

# Performance Evaluation of Software Architectures using Discrete Event Simulation



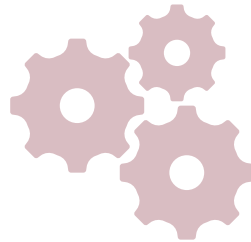
SAPIENZA  
UNIVERSITÀ DI ROMA

Tutti i diritti relativi al presente materiale didattico ed al suo contenuto sono riservati a Sapienza e ai suoi autori (o docenti che lo hanno prodotto). È consentito l'uso personale dello stesso da parte dello studente a fini di studio. Ne è vietata nel modo più assoluto la diffusione, duplicazione, cessione, trasmissione, distribuzione a terzi o al pubblico pena le sanzioni applicabili per legge

# Indice



**Introduzione  
al  
Problema**



**Design  
e  
Implementazione**



**Risultati  
e  
Validazione**

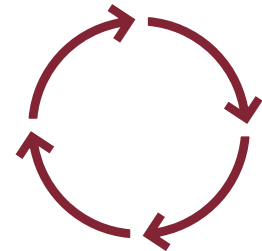
# Introduzione al Problema



**Progettazione di un  
Architettura Software**



**Valutazione delle  
Performance**



**Ripeti finche non sono  
soddisfatti i NFRs**

# Problemi



**Sovrautilizzo o Sotto-utilizzo  
delle risorse hardware**



**Insoddisficità degli  
Stakeholder**

# Simulazione dell'Architettura



**Simula il comportamento  
degli utenti**



**Schedulare i vari eventi**

# Pseudocodice delle componenti

## Client

```
state = 0;
while (True) {
    sleep(sleeps[state]);
    send(request[state]);
    waitResponse();
    state = nextState(state);
}
```

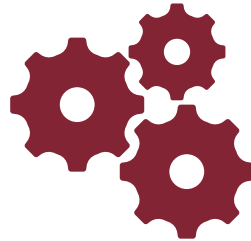
## Server

```
while (True) {
    waitForRequest();
    sleep(request);
    sendResponse();
}
```

# Indice



**Introduzione  
al  
Problema**

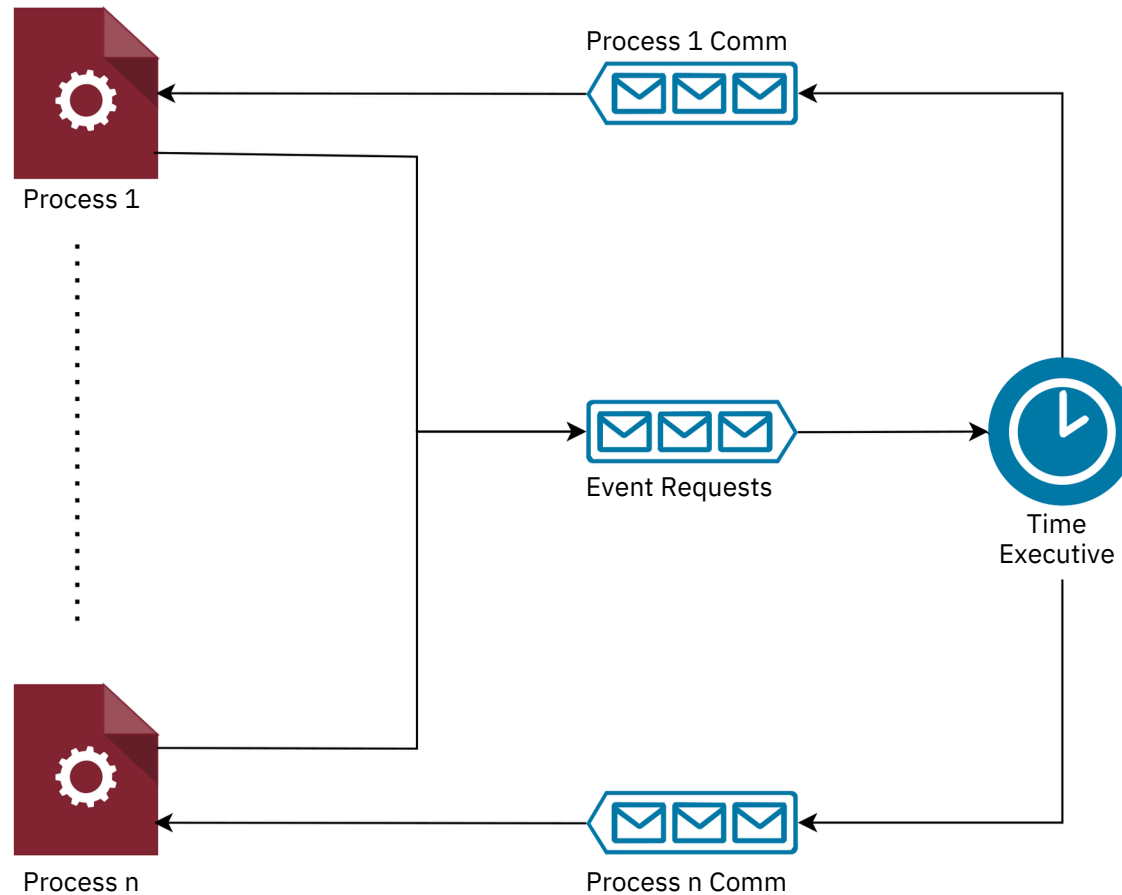


**Design  
e  
Implementazione**



**Risultati  
e  
Validazione**

# Architettura della simulazione





# Modifiche Pseudocodice Client

## Client

```
state = 0;
while (True) {
    sleep(sleeps[state]);
    send(request[state]);
    waitResponse();
    state = nextState(state);
}
```

## Primitive Aggiunte:

# Modifiche Pseudocodice Client

## Client

```
connectToSimulation();  
state = 0;  
while (True) {  
    sleep(sleeps[state]);  
    send(request[state]);  
    waitResponse();  
    state = nextState(state);  
}
```

## Primitive Aggiunte:

- **connectToSimulation();**

# Modifiche Pseudocodice Client

## Client

```
connectToSimulation();  
state = 0;  
while (True) {  
    mySleep(sleeps[state]);  
    send(request[state]);  
    waitResponse();  
    state = nextState(state);  
}
```

## Primitive Aggiunte:

- **connectToSimulation();**
- **mySleep(long double T);**

# Modifiche Pseudocodice Client

## Client

```
connectToSimulation();  
state = 0;  
while (True) {  
    mySleep(sleeps[state]);  
    send(request[state]);  
    alertBlockingCall();  
    waitResponse();  
    state = nextState(state);  
}
```

## Primitive Aggiunte:

- **connectToSimulation();**
- **mySleep(long double T);**
- **alertBlockingCall();**

# Modifiche Pseudocodice Client

## Client

```
connectToSimulation();  
state = 0;  
while (True) {  
    mySleep(sleeps[state]);  
    send(request[state]);  
    alertBlockingCall();  
    waitResponse();  
    exitBlockingCall();  
    state = nextState(state);  
}
```

## Primitive Aggiunte:

- **connectToSimulation();**
- **mySleep(long double T);**
- **alertBlockingCall();**
- **exitBlockingCall();**

# Modifiche Pseudocodice Client

## Client

```
connectToSimulation();
state = 0;
while (True) {
    mySleep(sleeps[state]);
    send(request[state]);
    alertBlockingCall();
    waitResponse();
    exitBlockingCall();
    state = nextState(state);
}
disconnectFromSimulation();
```

## Primitive Aggiunte:

- **connectToSimulation();**
- **mySleep(long double T);**
- **alertBlockingCall();**
- **exitBlockingCall();**
- **disconnectFromSimulation();**

# Modifiche Pseudocodice Server

## Server

```
while (True) {  
    waitForRequest();  
    sleep(request);  
    sendResponse();  
}
```

## Primitive Aggiunte:

- **connectToSimulation();**
- **mySleep(long double T);**
- **alertBlockingCall();**
- **exitBlockingCall();**
- **disconnectFromSimulation();**

# Modifiche Pseudocodice Server

## Server

```
connectToSimulation();  
while (True) {  
    alertBlockingCall();  
    waitForRequest();  
    exitBlockingCall();  
    sleep(request);  
    sendResponse();  
}  
disconnectFromSimulation();
```

## Primitive Aggiunte:

- **connectToSimulation();**
- **mySleep(long double T);**
- **alertBlockingCall();**
- **exitBlockingCall();**
- **disconnectFromSimulation();**



# Modifiche Pseudocodice Server

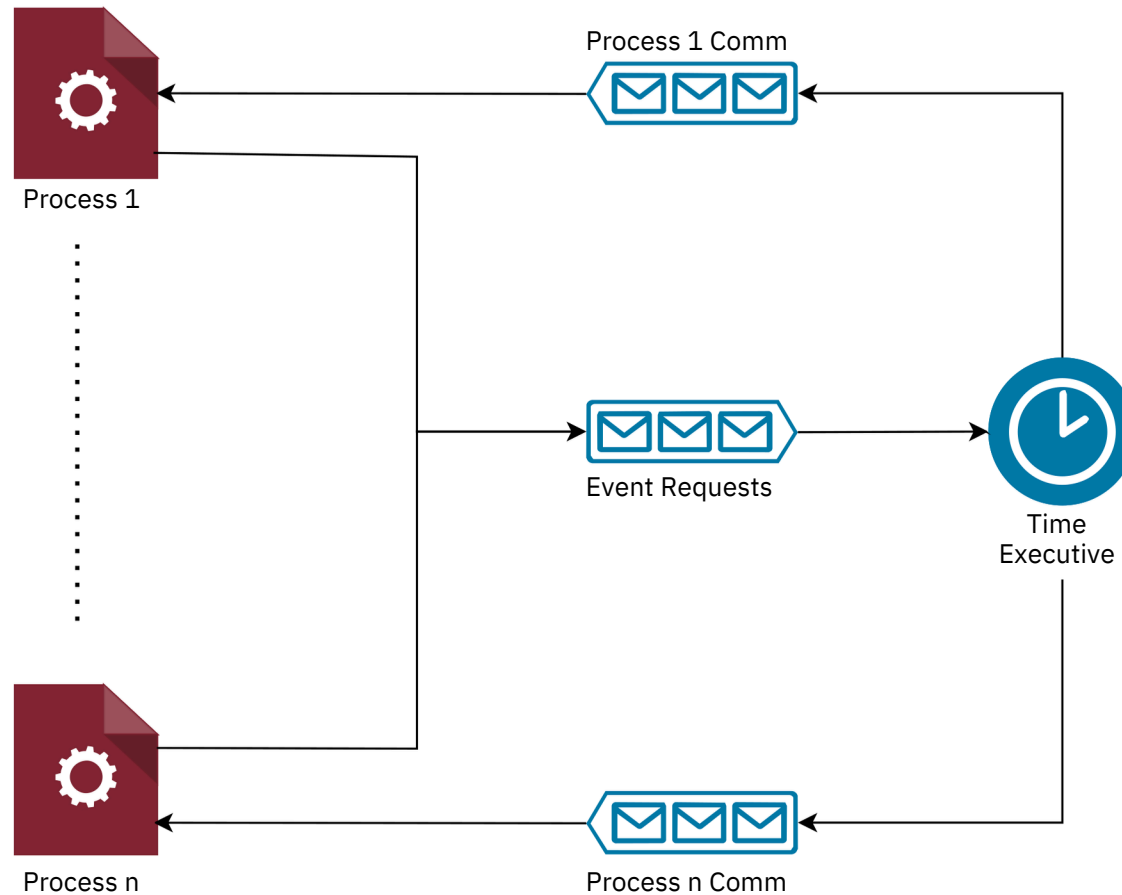
## Server

```
connectToSimulation();  
while (True) {  
    alertBlockingCall();  
    waitForRequest();  
    exitBlockingCall();  
    syncSleep(request);  
    sendResponse();  
}  
disconnectFromSimulation();
```

## Primitive Aggiunte:

- **connectToSimulation();**
- **mySleep(long double T);**
- **alertBlockingCall();**
- **exitBlockingCall();**
- **disconnectFromSimulation();**
- **syncSleep(long double T);**

# Architettura della simulazione



# Stati dei processi nella Simulazione

- **Running:** esegue delle istruzioni.

# Stati dei processi nella Simulazione

- **Running:** esegue delle istruzioni.
- **Sleeping:** sta aspettando di eseguire il prossimo evento.

# Stati dei processi nella Simulazione

- **Running:** esegue delle istruzioni.
- **Sleeping:** sta aspettando di eseguire il prossimo evento.
- **Blocking:** sta aspettando un messaggio da un altro processo.

# Stati dei processi nella Simulazione

- **Running:** esegue delle istruzioni.
- **Sleeping:** sta aspettando di eseguire il prossimo evento.
- **Blocking:** sta aspettando un messaggio da un altro processo.
- **Disconnected:** è disconnesso dalla simulazione.

# Stati dei processi nella Simulazione

- **Running:** esegue delle istruzioni.
- **Sleeping:** sta aspettando di eseguire il prossimo evento.
- **Blocking:** sta aspettando un messaggio da un altro processo.
- **Disconnected:** è disconnesso dalla simulazione.

Il prossimo evento puo' essere eseguito

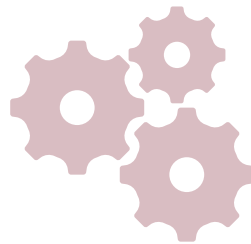


$$\#sleeping = n - \#disconnected - \#blocking$$

# Indice



**Introduzione  
al  
Problema**



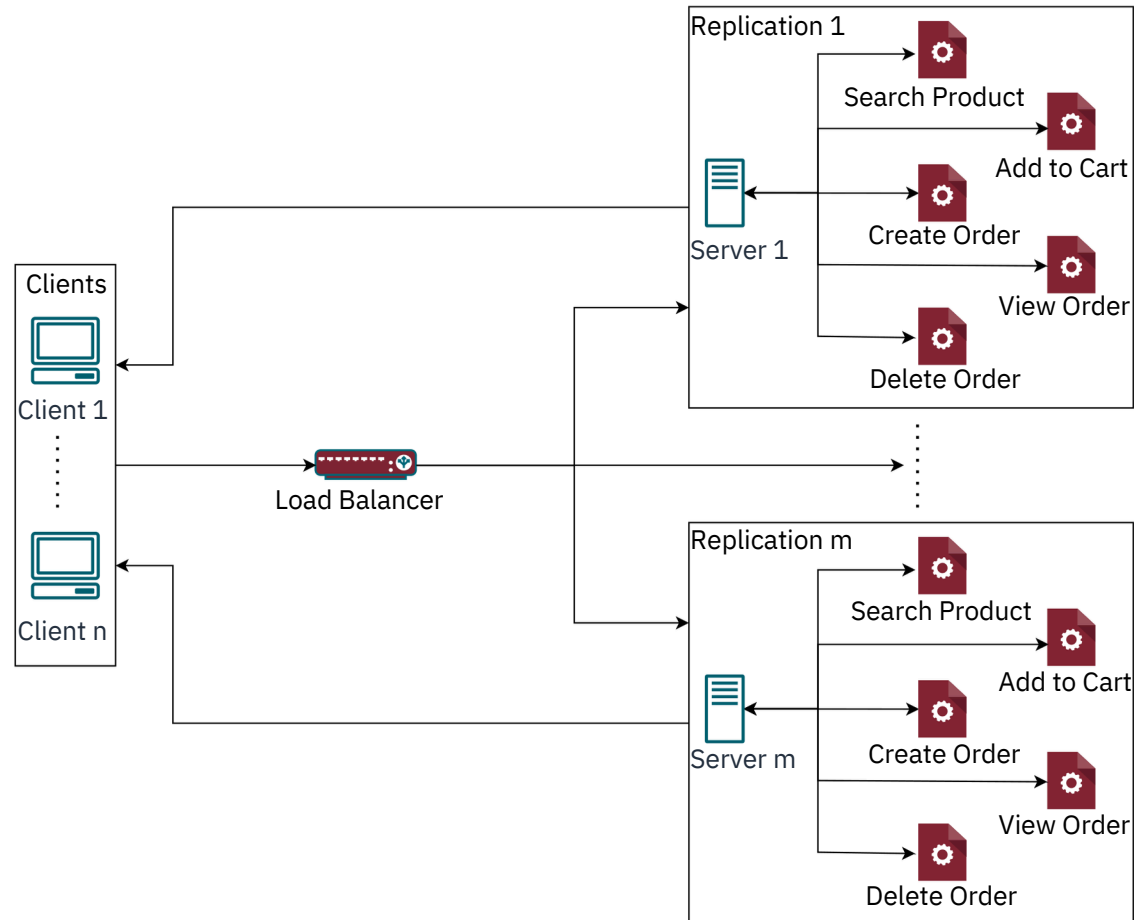
**Design  
e  
Implementazione**



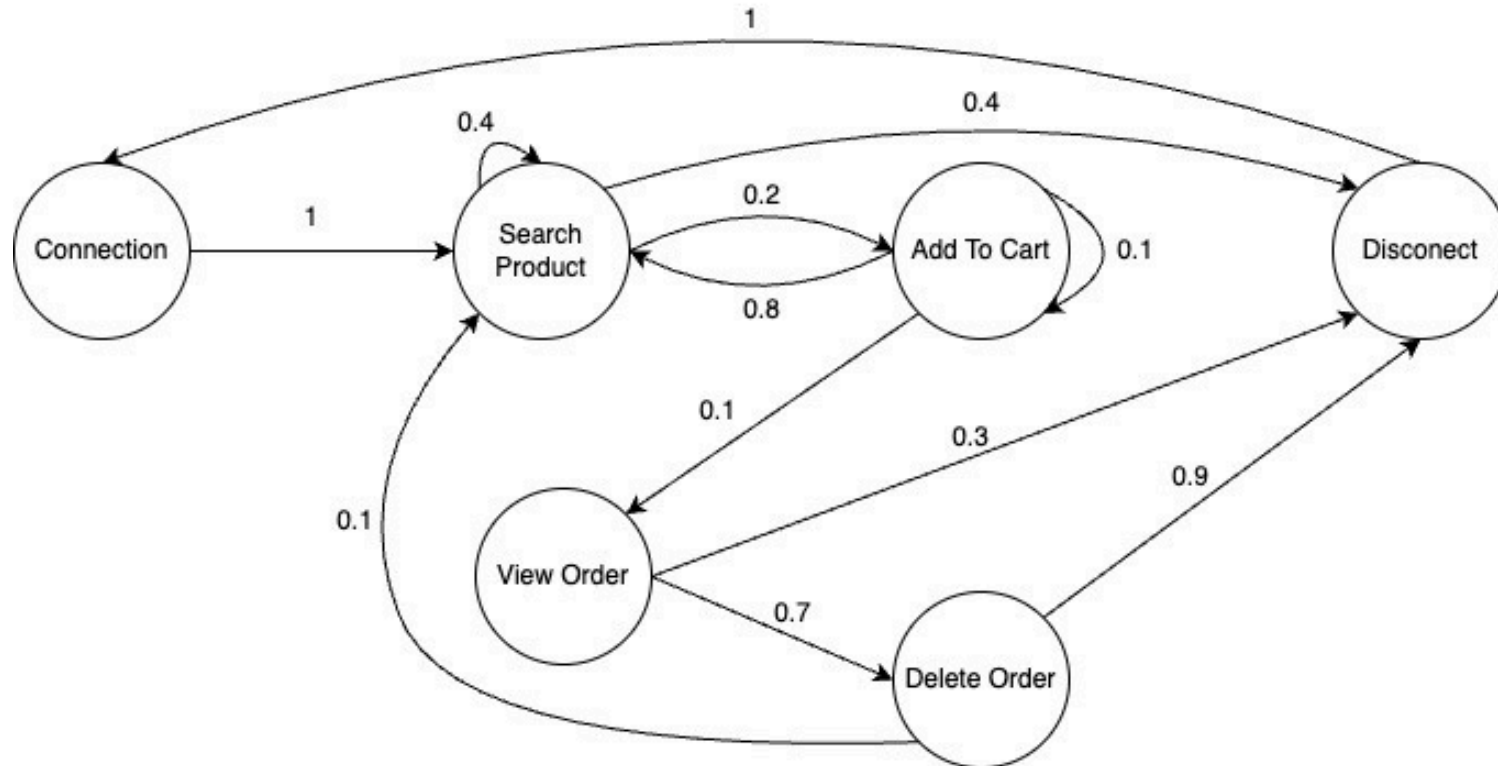
**Risultati  
e  
Validazione**



# Architettura da Simulare



# Comportamento degli Utenti



# Performance

## 1 Replica e 1000 Client

Tipo Richiesta	Min Req	Max Req	Mean Req	#Req
AddToCart	0.4	0.63	0.41	3028
CreateOrder	0.5	0.56	0.52	300
SearchProduct	0.7	1.4	0.71	14197
ViewOrder	0.4	0.44	0.45	207

# Sim. in tempo Reale vs Sim. con salti

**Obiettivo:** speed-up10x

# Sim. in tempo Reale vs Sim. con salti

Obiettivo: speed-up10x

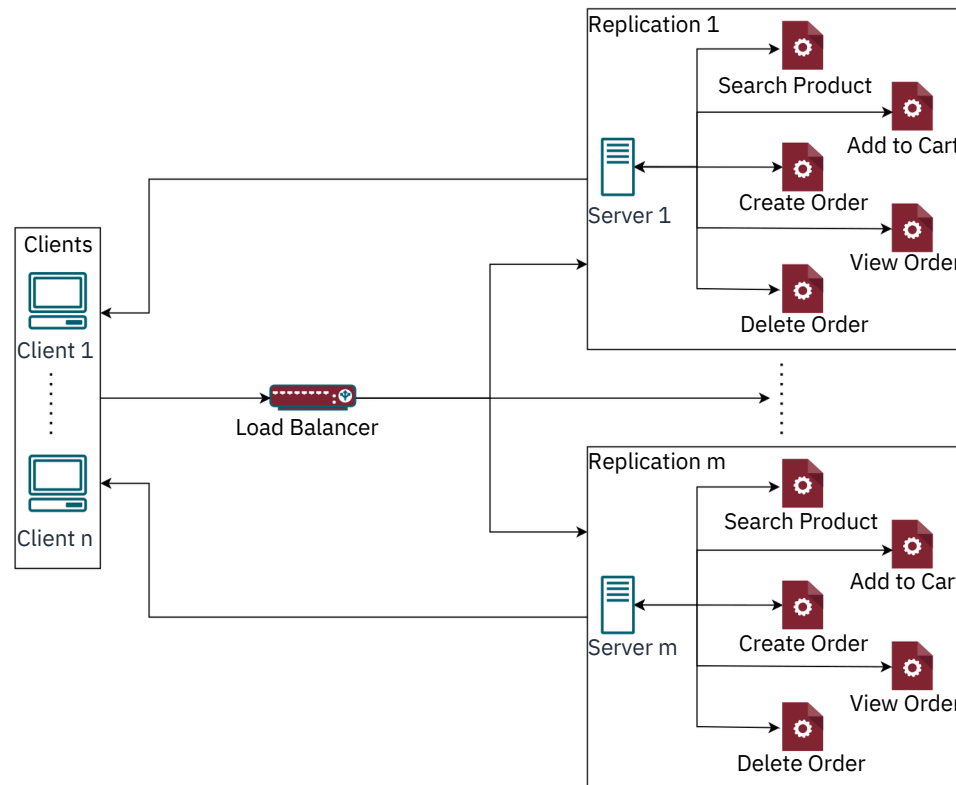
$p := \# \text{richieste dei client}$

$c := \# \text{client}$

$$\text{SpeedUp}(p, c) = \frac{T_{Real}(p, c)}{T_{Simulated}(p)}$$

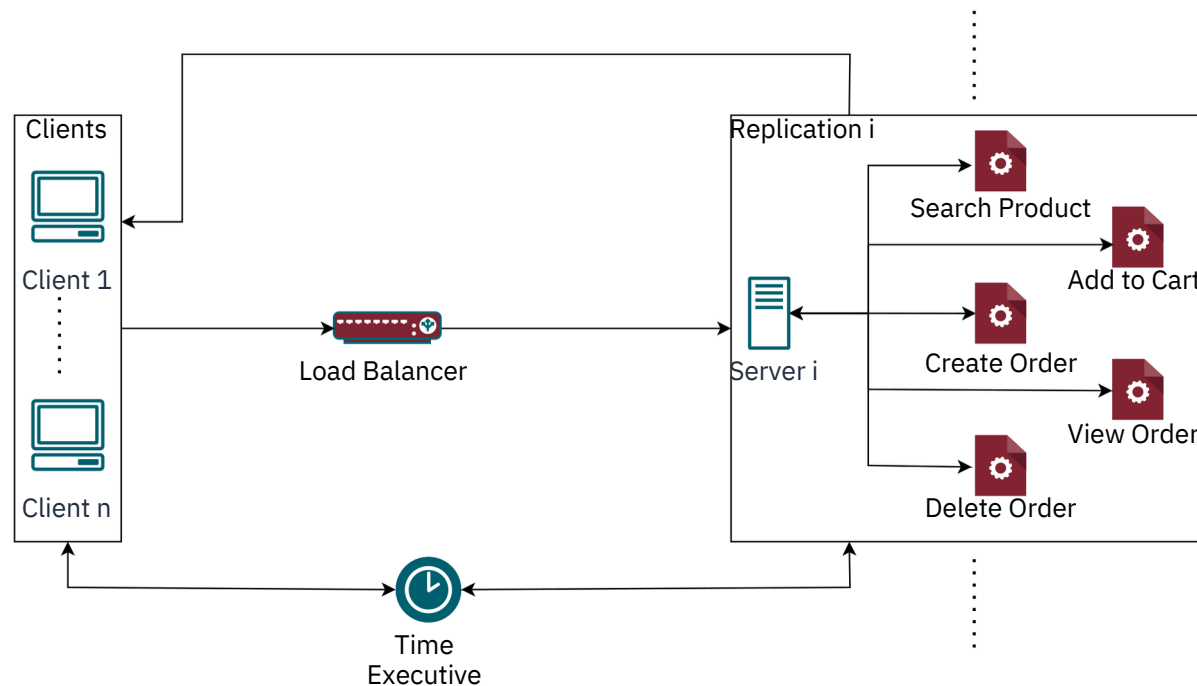
# Definizione tempo reale

$$T_{Real}(p, c) = T_{Client_{Real}}(p, c) + T_{Architecture_{real}}(p, c)$$

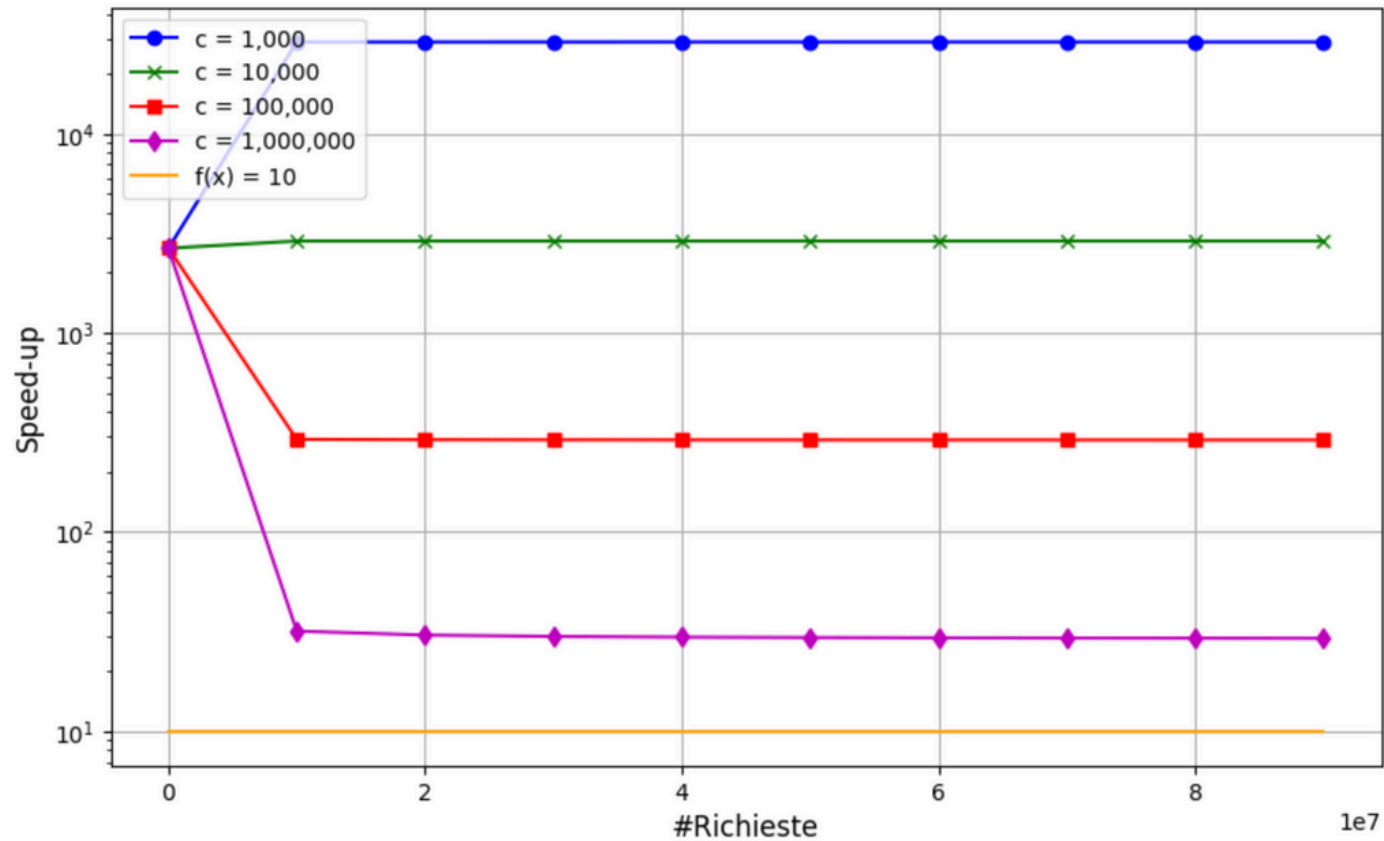


# Definizione tempo simulato

$$T_{simulated}(p) = T_{Client_{simulated}}(p) + T_{Architecture_{simulated}}(p) + T_{SimExecutive_{simulated}}(p)$$



# Risultati Speed-up

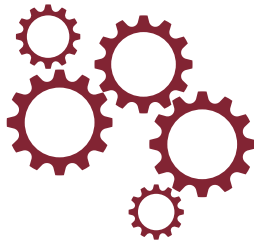




# Validazione

Richieste	Clients	T_Simulated	T_Simulazione	Errore
10000	500	60.40	58	3.97%
10000	1000	60.40	61	0.99%
20000	500	90.80	90	0.88%
20000	1000	90.80	91	0.22%
30000	500	121.20	118	2.64%
30000	1000	121.20	122	0.66%

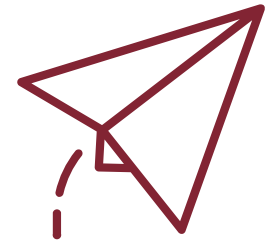
# Lavori futuri



**Simulazione di  
Software piu'  
Complessi**



**Validazione  
con piu'  
processi**



**Ottimizzazione  
delle  
Comunicazioni**



**Grazie per l'attenzione!**  
**Domande?**