Corso di **Tecnologie per il cloud**

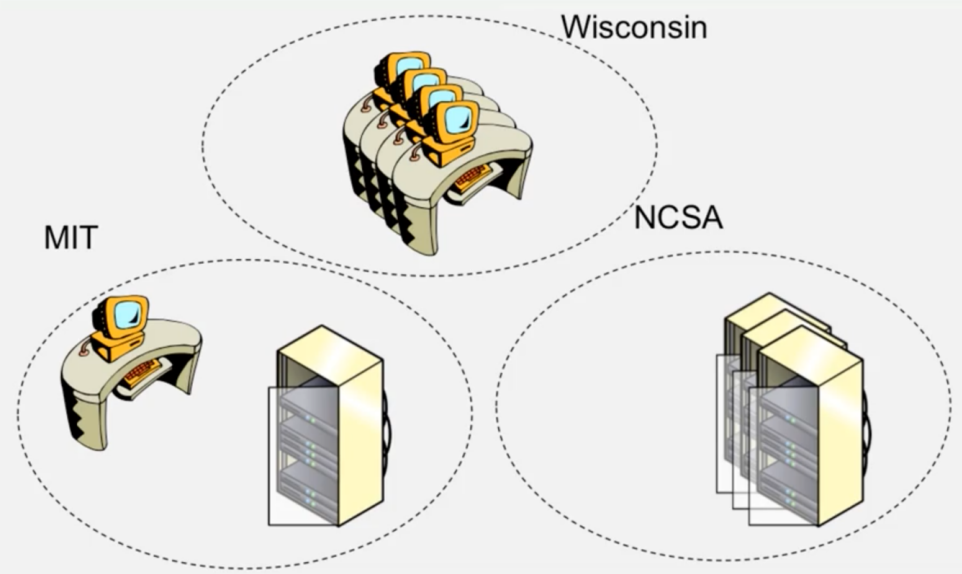
5. Grid applications

(griglia computazionale)

Introduzione

Le **grid applications** o **grid computing** sono un'architettura di elaborazione distribuita che utilizza una rete di computer interconnessi per creare una "griglia" di risorse di elaborazione condivise. Le applicazioni di "grid computing" sfruttano questa architettura distribuita per eseguire compiti complessi in parallelo su un gran numero di nodi di elaborazione. Queste applicazioni sono spesso utilizzate nell'ambito del cloud computing per fornire una maggiore scalabilità e flessibilità alle applicazioni distribuite, consentendo loro di utilizzare in modo efficiente le risorse di elaborazione disponibili sulla griglia. Ad esempio, un'applicazione di "grid computing" potrebbe essere utilizzata per eseguire un'analisi su una grande quantità di dati, suddividendo il lavoro in piccole unità di elaborazione che possono essere distribuite su diversi nodi di elaborazione all'interno della griglia.

Questo particolare tipo di applicazioni è noto anche come **HPC (High Performance Computing)** e sono impiegate per calcoli intensivi, che richiedono anche un numero esiguo di processori e risorse hardware in generale.



Abbiamo parlato di "grid applications", ma il cloud computing non si basa solo su di esse; Infatti il cloud computing comprende anche altre tecnologie, come ad esempio la virtualizzazione, che consente di creare ambienti di elaborazione virtuali sulle risorse fisiche disponibili, e l'automazione, che consente di gestire le risorse di elaborazione in modo efficiente e scalabile.

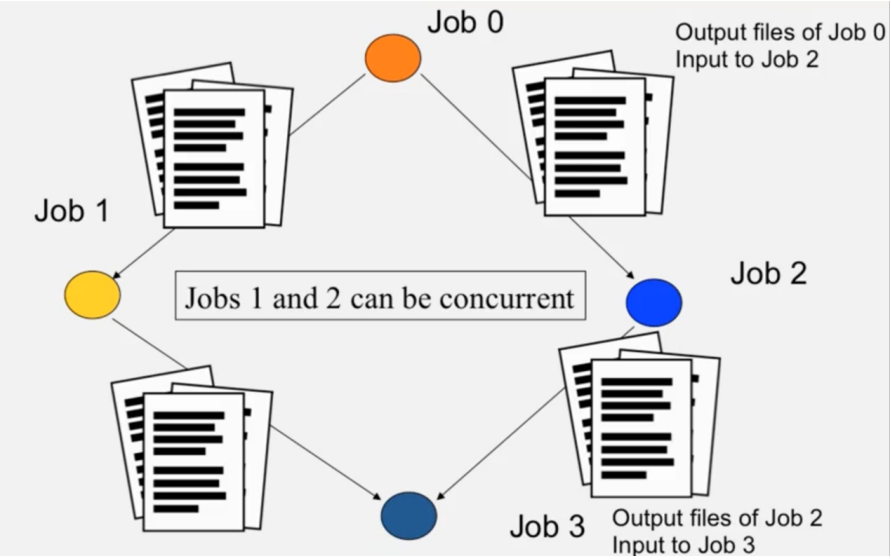
Inoltre, il cloud computing si basa anche su servizi web, come ad esempio i servizi di archiviazione dati e i servizi di elaborazione dati in remoto, che vengono forniti tramite Internet.

In sintesi, le "grid applications" sono una delle tecnologie utilizzate nel cloud computing, ma non rappresentano l'unico elemento su cui si basa questa tecnologia.

IVAN SOLLAZZO - ANDREA PISCOPO - NICOLAS BRUNO
PROFESSORE PIERLUIGI GALLO - INGEGNERIA CIBERNETICA
UNIVERSITA' DEGLI STUDI DI PALERMO

Concetto di job

Un'applicazione può essere costituita da una serie di job, può sfruttare come input l'output di un altro job, fino al completamento degli stessi.



Questi job possono essere inoltre eseguiti in postazioni geografiche differenti, ad esempio:

- Il **job 0** nel data center dell'**NCSA** e il **job 1** in quello del **MIT**;
- Il **job 1** e il **job 2**, essendo indipendenti, possono essere eseguiti in **parallelo** e trovarsi anch'essi in posti geografici differenti.

I dati che viaggiano da un job all'altro sono nell'ordine dei gigabyte e possono richiedere anche giorni per un trasferimento completo.

Ci sono sostanzialmente quattro fasi per un job:

1. **Inizializzazione:** inizializza il job;
2. **Stage in:** raccogli l'output del job precedente;
3. **Esecuzione:** esegue il job;
4. **Stage out:** genera il nuovo output e lo manda al prossimo job;
5. **Pubblicazione (facoltativo):** rende disponibile i risultati immediati del job trascorso.

Ciascun job può essere inoltre suddiviso in diversi **task** per favorire la **parallelizzazione**, considerando il grande quantitativo di dati da elaborare. Tali task possono essere eseguiti anche su più macchine nella rete distribuita.

Schedulazione dei job

La **schedulazione dei job** è un'importante funzionalità delle **grid applications** che consente di suddividere il lavoro in unità di elaborazione più piccole e distribuirle su diversi nodi di elaborazione all'interno della griglia. La schedulazione dei job è fondamentale per garantire che le risorse di elaborazione vengano utilizzate in modo efficiente e che i compiti siano completati nel minor tempo possibile.

La schedulazione dei job può essere realizzata mediante diversi algoritmi di scheduling, come ad esempio l'algoritmo **FCFS** (First-Come, First-Served) che assegna i job in ordine di arrivo, l'algoritmo **SJF** (Shortest-Job-First) che assegna i job con il tempo di esecuzione più breve, o l'algoritmo Round Robin che assegna i job a turni.

Inoltre, le **grid applications** possono utilizzare anche tecniche di scheduling più avanzate, come ad esempio la pianificazione dinamica, che consente di modificare la priorità dei job in base al carico di lavoro della griglia o alle prestazioni dei nodi di elaborazione.

La scelta dell'algoritmo di scheduling dipende dalle esigenze dell'applicazione e dalle caratteristiche della griglia, come ad esempio la disponibilità delle risorse di elaborazione e il tempo di risposta richiesto per i job.

HTCondor e Globus

HTCondor e **Globus** sono due software open source spesso utilizzati insieme per creare grid applications.

- **HTCondor** è un **gestore di job** che può eseguire lavori su una varietà di risorse di elaborazione.

- **Globus** fornisce un set di strumenti per il **trasferimento** sicuro dei dati e la **ricerca e individuazione** di **risorse** che possono essere utilizzate da una grid application, ciò include la ricerca di **computer**, di **archiviazione** e di altre risorse che sono disponibili e possono essere accessibili dall'applicazione

HTCondor viene usato **per inviare job** alla grid di Globus il quale trasferirà i lavori alle risorse appropriate per eseguirli.

HTCondor e Globus possono essere utilizzati per eseguire delle simulazioni su larga scala che richiedono l'accesso a più risorse di elaborazione, oppure è possibile utilizzarli per creare un sistema di analisi dei dati distribuito che può accedere a dati provenienti da più fonti.

HTCondor e Globus sono strumenti che possono aiutare a creare applicazioni grid efficienti e scalabili. Se si sta cercando un modo per sfruttare la potenza di più risorse di elaborazione, allora HTCondor e Globus sono un'ottima opzione.

IVAN SOLLAZZO - ANDREA PISCOPO - NICOLAS BRUNO
PROFESSORE PIERLUIGI GALLO - INGEGNERIA CIBERNETICA
UNIVERSITA' DEGLI STUDI DI PALERMO

Mappa Capitolo: