Processi, Threads e Agenti

Processi in Sistemi Distribuiti

- Un **sistema software distribuito** è composto da un insieme di processi in esecuzione su più nodi del sistema.
- Un **algoritmo distribuito** può essere definito come un insieme $\{P_1, P_2, ..., P_n\}$ dove P_i è un processo.
- I processi possono comunicare, sincronizzarsi e cooperare con le stesse modalità sia se sono in esecuzione su nodi remoti, sia se eseguono sullo stesso nodo.

Processi e Thread

- Il processo è una entità autonoma ed attiva che svolge un compito specifico.
- Il concetto di Thread ha permesso di ottimizzare l'esecuzione dei dei programmi, tramite una ottimizzazione delle fasi di allocazione ed esecuzione (condivisione dello stato e dello spazio degli indirizzi).
- Un singolo thread completo costituisce un processo.
- La condivisione di risorse rende differenti i thread e i processi.
- I thread permettono l'esplicitazione parallelismo (*multithreading*) in un processo.

Multi-Threading

- Alcune fasi indipendenti in un processo possono essere eseguite tramite un insieme di threads (server multi-threaded).
- Il multi-threading può essere usato per ridurre la sincronizzazione e l'impatto delle operazioni di I/O bloccanti.
- In un server multi-thread, si possono attivare tanti thread per quante richieste sono da servire (un thread per client).
- Si può usare un pool di thread per limite I costi della crezione di threads.

Migrazione del Codice

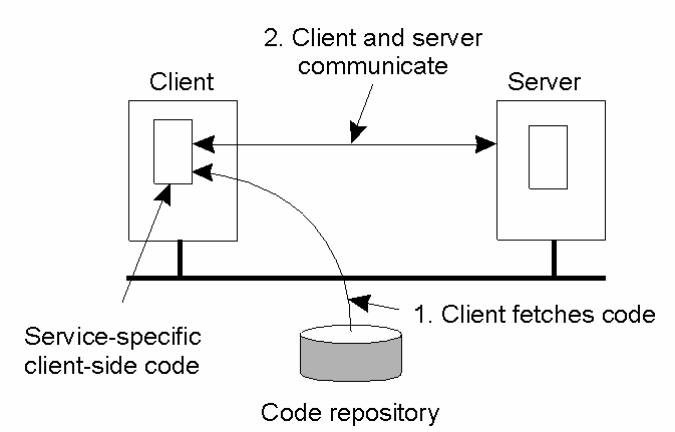
- La migrazione del Codice nei sistemi distribuiti :

  process migration

- *statica*
- *dinamica*

Per migliorare le performance o per obiettivi applicativi.

Motivi per la Migrazione del Codice



La configurazione dinamica di un cliente per comunicare con un server.

- Il cliente prima carica il software necessario (1),
- Quindi invoca il server (2).

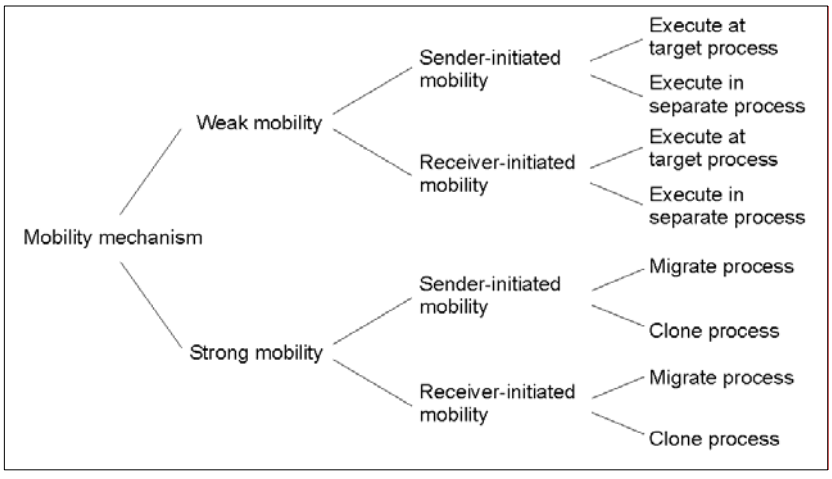
Migrazione del Codice

- **Weak mobility (mobilità debole)**: si sposta il *code segment* e i dati di inizializzazione.

  (es. Java applet)

- **Strong mobility (mobilità forte)**: si sposta il *code segment*, *execution segment* e i dati di inizializzazione.

Modelli di Migrazione del Codice



Alternative per la migrazione codice

Migrazione e Risorse Locali

Collegamento Risorsa-macchina

Collegamento processo-risorsa	Unattached	Fastened	Fixed
By identifier	*MV (or GR)*	*GR (or MV)*	*GR*
By value	*CP (or MV, GR)*	*GR (or CP)*	*GR*
By type	*RB (or GR, CP)*	*RB (or GR, CP)*	*RB (or GR)*

- Azioni da prendere rispetto ai riferimenti alle risorse locali quando si ha migrazione di codice verso un'altra macchina:

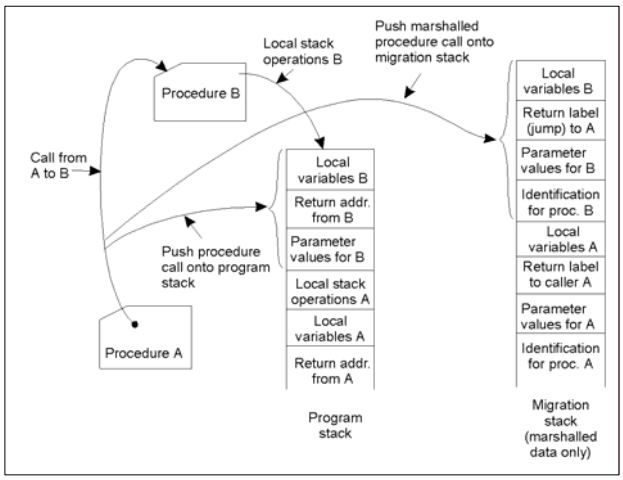
  MV (move) *GR* (global referenced)

  CP (copy) *RB* (rebind)

Migrazione in Sistemi Eterogenei

- E' più complesso.
- Richiede portabilità del codice.
- E' usato un approccio basato sul modello di macchina virtuale.
- La Weak mobility è più semplice.
- Nella strong mobility è necessario gestire l'execution segment.
- E' usato un migration stack.

Migrazione in Sistemi Eterogenei



Il funzionamento basato sul migration stack per supportare la migrazione di un execution segment in un ambiente eterogeneo.

Agenti Software in Sistemi Distribuiti

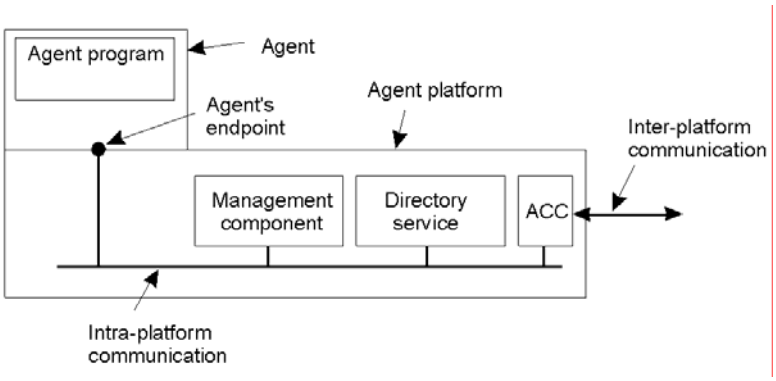
- **Software agent**: un processo *autonomo* capace di reagire e iniziare dei cambiamenti in *collaborazione* con utenti e altri agenti.
- Un agente software può prendere l'iniziativa.
- Diversi tipi di agenti:
  - Mobile agents,
  - Interface agents,
  - Information agents.

Agenti Software in Sistemi Distribuiti

Proprietà	Comune a tutti gli agenti?	Descrizione
Autonomous	Si	Può agire di propria iniziativa
Reactive	Si	Risponde alle modifiche dell'ambiente
Proactive	Si	Inizia azioni che modificano l'ambiente esterno
Communicative	Si	Può scambiare informazione con utenti e agenti
Continuous	No	Ha un tempo di vita relativamente lungo
Mobile	No	Può migrare da un processore all'altro
Adaptive	No	E' capace di apprendere

Alcune proprietà importanti per distinguere gli agenti

Tecnologia degli Agenti



Il modello generale per una piattaforma ad agenti (da [fipa98-mgt]).

Agent Communication Languages (1)

Scopo del messaggio	Descrizione	Contenuto del messaggio
INFORM	Informa che una data proposition è vera	Proposition
QUERY-IF	Chiede se una data proposition è vera	Proposition
QUERY-REF	Cerca un dato oggetto	Expression
CFP	Chiede una proposal	Proposal specifics
PROPOSE	Fornisce una proposal	Proposal
ACCEPT-PROPOSAL	Afferma che una data proposal è accettata	Proposal ID
REJECT-PROPOSAL	Afferma che una data proposal è rifiutata	Proposal ID
REQUEST	Richiede che una data azione sia eseguita	Action specification
SUBSCRIBE	Sottoscrive per una sorgente informativa	Reference to source

Esempi di tipi di messaggi nel FIPA ACL [fipa98-acl], che descrivono lo scopo e il contenuto di un messaggio.

Agent Communication Languages (2)

Campo	Valore
Purpose	INFORM
Sender	max@http://fanclub-beatrix.royalty-spotters.nl:7239
Receiver	elke@iiop://royalty-watcher.uk:5623
Language	Prolog
Ontology	genealogy
Content	female(beatrix),parent(beatrix,juliana,bernhard)

Un semplice esempio di una messaggio nel FIPA ACL scambiato tra due agenti che fanno uso di regole Prolog per esprimere informazioni genealogiche.