

**Esame di SISTEMI DISTRIBUITI**

**APPELLO DEL 6 SETTEMBRE 2004 – TEMPO A DISPOSIZIONE: 2.5 ORE**

Si consideri un sistema di calcolo distribuito composto da un insieme di nodi distinti in tre tipologie: **Resource Node**, **Client Node** e **Broker**. I **Resource Node** gestiscono risorse di vario tipo, e le mettono a disposizione dei **Client Node** associando ad ognuna di esse un costo per minuto di utilizzo. I **Broker** svolgono la funzione di intermediari tra le altre due tipologie di nodi.

Ogni **Resource Node** mantiene l'elenco delle risorse che esso gestisce, descritte mediante triple  $\langle \text{tipoRisorsa}, \text{disponibilità}, \text{costo} \rangle$ , dove *tipoRisorsa* è una stringa, *disponibilità* rappresenta la quantità di risorsa correntemente disponibile presso il nodo (si esprima tale quantità mediante un intero nell'intervallo  $[0, 1000]$ ) e *costo* è il costo per minuto di utilizzo (tale quantità, espressa nell'intervallo  $[0, 100]$ , può variare in base alle richieste ricevute ed alla disponibilità attuale). Inoltre, ogni **Resource Node** esporta i seguenti metodi:

- *risorseDisponibili*, che restituisce l'elenco delle tipologie di risorse attualmente disponibili su quel nodo;
- *costoRisorsa*, che restituisce il costo attuale di una data quantità di una risorsa (tale quantità, nei 3 minuti successivi all'invocazione del metodo, viene inoltre riservata al nodo che ha effettuato la richiesta, e durante tale periodo il costo viene considerato invariato);
- *usaRisorsa*, che rende disponibile ad un **Client Node** una data quantità di una risorsa per un dato periodo di tempo (verificando ovviamente che la quantità di risorsa richiesta sia stata precedentemente riservata).

I **Client Node** mantengono un elenco di **Broker** e **Resource Node** conosciuti, e per l'utilizzo delle risorse gestite dai **Resource Node** possono interagire direttamente con essi o rivolgersi ai **Broker**. Inoltre, ogni **Client Node** esporta i metodi *registraResourceNode* e *registraBroker*, che ricevono e registrano l'identità di un **Resource Node** e di un **Broker**, rispettivamente.

I **Broker** mantengono un elenco di **Resource Node** conosciuti, ed offrono funzionalità di prenotazione di risorse per l'esecuzione di applicazioni parallele. In particolare, ogni **Broker** esporta i seguenti metodi:

- *registraResourceNode*, che riceve e registra l'identità di un **Resource Node**;
- *prenotaRisorse*, che riceve la descrizione di un'applicazione parallela, espressa mediante un insieme di triple  $\langle \text{tipoRisorsa}, \text{quantitàNecessaria}, \text{tempoRichiesto} \rangle$ , ed effettua la prenotazione dell'insieme di risorse necessario ad eseguire l'applicazione a costo minimo (in altri termini, per ogni tripla  $\langle r, q, t \rangle$  nell'insieme, prenota una quantità  $q$  di risorsa  $r$  per  $t$  minuti presso il **Resource Node** che offre tale quantità di risorsa al minor costo); il metodo restituisce un insieme di coppie  $\langle \text{tipoRisorsa}, \text{resourceNodeID} \rangle$  che associa ad ogni tipo di risorsa necessario all'applicazione l'identificativo del **Resource Node** presso cui tale risorsa è stata prenotata (cioè, per ogni tripla  $\langle r, q, t \rangle$  nell'insieme ricevuto, l'insieme restituito deve contenere una coppia  $\langle r, n \rangle$  dove  $n$  è l'identificativo del nodo presso cui la risorsa è stata prenotata).

Si progetti il sistema descritto e lo si implementi utilizzando i meccanismi RMI di Java. Inoltre, nel *main* del **Client Node**, si includa un semplice esempio di utilizzo delle funzionalità del sistema.