DPIoT - Duck Typing in Python

Tommaso Puccetti
Studente presso Universita degli studi di Firenze

November 5, 2019

Contents

List of Tables

List of Figures

1 Communication Mechanisms

1.1 Basi

1.1.1 Middleware

Il **middleware** un insieme di applicazioni e protocolli "**general purpose**" che risiedono all'interno del livello applicativo. dunque un livello software che astrae dall'eterogeneit di rete, hardware, sistemi operativi e linguaggi di programmazione, con lo **scopo di fornire interfacce comuni che assicurino**

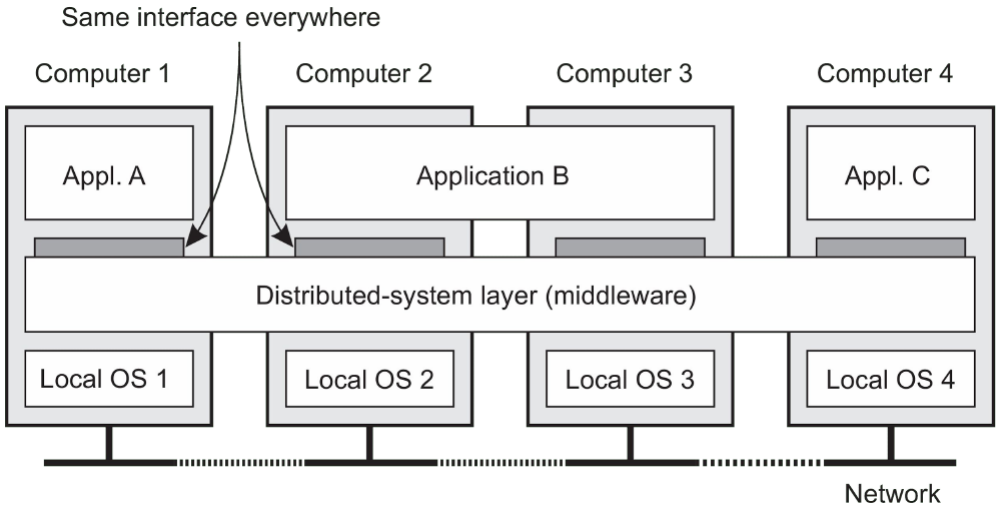

Figure 1: Middleware layer

modelli di comunicazione e di computazione uniformi. Questo livello, dunque, costituisce un insieme di protocolli condivisi dalle applicazioni pi specifiche al livello soprastante. In sintesi, un livello middleware offre servizi alle applicazioni quali:

- Comunicazione;
- Meccanismi di sicurezza;
- Transazioni
- Error-recovery;
- Gestione di risorse condivise.

Questi servizi sono indipendenti rispetto alle specifiche applicazioni. Alcuni esempi:

- Protocolli di autenticazione e autorizzazione (criptografia ssh)
- Protocolli di commit. Sono utilizzati per realizzare l'atomicit nelle transazioni. Stabiliscono se in un insieme di processi tutti hanno svolto una particolare operazione o se non stata svolta affatto.

Nello specifico vedremo come i **protocolli di comunicazione middleware supportino servizi di comunicazione ad alto livello** e permettano, per esempio, la chiamata a procedure o oggetti remoti in modo **trasparente.**

1.1.2 Coordinazione diretta

Un tipi di comunicazione nella quale le componenti partecipanti sono:

- **Referentially coupled**: durante la comunicazione gli attori utilizzano riferimenti espliciti ai loro interlocutori.
- **Temporally coupled**: entrambe le componenti devono essere in esecuzione (up and running).

Il libro propone un'introduzione ai tipi di comunicazione (persist, transient, synchronous, asynchronous).

1.2 Remote Procedure Call

Molti sistemi distribuiti sono basati sullo scambio di messaggi tra processi, tuttavia questo tipo di approccio non permette di nascondere la comunicazione tra le componenti in modo da rendere trasparente il contesto distribuito.
Una soluzione al problema stata proposta da Nelson e Birrell (1984) introducendo una modalit completamente differente nella gestione della comunicazione nel contesto di un sistema distribuito. In breve la proposta quella di chiamare procedure che sono localizzate su macchine remote:

1. quando A chiama B il processo chiamante in A sospeso;
2. l'esecuzione della procedura chiamata ha luogo in B;
3. A invia i parametri della chiamata a B che a sua volta risponder con il risultato della chiamata;
4. **Nessun passaggio di messaggi visibile dal punto di vista del programmatore.**