

Grid Scheduling e WS-Agreement

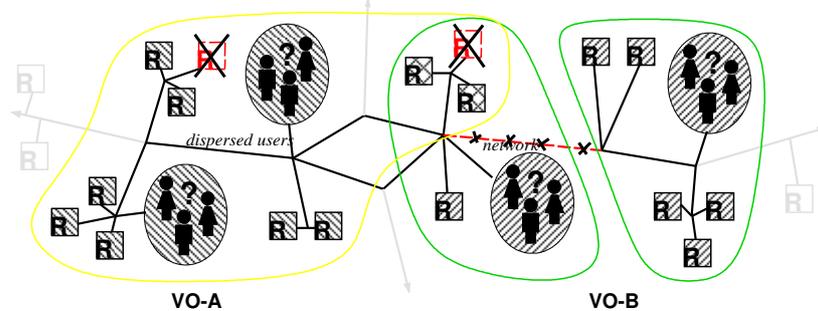
D. Talia - UNICAL

Griglie e Sistemi di Elaborazione Ubiqui

Sommario

- Il Problema del Resource Management
 - Applicazioni in più domini
 - Obiettivi del proprietario vs. obiettivi delle applicazioni
- Una Architettura Aperta per Gestire Risorse
 - WS-Agreement
 - GRAM e Servizi Gestiti

Ambiente di Risorse di Griglia



- Utenti e Risorse Distribuiti
- Stato delle Risorse Variabile
- Raggruppamento e Connettività Variabile
- Scheduling e Politiche Decentralizzate

3

Conflitti Sociali/di Politiche

- **Obiettivi dell'Applicazione**
 - Utenti: scadenze e obiettivi di disponibilità
 - Applicazioni: servono risorse coordinate
- **Obiettivi del proprietario della risorsa**
 - Politiche verso gli utenti
 - Obiettivi di ottimizzazione
- **Gli obiettivi della Comunità Emergono come:**
 - Un aggregato utente/applicazione?
 - Una risorsa virtuale? Entrambi!

4

Scheduling Tradizionale

- **Modello Sistema Chiuso**
 - Assume un proprietario/autorità globale
 - Applicazioni “sandboxed” senza interazioni
 - “Lancia un job oltre la rete e aspetta”
- **Utilizzazione come Metrica Primaria**
 - Code batch profonde permettono un uso più efficiente
 - Nessun incentivo per soddisfare lo scheduling dell’utente
- **Politiche contro il sito**
 - Gli utenti usano trucchi per “farsi gioco” del sito.

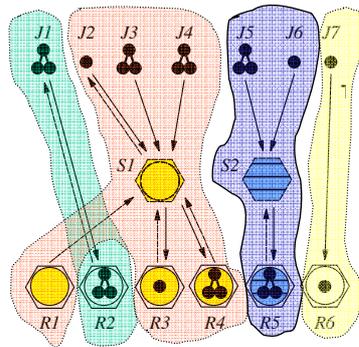
5

Un Modello di Negoziazione Aperto

- **Risorse in un Contesto Globale**
 - Avviso e negoziazione
 - Interfaccia Client remota normalizzata
 - Le risorse mantengono l’autonomia
- **Utenti o Agenti Collegano le Risorse**
 - Gestione della fornitura e sottomissione di task
 - Azioni Coordinate tra domini
- **Mediazione Community-based**
 - Coordinamento per interessi collettivi

6

Esempio di Community Scheduling



- **Utenti individuali**
 - Richiedono servizi
 - Hanno obiettivi applicativi
- **Community schedulers**
 - Servizi di brokering
 - Scheduling Aggregato
- **Risorse individuali**
 - Forniscono servizi
 - Hanno autonomia di politiche
 - Servono i clienti

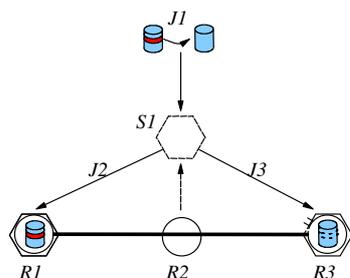
7

Fasi di Negoziazione

- **Discovery**
 - “Quali sono le risorse di interesse?”
 - Trova i service providers
- **Monitoring**
 - “Cosa sta avvenendo con le risorse?”
 - Confronta i service providers
- **Agreement**
 - “Loro mi forniscono quello che serve a me?”
 - **Il Problema centrale del Resource Management**
 - Questo processo può iterare per trovare un adattamento.

8

Estensione di Dominio : File Transfer



- **Singolo obiettivo**
 - Trasferimento con “deadline” sicura
- **Scheduler Specializzato**
 - Servizi semplici di brokering
 - Creazione di nuovi servizi
 - > Logica di gestione di guasti
- **Risorse Distribuite**
 - Spazio di memoria
 - Banda di accesso ai dati
 - Banda di rete

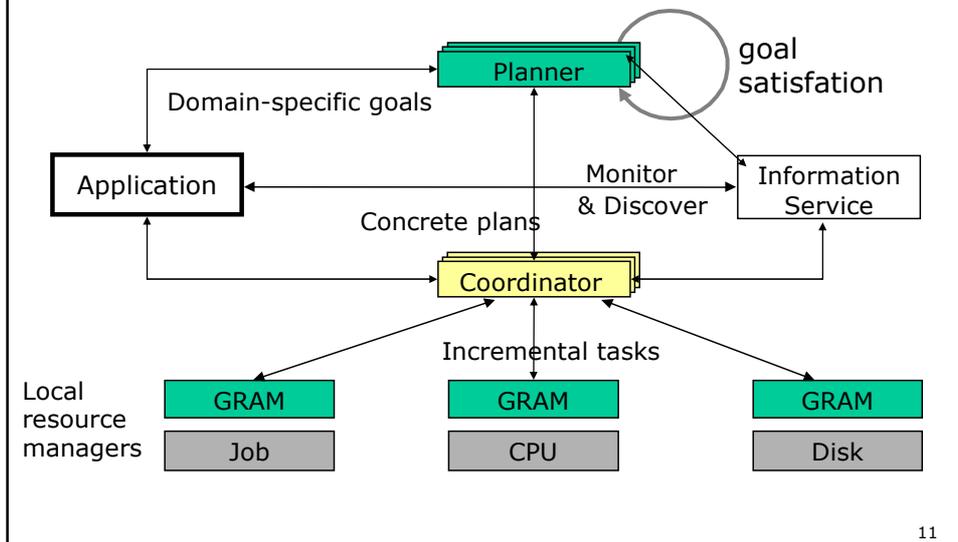
9

Sfide Tecniche

- **Requisiti di Sicurezza Complessi**
 - Diritti, audit, identità, ...
- **Scalabilità Globale**
 - Obiettivi simili ad Internet
 - Infrastruttura Interoperabile
 - Configurabile (su politiche) per ragioni sociali
- **Permanenza o “Evolve in Place”**
 - Non si può mettere tutti off-line per motivi di servizio
 - Nel tempo: upgrade, estendi, adatta
 - Accetta l’eterogeneità

10

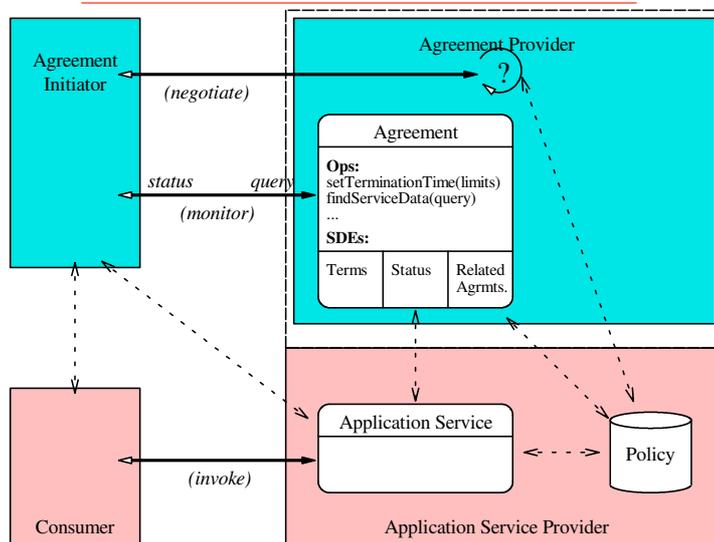
Architettura GRAM



WS-Agreement

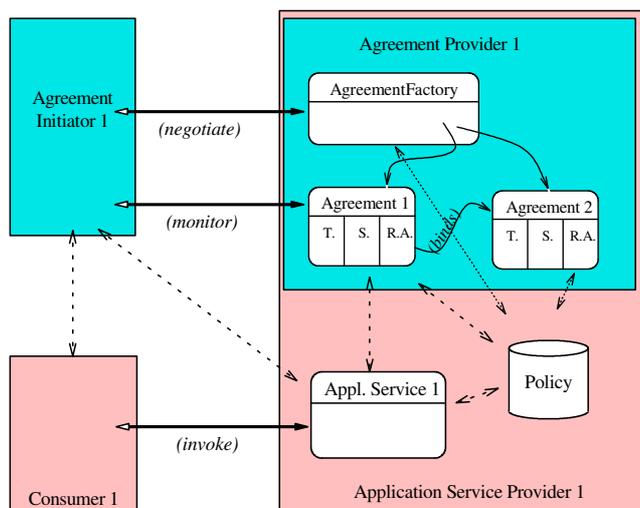
- Nuovo impegno di standardizzazione
 - GRAAP-WG dell'OGF
- Generalizza le idee del GRAM
 - Architettura Service-oriented
 - Una *Risorsa* diventa un *Service Provider*
 - Un *Task* diventa un *Negotiated Service*
 - Lo stato rappresentato come *Agreement Service*

Entità in un WS-Agreement



13

WS-Agreement richiede una Gestione



14

Prenotazione Anticipata di Job

- **Commitment del servizio basato sullo Scheduling**
 - Richiede termini di Agreement basati sullo schedule
- **Pre-Agreement Opzionale**
 - Agreement per facilitare futuri Job Agreement
 - Caratterizza le risorse virtuali necessarie al Job
 - Potrebbe non necessitare di termini orientati al job
- **Job Agreement quasi normale**
 - Può sfruttare il Pre-Agreement
 - > Fa riferimento a promesse esistenti di scheduling di risorse
 - Può ottenere il commitment in un colpo solo
 - > Include direttamente termini di schedule
 - > (Si può pensare come ad una richiesta atomica anticipata)

17

Sistemi Affini

- **Sistemi Accademici recenti**
 - Condor Matchmaking
 - Economy-based Scheduling
 - Workflow Planning
- **Esempi di Scheduler Commerciali**
 - Community Scheduler Framework (CFS)
 - Platform Computing
 - > LSF usato per *molti job*
 - > MultiCluster per condivisione di risorse tra siti

18

Condor Matchmaking

- Essenzialmente: un algoritmo di scheduling
- Euristiche per accoppiare job e risorse
 - Associa annunci simmetrici
 - Ottimo per matching di grandi dimensioni
- Offre una vista a “sistema chiuso”
 - Usa le risorse attraverso il “prestito”
 - Ambiente job sandboxed
 - Favorisce l’integrazione verticale rispetto alla generalità
 - High-throughput system