: richieste particolari di sistemi avanzati con alte prestazioni.

Perchè Adesso?

- I miglioramenti di legge di Moore nei sistemi di elaborazione produce sistemi finali altamente funzionali.
- Internet e le diverse reti wired o wireless forniscono una connettività globale.
- I cambiamenti nel modo di lavorare in team e orientato alle soluzioni favoriscono questa soluzione.
- Le elevate prestazioni delle reti producono cambiamenti drammatici in termini geometrici e geografici.

Prestazioni Elevate delle Reti

- Prestazioni delle reti e dei calcolatori
 - La velocità dei calcolatori raddoppia ogni 18 mesi
 - La velocità delle reti raddoppia ogni 9 mesi
 - Differenza = un ordine di grandezza ogni 5 anni
- Dal 1986 al 2000
 - Computers: x 500
 - Reti: x 340.000
- Dal 2001 al 2010
 - Computers: x 60
 - Reti: x 4000



[Moore's Law vs. storage improvements vs. optical improvements](#). Graph from *Scientific American* (Jan-2001) by Cleo Vilett, source Vined Khoslan, Kleiner, Caufield and Perkins.

Il Globus Project™

- Stretta collaborazione con progetti “reali” di Griglia nella scienza e nell’industria.
- Sviluppo e promozione dei protocolli standard e delle interfacce di griglia per permettere interoperabilità ed infrastruttura comune
- Il Globus Toolkit™: Open source, software di base di riferimento per la costruzione dell’infrastruttura e le applicazioni di griglia
 - GT2: basato su Java in ambiente Linux
 - GT4: Nuova implementazione basata sui Grid Services (che estendono i Web Services)
- Open Grid Forum: Sviluppo di protocolli standard e API per Grid computing (www.ogf.org)

Alcune Definizioni

Alcune Importanti Definizioni

- Risorsa
- Network protocol
- Network enabled service
- Application Programmer Interface (API)
- Software Development Kit (SDK)
- Sintassi

- Non discusse, ma importanti: politiche

Risorsa

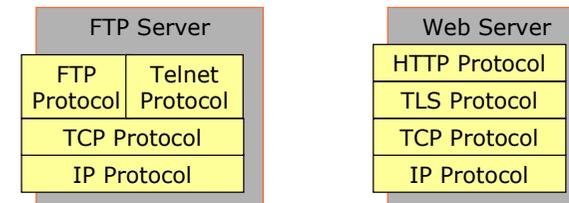
- Una entità da condividere
 - Es., computers, memorie, dati, software
 - Definita in termini di interfacce, non di dispositivi
 - Es. uno scheduler come LSF e PBS definisce una risorsa di calcolo come un cluster
 - Es., Open/close/read/write definiscono accessi ad un file system distribuito come NFS, AFS, Plan9.

Network Protocol

- Una descrizione formale di formati di messaggi e un insieme di regole per lo scambio di messaggi.
 - Le regole possono definire sequenze di scambio di messaggi
 - Un protocollo può definire il cambio di stato nel punto finale, es., cambio di stato di un file system
 - I Protocolli possono prevedere più livelli.
- Esempi di protocolli
 - IP, TCP, TLS (era SSL), HTTP, Kerberos

Network Enabled Services

- Implementazione di un protocollo che definisce un insieme di capabilities
 - Il protocollo definisce l'interazione con il servizio
 - Tutti i servizi di rete richiedono protocolli
 - Non tutti i protocolli sono usati per fornire servizi (es. IP, TLS)
- Esempi: FTP e Web servers



Application Programming Interface

- Una specifica di un insieme di routine per facilitare lo sviluppo di applicazioni
 - Si riferiscono alla definizione, non all'implementazione
 - Es., vi sono implementazioni di MPI
- Specifiche spesso legate ad un linguaggio
 - Nome della routine, numero, ordine e tipo degli argomenti; mapping a costrutti del linguaggio
 - Comportamento o funzione della routine
- Esempi
 - GSS API (security), MPI (message passing)

Software Development Kit

- Una particolare istanziazione di una API
- Un SDK consiste di librerie e strumenti
 - Fornisce una implementazione di una specifica di una API
- Possono esistere diversi SDK per una API
- Esempi di SDK
 - MPICH, Motif Widgets

Sintassi

- Regole per codificare l'informazione, es.
 - XML, Condor ClassAds, Globus RSL
 - X.509 certificate format (RFC 2459)
 - Cryptographic Message Syntax (RFC 2630)
- Distinta dai protocolli
 - Una sintassi può essere usata da molti protocolli (e.g., XML) e utile per molti scopi.
- Si possono avere sintassi a più livelli
 - Es., Condor ClassAds -> XML -> ASCII
 - Importante capire la stratificazione quando si paragonano e valutano più sintassi.

Un Protocollo può avere più API

- Le API TCP/IP includono i sockets BSD, Winsock, System V streams, ...
- Il protocollo fornisce interoperability: programmi che fanno uso di API differenti possono scambiarsi informazioni.
- Io non ho bisogno di conoscere quali API sta usando un utente remoto.



Una API può avere più Protocolli

- MPI fornisce la portabilità: qualsiasi programma corretto compila e “gira” su una piattaforma.
- Non fornisce interoperabilità: tutti i processi devono comunicare tramite la stessa SDK
 - E.g., MPICH and LAM versions of MPI



API e Protocolli sono Entrambi Importanti

- API/SDK standard sono importanti
 - Permettono la *portabilità* delle applicazioni
 - Ma senza protocolli standard, interoperabilità è difficile (ogni SDK parla con ogni protocollo?)
- Protocolli standard sono importanti
 - Permettono *interoperabilità* tra siti diversi
 - Permettono una struttura condivisa
 - Ma senza API/SDK standard, la portabilità delle applicazioni è difficile (macchine differenti fanno uso di un protocollo in maniera differente)

Architettura di Grid

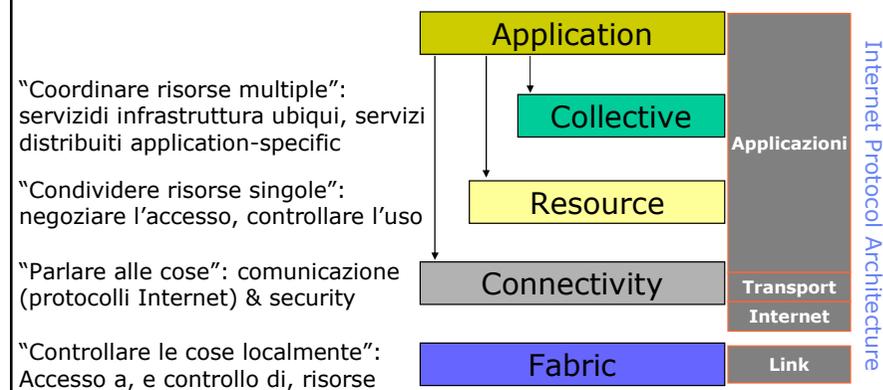
Aspetti del Problema dei Sistemi

- 1) Necessità di interoperabilità quando differenti gruppi condividono risorse
 - Diverse componenti, politiche, meccanismi
 - Es., notioni standard di identità, mezzi di comunicazione, descrizione di risorse
- 2) Necessità servizi di infrastruttura condivisi per evitare sviluppi e configurazioni ripetute
 - Es., una porta/servizio/protocollo per accesso remoto all'elaborazione, non uno per tool/applicazione
 - Es., Autorità di Certificazione : costose
- Necessità comune per protocolli e servizi

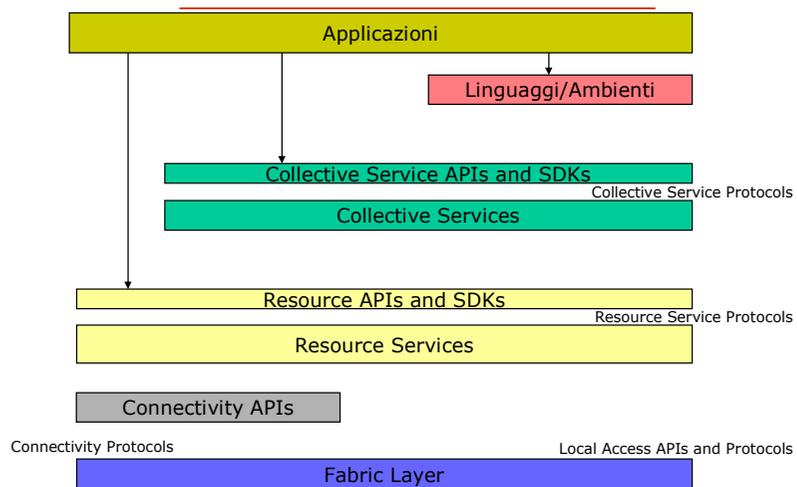
Quindi, una Vista Protocol-Oriented dell'Architettura di Grid orientata a ...

- Sviluppo di protocolli e servizi di Grid
 - Accesso “Protocol-mediated” a risorse remote
 - Nuovi servizi: es., brokering di risorse
 - “On the Grid” = uso di protocolli Intergrid
 - Essenzialmente (estensioni di) protocolli esistenti
- Sviluppo di Grid APIs & SDKs
 - Interfacce a protocolli e servizi di Grid
 - Facilitare lo sviluppo di applicazioni attraverso astrazioni di più alto livello
- Il modello (largamente vincente) è Internet.

Architettura di Grid a Livelli (per analogia con l'Architettura di Internet)



Protocolli, Servizi, e API ad Ogni Livello



Aspetti Importanti

- Costruita sui protocolli e i servizi di Internet
 - Comunicazione, routing, risoluzione dei nomi, ecc.
- “Stratificazione” qui è concettuale, non implica vincoli su chi può chiamare cosa
 - Protocolli/servizi/API/SDK sono, largamente auto-contenuti
 - Alcune cose sono fondamentali: es., comunicazione e sicurezza
 - E' vantaggioso per funzioni di più alto livello usare funzioni comuni di più basso livello.

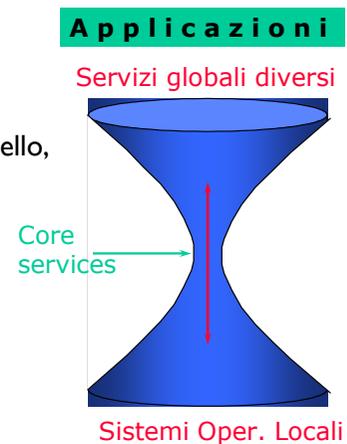
Il Modello a Clessidra

- **Focus sui problemi architetturali**

- Un insieme di core services come infrastruttura di base
- Costruzione di soluzioni di alto livello, domain-specific

- **Principi di progettazione**

- Mantenere basso il costo di partecipazione
- Permettere il controllo locale
- Supportare l'adattamento
- Modello "clessidra IP"



Livello di Connettività Protocolli & Servizi

- **Comunicazione**

- protocolli Internet : IP, DNS, routing, ecc.

- **Sicurezza: Grid Security Infrastructure (GSI)**

- Autenticazione uniforme, autorizzazione, e meccanismi di protezione di messaggi in ambienti multi-istituzioni
- Singola iscrizione, delega, mapping dell'identità
- Tecnologia a Chiave Pubblica, SSL, X.509, GSS-API
- Infrastruttura di Supporto : Certificate Authorities, gestione di certificati & chiavi, ...

GSI: www.gridforum.org/security

Livello di Risorse Protocolli & Servizi

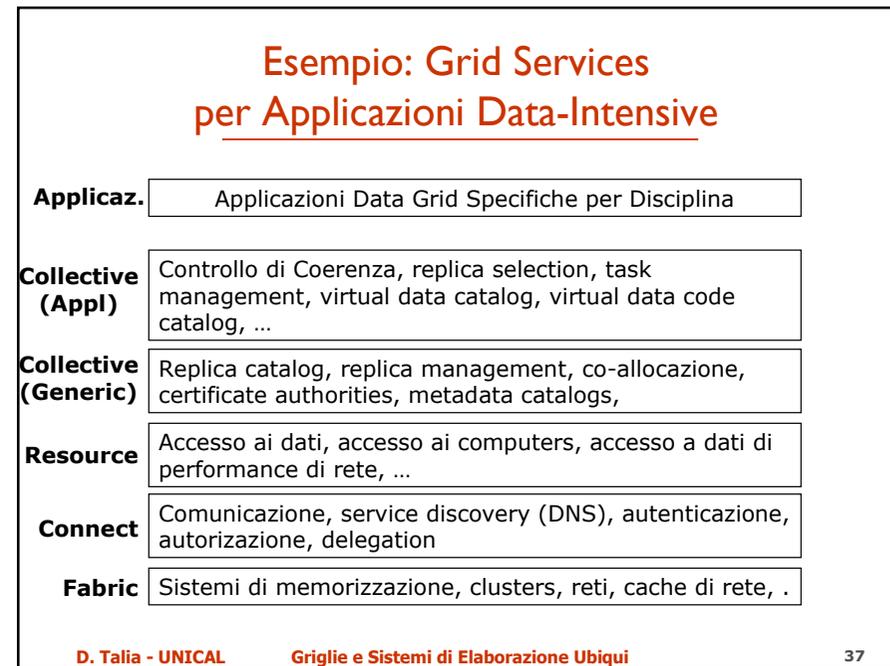
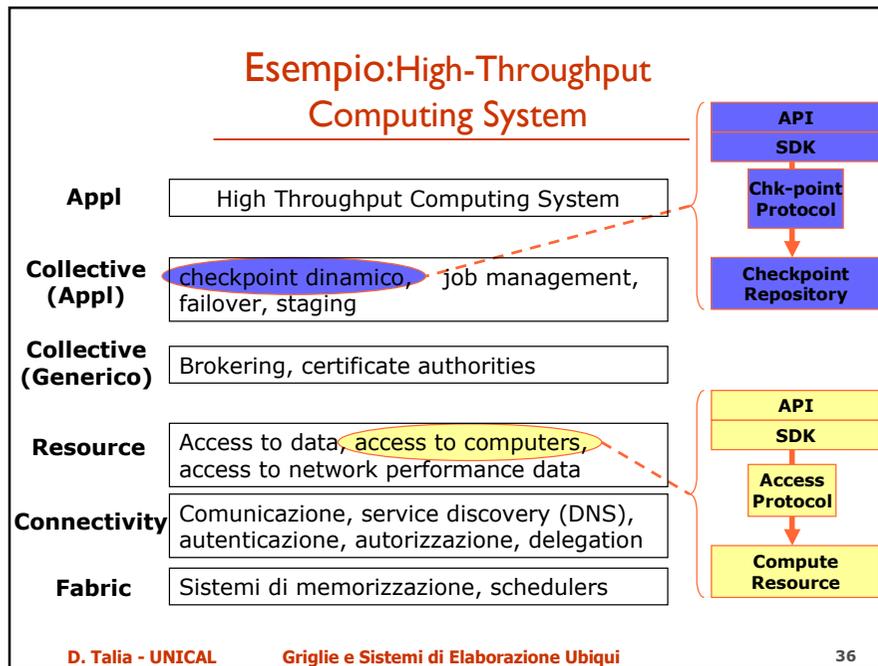
- Grid Resource Allocation Mgmt (GRAM)
 - Allocazione Remota, prenotazione, monitoraggio, controllo delle risorse di calcolo
- Protocollo GridFTP (estensioni FTP)
 - Accesso a dati e trasporto ad alte prestazioni
- Grid Resource Information Service (GRIS)
 - Accesso a informazioni di struttura e di stato
- Network reservation, monitoring, controllo
- Tutto costruito sul livello di connettività: GSI & IP

GridFTP: www.gridforum.org
GRAM, GRIS: www.globus.org

Livello Collective Protocolli & Servizi

- Index server (es. Monitoring and Discovery Service)
 - Viste personalizzate su collezioni di risorse dinamiche assemblate da una comunità
- Resource brokers (e.g., Condor Matchmaker)
 - Scoperta e allocazione di risorse
- Servizi di Replica Location e Management
- Servizi di gestione di Metadati
- Servizi di Co-reservation and co-allocation
- Servizi di Gestione di Workflow
- Ecc.

Condor: www.cs.wisc.edu/condor



Architettura OGSA

