Condor-G: Un Agente per la Gestione dell'Elaborazione in Multi-Institutional Grids

James Frey, Todd Tannenbaum, Miron Livny,
Ian Foster, Steven Tuecke

Condor-G

- Sfrutta:
  - Security, comunicazioni, resource discovery, accesso a risorse in ambienti multi-dominio offerti dal Globus Toolkit
  - Gestione dell'elaborazione e raccolta di risorse in un singolo dominio amministrativo forniti da Condor
- Condor-G:
  - Permette agli utenti di integrare risorse appartenenti a più domini come se appartenessero ad un unico dominio personale.

Sfide per Costruire e Gestire una Computazione Multi-Site

- Siti differenti hanno differenti
  - Politiche di security e di uso delle risorse
  - Schedulers
  - Hardware
  - Sistemi operativi
  - File system
- Utenti possono avere una conoscenza limitata delle risorse in altri siti
- Fallimenti di possono verificare su siti remoti
- Il monitoring è complesso

Approccio Condor-G

- Separazione dei diversi aspetti

1. Accesso a Risorse Remote
   - Richiede che le risorse remote usino protocolli standard per la gestione di risorse remote
   - **Uso di protocolli definiti dal Globus Toolkit**
2. Gestione dell'elaborazione
   - Introduce agenti per la gestione dei processi utente
   - Responsabile per: resource discovery, job submission, job management, error recovery
   - **Uso del sistema Condor**
3. Ambiente di esecuzione remota
   - Uso della tecnologia di sandboxing per creare un ambiente di esecuzione su un nodo remoto
   - **Uso del sistema Condor**

Protocolli di Grid per l'Accesso a Risorse Remote

- GSI (Security)
- GRAM (sottomissione remota di richieste di elabor.)
  - Security
  - Two-phase commit (aggiunto dal Condor team)
    - > Fornisce una semantica di esecuzione exactly-once
    - > Le richieste di risorse dei client includono numeri di sequenza
    - > Il client riceve risposta da una risorsa e invia un messaggio di commit per indicare che l'esecuzione può iniziare
  - Fault tolerance (aggiunto dal Condor team)
    - > Il GRAM memorizza informazione sui job attivi in memoria stabile sul nodo client
    - > Accede all'informazione se il GRAM server va in crash e riparte
- MDS-2 (protocolli GRRP e GRIP)
- GASS (Global Access to Secondary Storage)
  - Obsoleto, sostituito da GridFTP

Gestione dell'Elaborazione: il Condor-G Agent

- Interfaccia utente
  - API e strumento a linee di comando
  - Sottomette jobs
  - Interroga lo stato dei job
  - Cancella job
  - Informa della termination dei job dei problemi attraverso callbacks
  - Gestisce i log dei job

Gestione dell'Elaborazione: il Condor-G Agent

- Supporta l'esecuzione remota
- L'Agente esegue applicazioni su risorse di nodi remoti per conto di utenti
  - Gestisce lo standard I/O e gli eseguibili dei job usando il GridFTP
  - Sottomette un job su un nodo remoto usando il GRAM
  - Monitoring dei job dei fallimentio remoti via GRAM
  - Autenticazione di tutte le richieste usando il GSI
  - Riesegue job falliti
  - Comunica con l'utente in caso di errori
  - Memorizza lo stato della computazione su memoria stabile per supportare il restart in caso di fallimento di un agent

Implementazione del Condor-G Agent

1. Lo **Scheduler** risponde alla richiesta di un utente
2. Crea un nuovo daemon **GridManager** per eseguire e gestire job
   - Un processo gestisce tutti i job di un singolo utente
   - Termina quando tutti i job sono completati
3. Ogni richiesta di esecuzione di job al **GridManager** crea un daemon Globus **JobManager**
   - Un Gatekeeper per sito remoto, un JobManager per sottomissione di job
   - I JobManager comunicano con il GridManager per trasferire gli eseguibili di job e per l' I/O
4. Il **JobManager** sottomette i job per l'esecuzione allo scheduler locale del sito
5. Aggiornamenti sullo stato dei job sono inviati dal **JobManager** al **GridManager** e quindi allo **Scheduler** Condor-G

Fallimenti Tollerati da Condor-G

1. Crash del Globus JobManager
2. Crash della macchina che gestisce la risorsa remota (es., il GateKeeper e/o il JobManager)
3. Crash della macchina su cui è in esecuzione il GridManager (o crash del GridManager)
4. Fallimenti nei collegamenti di rete tra le macchine coinvolte

Failure detection (rilevazione dei guasti)

- Rilevati dal GridManager, che periodicamente controlla tutti i JobManagers
- Se il JobManager non risponde, controlla il Gatekeeper
  - Se il Gatekeeper risponde, saprà che il JobManager è fallito
  - altrimenti, saprà che il sito remoto ha avuto un crash OPPURE la rete si è disconnessa.

Recupero dai Fallimenti

- Se il JobManager è fallito
  - GridManager tenta di eseguire un nuovo JobManager
- Se non c'è contatto con il nodo remoto, il GridManager attende fino a poter ristabilire il contatto
  - Quindi tenta di connettersi al JobManager
    - > Il JobManager può essere terminato normalmente
  - Se non riesce a connettersi al JobManager, crea un nuovo JobManager
    - > Il JobManager controllerà i job o potrà comunicare al GridManager che i job sono terminati con successo
- Per proteggersi in caso di crash locale, lo stato dei job è memorizzato in modo persistente nella coda dei job dello scheduler Condor-G
  - GridManager riparte dopo un crash locale
  - Reconnettersi a qualsiasi JobManager in esecuzione al momento del crash

Condor-G Agent e Gestione delle Credenziali

- L'agente Condor-G usa Credenziali Proxy del GSI per autenticare l'utente presso le risorse remote
- Credenziali Short-lived
- Le applicazioni Condor-G con durate lunghe devono gestire la terminazione delle credenziali
- L'agente Condor-G periodicamente analizza le credenziali degli utenti con job in esecuzione
- Gli utenti sono informati della prossima scadenza delle credenziali
- Le credenziali vanno rinnovate su ogni sito coinvolto
- MyProxy: permette agli utenti di memorizzare credenziali a lunga scadenza su un server sicuro
  - Servizi remoti possono ottenere per conto dell'utente short-lived proxy da MyProxy
  - Condor-G rinnova le credenziali dal MyProxy server

Resource Discovery e Scheduling

- Come l'agente Condor-G determina dove exeguire i job?
- Implementazione Iniziale : lista dei GRAM server fornita dall'utente
- Soluzione più complessa: un resource broker che combina informazioni come: autorizzazioni utente, requisiti dell'applicazione, stato delle risorse (dall'MDS)
- Condor Matchmaker
  - Descrive le risorse e i requisiti dei job usando i "Classified Ads"
  - Matchmaker trova ClassAds compatibili

Meccanismo Glide-In

- Cosa accade quando un job viene eseguito su un sito remoto dove
  - i file richiesti non sono disponili
  - La politica locale non permette l'accesso ad un file system locale
  - La politica locale impone restrizioni sui tempi di esecuzione dei job

- Mobile sandboxing
  - Esegue su un computer remoto un processo daemon che:
  - Informa il Condor Collector, che formisce informazioni sulle risorse disponibili allo scheduler
  - Associa il job alle risorse pubblicate dai daemons e li esegue sui nodi remoti
  - Esegue ogni task utente in un "sandbox" usando mecchanismi di trap delle system call per ridirigere le system calls eseguite da un task verso il sistema di origine (migliore portabilità e protezione del sistema locale)

Meccanismo Glide-In

- Migrazione di un job in un altro nodo se richiesto
- Queste funzioni sono simili in tutti i computer che participano in un Condor pool
- In Condor-G, i processi daemon sono esguiti dal GRAM invece che dall'utente
- Il Condor-G Glide-in usa protocolli di Grid per creare dinamicamente un Condor pool personale dalle risorse di Grid usando i daemon Condor sulle risorse remote
- Implementazione:
  - Il GlideIn eseguibile iniziale è uno shell script portabile
  - Usa GridFTP per scaricare gli eseguibili Condor da un repository centrale, così gli utenti non devono memorizzare eseguibili per tutte le architetture