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 $\{P_1, P_2, \ldots, P_n\}$ 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 di elaborazione.

Processi e Thread

- Il processo è una entità autonoma che si crea/attiva e 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. Questo è vero anche su PC multi-core.

Multi-Threading

- Alcune fasi indipendenti in **un processo** possono essere eseguite tramite **un insieme di threads** (ad es.: server multi-threaded).

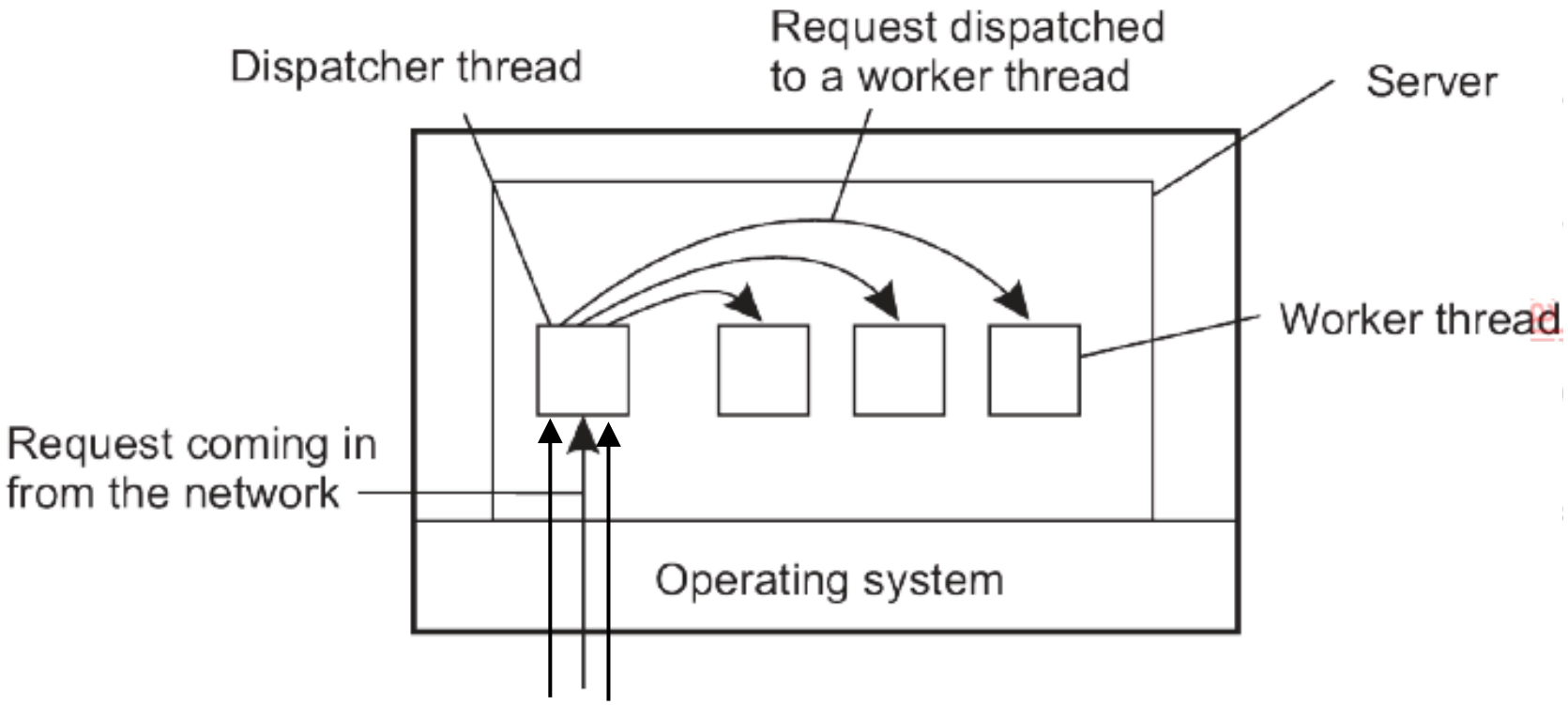
- Il multi-threading può essere usato per ridurre la sincronizzazione e l'impatto delle operazioni di I/O bloccanti (uso nei sistemi concorrenti e nei sistemi operativi). **Fondamentale nei sistemi multi-core.**

- In un server multi-thread, si possono attivare tanti thread per quante sono le richieste da servire (un thread per client).

- Si può usare un pool di thread per limitare i costi della crezione di threads. Quando necessario si attiva uno dei thread inattivi del pool.

Multi-Threading

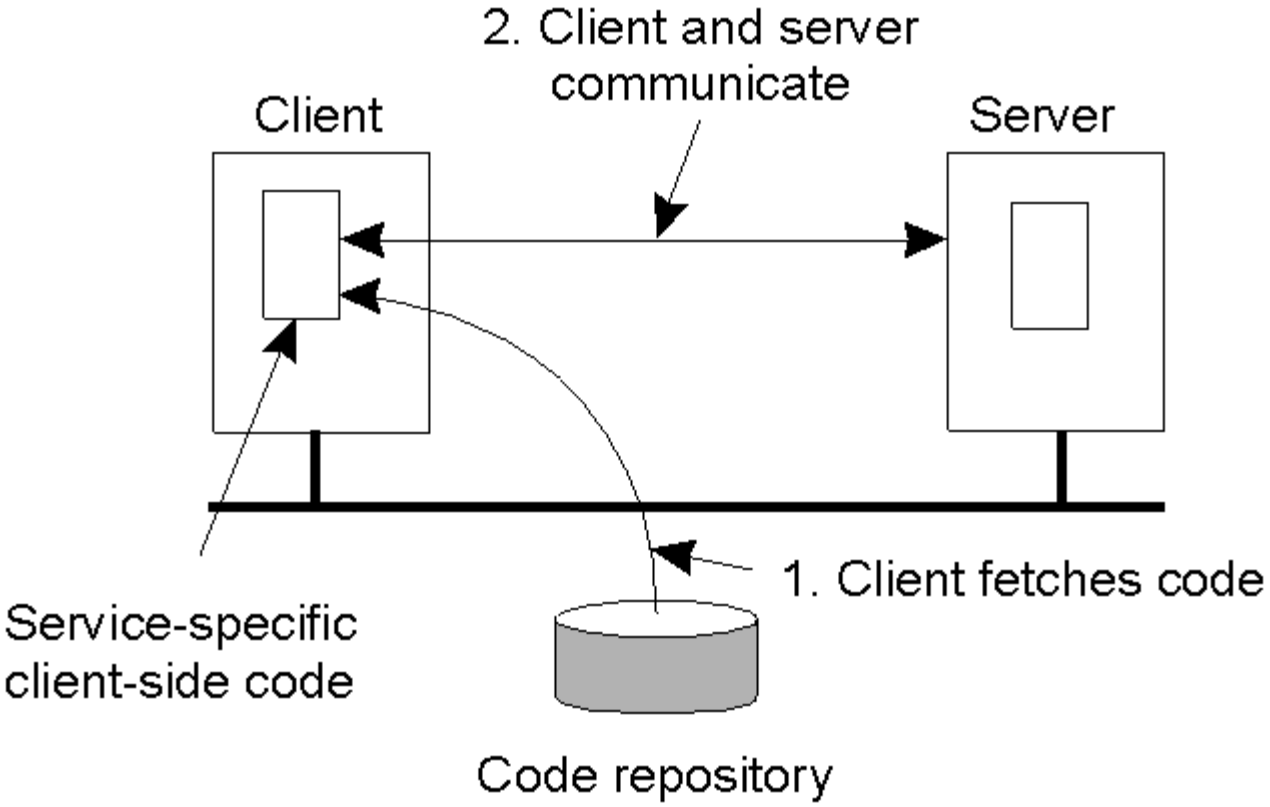
- Un server multi-threaded per gestire richieste in parallelo.



Migrazione del Codice

- In un sistema distribuito un processo o un thread puo' essere spostato da un computer all' altro.
- PERCHÈ: Per migliorare le performance o per obiettivi applicativi.
- La migrazione del Codice nei sistemi distribuiti (**process migration**) può essere:
  - *statica*
  - *dinamica*

Motivi per la Migrazione del Codice



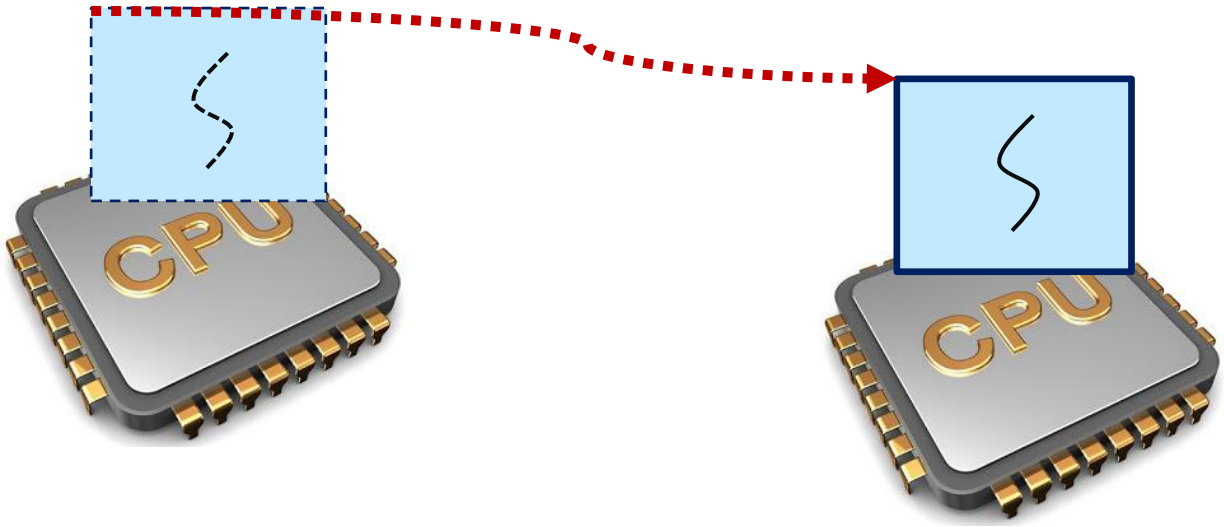
La configurazione dinamica di un cliente per comunicare con un server.

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

Migrazione del Codice

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

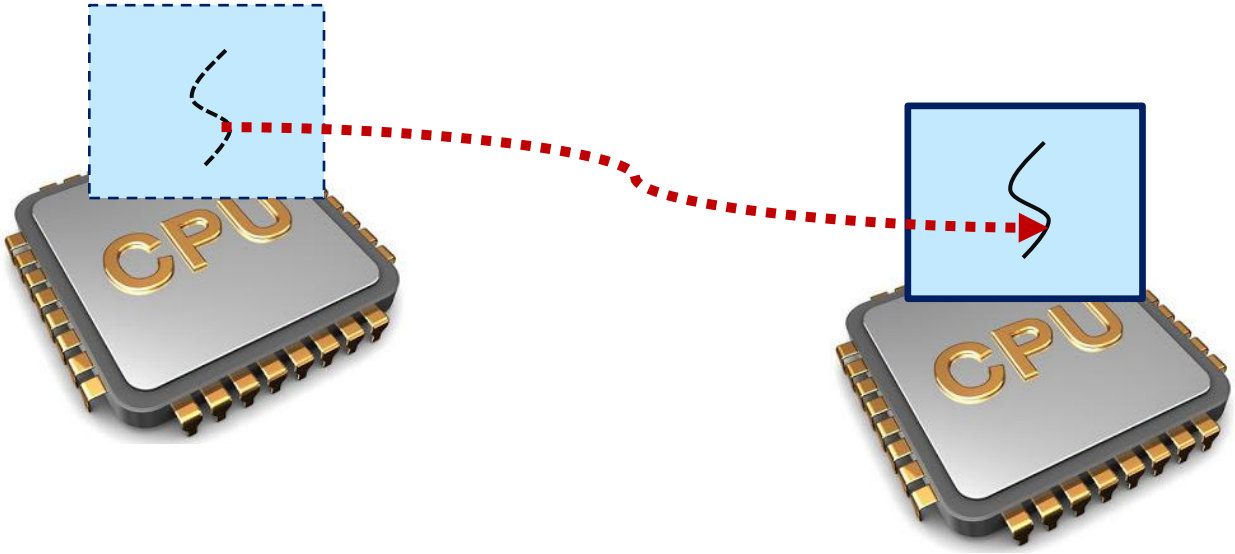
  (es. Java applet)



Weak mobility (codice e dati**)**

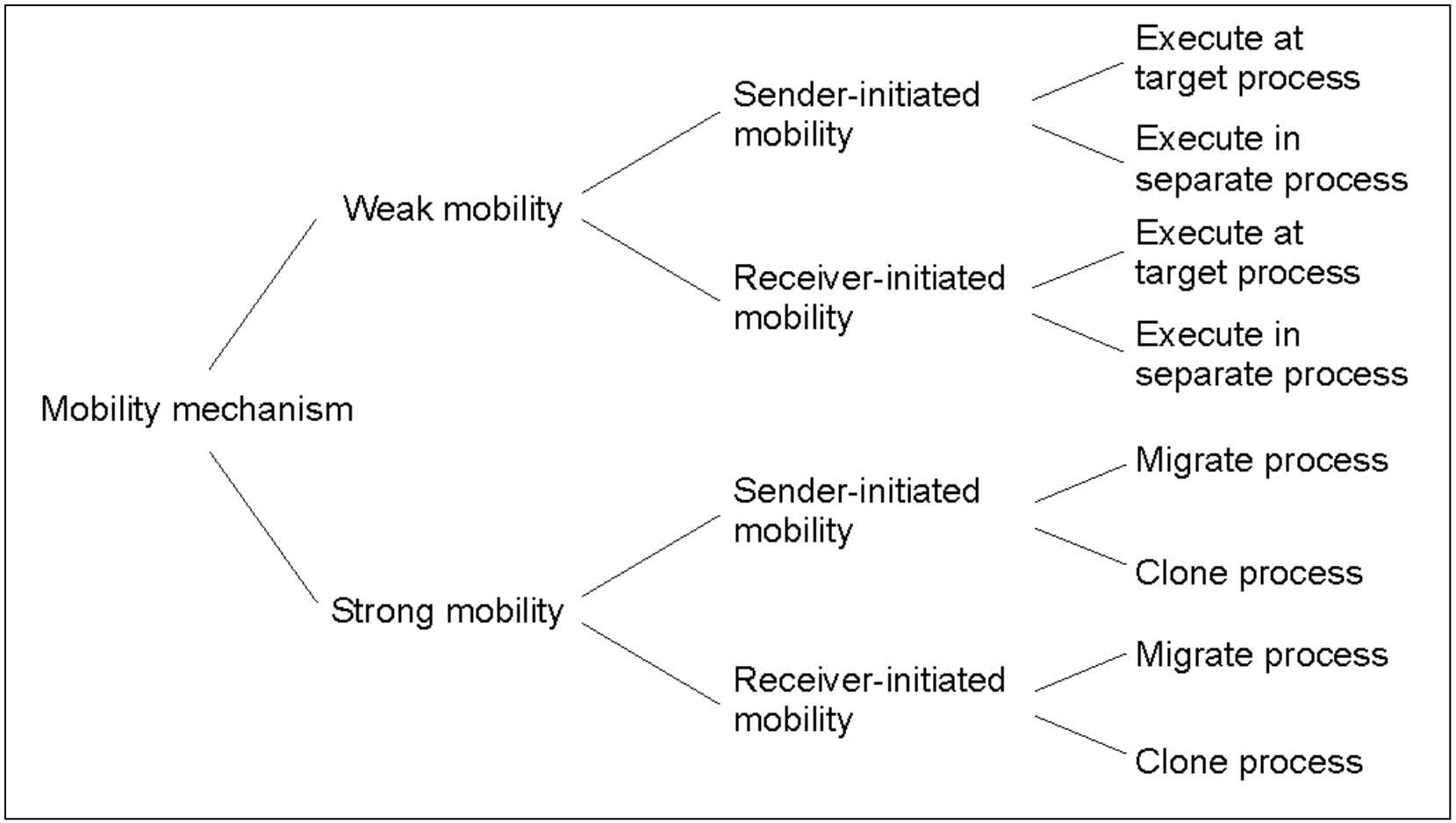
Migrazione del Codice

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



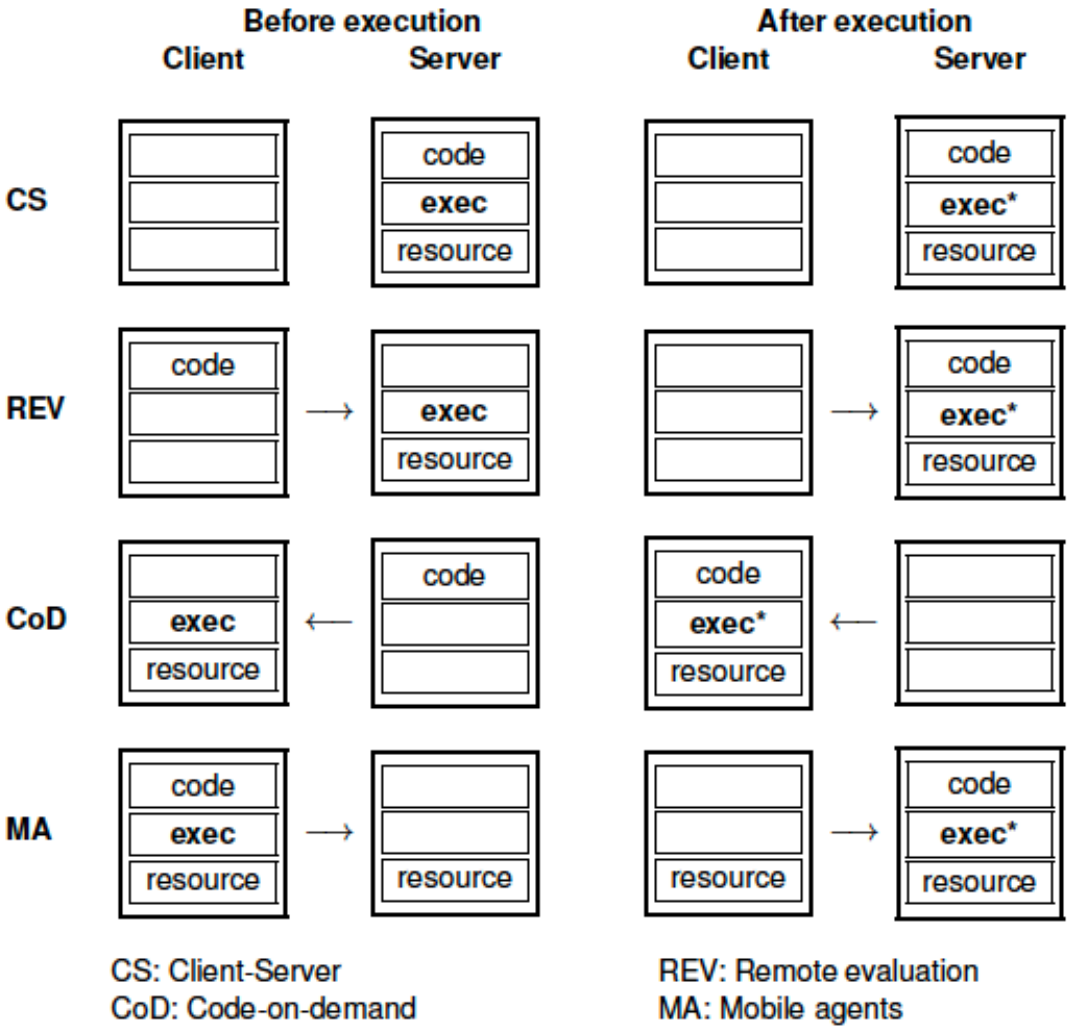
Strong mobility (codice, dati e stato esecuzione**)**

Modelli di Migrazione del Codice



Alternative per la migrazione codice

Modelli di Migrazione del Codice



Quattro paradigmi per la migrazione del 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)*

- Collegamenti: ***Unattached*** (mobile), ***Fastened*** (mobile con difficoltà e alto costo), ***Fixed*** (non mobile).

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

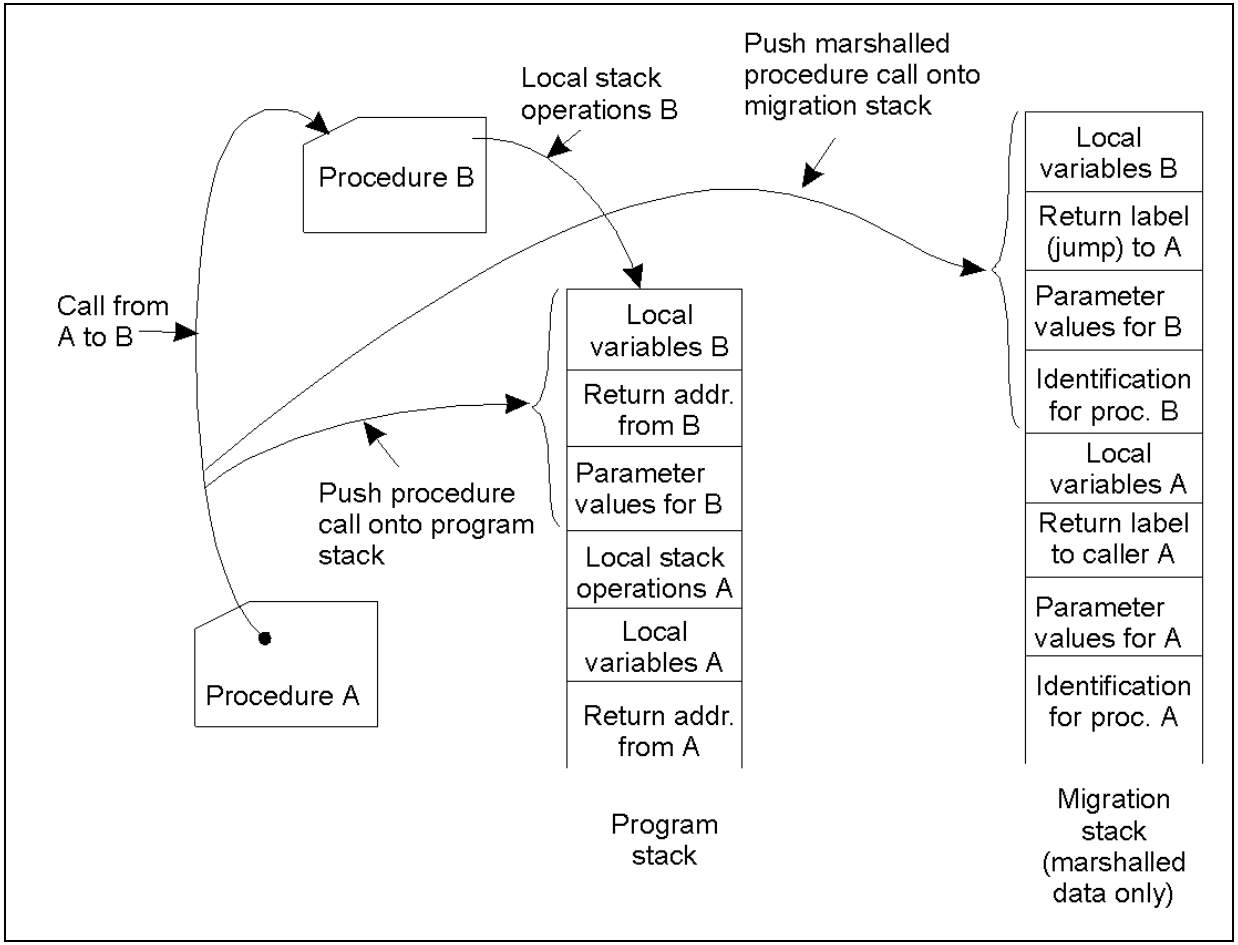
  MV (move) *GR* (global referenced)

  CP (copy) *RB* (rebind)

Migrazione in Sistemi Eterogenei

- E' più complessa.
- 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 (di svolgere operazioni, di comunicare, di interagire con altri agenti e con il mondo esterno).
- Diversi tipi di agenti:
  - Mobile agents,
  - Interface agents,
  - Information agents.

Agenti Software in Sistemi Distribuiti

Alcune proprietà importanti per distinguere gli agenti

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

DEFINIZIONE DI AGENTE MOBILE

Che cosa è un agente mobile?

- Gli agenti mobili sono programmi interoperabili, altamente compatibili che possono spostarsi su nodi di elaborazione differenti e vengono eseguiti in ambienti poco sensibili ai guasti, orientati agli oggetti e con memoria sicura.

DEFINIZIONE:

Agente mobile: entità software autocontenuta di stato e di comportamento, in grado di migrare in rete ed interagire con le risorse nei nodi visitati, scoprendo i servizi offerti.

CAMPI APPLICATIVI

- **Motori di Ricerca distribuiti**

  Utilizzare agenti mobili per raccogliere informazioni che verranno scaricate in un database centralizzato.
- **Amministrazione remota di sistemi distribuiti**

  Agenti mobili potrebbero circolare in una intranet e segnare eventuali anomalie delle risorse (stampanti senza carta, interconnessioni non funzionanti).
- **Messaggi Interattivi**

  Spedire messaggi di posta elettronica che interagiscono direttamente con l'utente, implementare wizard per guidare l'utente nello svolgimento di un'operazione.
- **Applicazioni client/server**

  Utilizzare client autonomi.

SICUREZZA

- **L'esecuzione di un agente mobile implica la gestione di diversi aspetti di sicurezza.**

- **SecurityManager**
  Controllare e limitare gli accessi al file system locale.

- **Digital signatures**
  Autenticare la provenienza di un oggetto.