



CONTRATTO DI RICERCA ENEA-CASPUR – CRESCO SPI.2

ANALISI E REALIZZAZIONE DI TOOL INNOVATIVI A SUPPORTO DELLE FUNZIONALITA GRID



OR2 – Ampliamento del sistema di delegazione dei diritti di amministrazione del sistema

Alessio Rocchi

Sommario

1. Introduzione	3
2. ARCHITETTURA DEL SISTEMA	4
2.1 – Overview	4
2.2 – Funzionalità dell'applicazione	5
3 – Dettagli sull'implementazione	6
3.1 – Connessioni remote.....	6
3.2 – Database di ARCO	6
3.2.1 – Tabella User	7
3.2.2 – Tabella Operations.....	7
3.2.3 - Tabella Machines.....	7
3.2.4 – Tabella Actions.....	9
3.2.5 – Tabella association	9

1. Introduzione

Oggetto del presente documento è la descrizione di ARCO (acronimo di AFS Remote Command Operator), un software realizzato in linea con l'attività OR2 (Ampliamento del sistema di delegazione dei diritti di amministrazione del sistema) del contratto di ricerca tra CASPUR e ENEA.

ARCO è un'applicazione progettata per eseguire comandi su macchine remote.

In origine tale tool era deputato alla gestione dei soli servizi correlati con LSF (Load Sharing Facility) Multicluster, e si è evoluto in seguito come applicazione general-purpose in grado di manipolare oggetti remoti di qualunque tipo.

Nel seguito verranno presentate l'architettura del sistema e le sue funzionalità (cap. 2), e saranno esposti (cap. 3) alcuni dettagli sull'implementazione.

2. ARCHITETTURA DEL SISTEMA

2.1 – Overview

ARCO è uno strumento progettato per l'amministrazione visuale di servizi remoti via web.

Il sistema è service-centered, e l'architettura d'insieme è descritta dalla seguente figura 1:

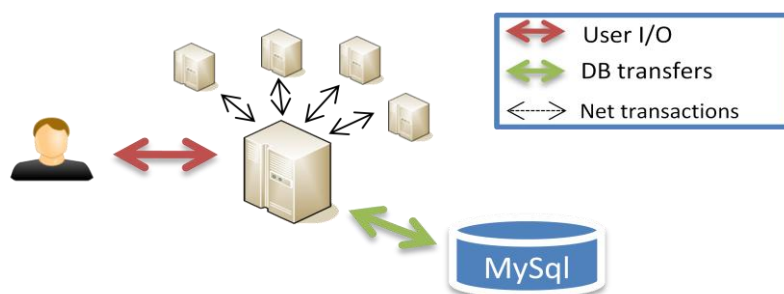


Figura 1 - Working Model di ARCO

Il sistema accetta connessioni utente, ed esegue su un set di macchine remote uno o più servizi specificati tramite ordinarie operazioni di I/O (elaborate tramite web form processing).

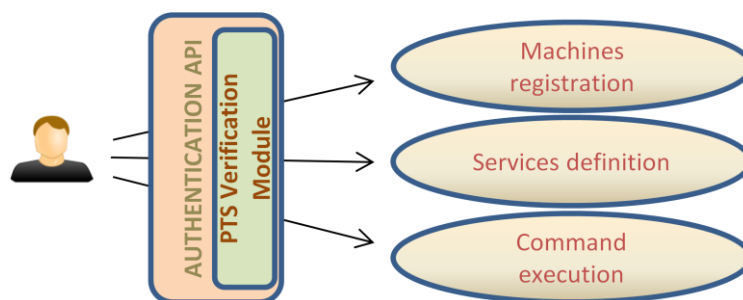


Figura 2 - Diagramma Pseudo-UML su Use-Case Study

Il diagramma pseudo-UML di figura 2 definisce i task ammessi per un utente di ARCO.

L'operatività è garantita a condizione che l'utente sia in grado di fornire credenziali adeguate: la gestione della security è granulare, e prevede che l'esecuzione di differenti servizi possa essere subordinata all'appartenenza a gruppi prestabiliti e configurabili.

Il controllo delle credenziali di accesso e della *group membership* è effettuato integrando l'AMACA Extended Authentication API (e, quindi, sulle strutture dati AFS).

2.2 – Funzionalità dell'applicazione

Se l'attore fornisce delle credenziali d'accesso valide, e può dunque effettuare con successo il login nell'applicazione, può usufruire di tre funzionalità esposte (o di un subset opportunamente calibrato dal mapping con la PTS *membership*):

- Definizione dei servizi da eseguire: si può trattare di comandi presenti in modo nativo nel sistema o di script *custom* realizzati ad-hoc;
- Gestione delle macchine su cui verranno eseguiti i servizi,
- Esecuzione dei servizi su un set di macchine specificato.

ARCO si appoggia ad un back-end MySQL per la registrazione delle entità associate ad ognuna delle *feature* appena menzionate, e per offrire anche utili funzionalità di logging (di ogni attività compiuta dai singoli utenti collegati).

In particolare, volendo caratterizzare opportunamente ciò che nei punti precedenti è stato genericamente nominato "Gestione", è opportuno sottolineare che questa "macro-funzionalità" prevede ovviamente la possibilità di definire, modificare e cancellare macchine e servizi all'interno del sistema.

Per agevolare l'amministratore nel management delle entità gestite da ARCO, sono state previste delle *facilities* che permettono di:

- Registrare set di macchine target mediante upload di file testuali che contengono gli *hostname* (è stato gestito anche il *parsing* dei file di configurazione di LSF).
- Raggruppamento delle macchine per cluster (con possibilità di manovrare operazioni su interi domini in maniera assolutamente rapida)

In particolare, è possibile assegnare ad una generica macchina lo status di *disabled*, nel caso in cui (per motivi di servizio) essa si trovi in *downtime* per un certo periodo. Tale status può essere rimosso non appena la messa in opera dell'host è ripristinata.

Nel prossimo capitolo 3 verranno esposti alcuni dettagli sull'implementazione.

3 – Dettagli sull'implementazione

3.1 – Connessioni remote

Uno dei risvolti più interessanti nella scrittura di ARCO è stato quello di integrare una libreria di funzioni in grado di gestire l'autenticazione SSH mediante scambio di chiavi.

Si è fatto uso di una *function library* per PHP (php.ssh2, <http://it2.php.net/ssh2>), in grado di gestire Secure Shell, che si occupa di gestire le connessioni trasferendo una *passphrase* la cui memorizzazione è gestita in modo crittato a due fasi (usando un secondo binding: php.mcrypt - <http://it2.php.net/mcrypt>).

L'intero sistema effettua transazioni http incapsulate con SSL, per garantire ulteriore sicurezza.

3.2 – Database di ARCO

La seguente figura 3 mostra il diagramma E-R del database associato ad ARCO:

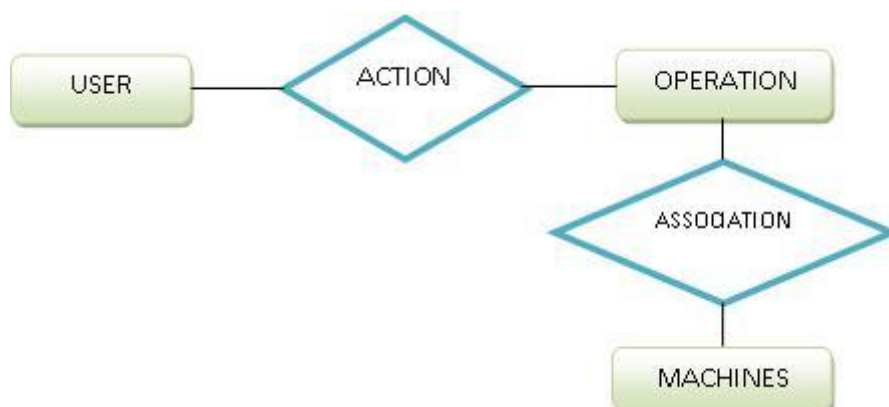


Figura 3 - Diagramma E-R

Le relazioni tra le varie entità sono tutte n-n.

Segue, dunque, il data dictionary risultante:

3.2.1 – Tabella User

Field	Type	Null	Key	Default	Extra
u_id	int(11)	NO	PRI	NULL	auto_increment
username	varchar(255)	NO			
lastlogin	datetime	YES		NULL	

Il campo <lastlogin> identifica l'ultimo accesso effettuato dall'utente <username>.

3.2.2 – Tabella Operations

Field	Type	Null	Key	Default	Extra
op_id	int(11)	NO	PRI	NULL	auto_increment
op_name	varchar(255)	NO			
op_path	varchar(255)	NO			
op_PTS	varchar(100)	YES		NULL	

Il campo <op_path> identifica il percorso assoluto del comando da eseguire (di cui <op_name> è un alias interno). Il campo <op_PTS> identifica il gruppo PTS a cui l'utente deve appartenere, per poter eseguire con successo <op_name>.

3.2.3 - Tabella Machines

Field	Type	Null	Key	Default	Extra
m_id	int(11)	NO	PRI	NULL	auto_increment
m_name	varchar(255)	NO			
disabled	enum('y','n')	NO		n	

Il campo <m_name> identifica l'hostname della generica macchina registrata. Tale macchina, come già detto, può o meno essere abilitata all'esecuzione di comandi remoti. Questa informazione è memorizzata nel campo <disabled>.

3.2.4 – Tabella Actions

Field	Type	Null	Key	Default	Extra
a_id	int(11)	NO	PRI	NULL	auto_increment
op_id	int(11)	NO	PRI		
u_id	int(11)	NO	PRI		
m_id	int(11)	NO	PRI		
start	datetime	YES		NULL	
end	datetime	YES		NULL	
status	text	YES		NULL	

I tre identificativi <{op, u, m}_id> rappresentano, rispettivamente, l'id dell'operazione eseguita, l'id dell'utente che l'ha lanciata, e l'id della macchina target (sono tutte chiavi esterne verso le rispettive tabelle).

I campi <start> e <end> descrivono il timestamp di inizio e fine dell'azione associata all'operazione eseguita, e <status> ne esprime l'esito (ovvero l'eccezione sollevata in caso d'errore, o un valore di default "Operation Succeeded").

3.2.5 – Tabella association

Field	Type	Null	Key	Default	Extra
op_id	int(11)	NO	PRI		
m_id	int(11)	NO	PRI		

Questa tabella rappresenta l'associazione tra macchine (<m_id>) e operazioni eseguibili su di esse (<op_id>). Entrambi i campi sono chiavi esterne verso, rispettivamente, le tabelle operations e machines.