

Universit'a degli Studi di Roma "Tor Vergata"
FACOLTA' DI INGEGNERIA
Corso di Laurea Magistrale in Ingegneria Informatica

Performance Modeling of Computer Systems and Networks

Project 2018-2019

Andrea Graziani - 0273395

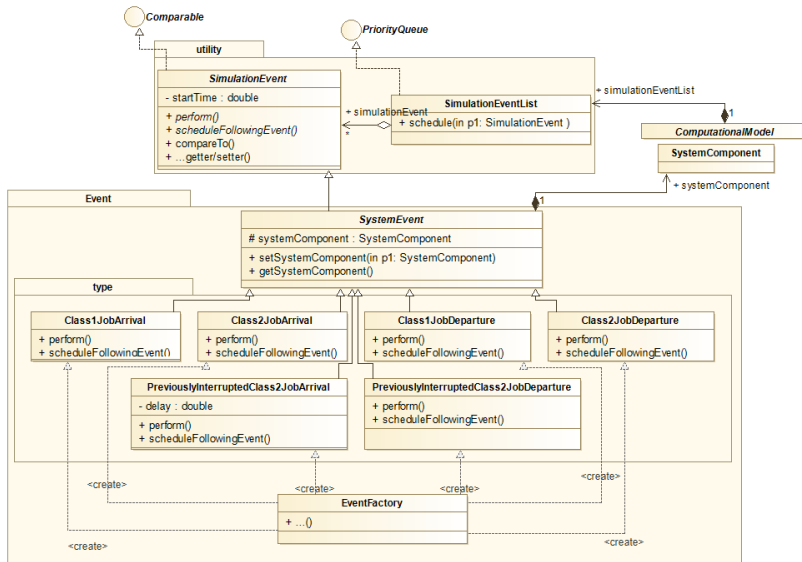
1 luglio 2019



Do **steady-state** statistics exist?

Statistics Results

Il processo client





Steady-state system statistics

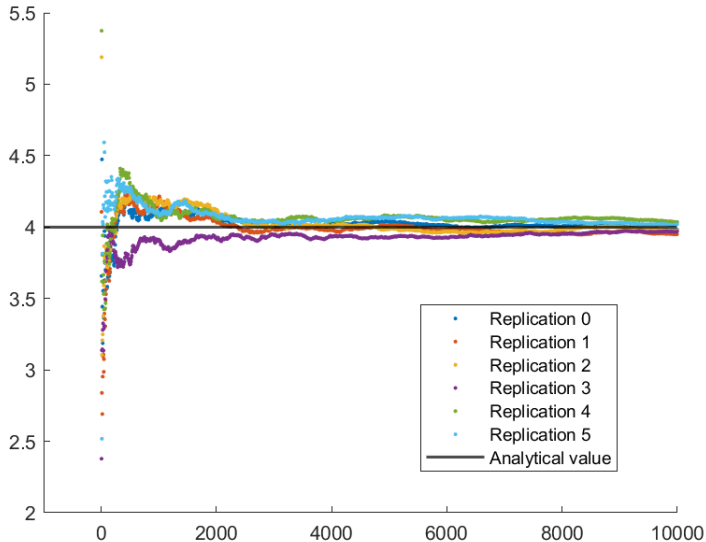
Steady-state system statistics are those statistics, if they exist, that are produced by simulating the operation of a **stationary** discrete-event system for an effectively **infinite** length of time.

In many systems, a steady state is not achieved until some time after the system is started or initiated. This initial situation is often identified as a transient state, start-up or warm-up period.

If a system is in a steady state, then the recently observed behavior of the system will continue into the future.

if the variables (called state variables) which define the behavior of the system or the process are unchanging in time.

Statistics Results





Analizzando la funzione `client_start` è facile intuire che l'invio di un messaggio di richiesta e dunque l'inizio di una trasmissione prevede:

- ▶ L'inizializzazione di una struttura dati denominata `RequestPacket`.
- ▶ L'invio della suddetta struttura al server attraverso l'ausilio di una struttura di tipo `DataNetwork`.
- ▶ Attesa della risposta dal server; in caso di mancata risposta occorre rinviare un nuovo messaggio di richiesta.



La struttura dati DataNetwork

Una struttura dati di tipo `DataNetwork` rappresenta tecnicamente un contenitore per tutte le informazioni necessarie per comunicare con un host tra cui l'**indirizzo IP** e il **numero di porta**.

Listing 1: Implementazione della struttura `DataNetwork`

```
1 typedef struct data_network {
2
3     /* Socket file descriptor */
4     int socket_fd;
5     /* Internet socket address */
6     struct sockaddr_in address;
7     /* Internet socket address length */
8     socklen_t address_len;
9
10 } DataNetwork;
```



Problema

Il server UDP deve poter scambiare più di un pacchetto con il client per poter soddisfarne le richieste. Il problema è che **l'unico numero di porta noto** al client per poter comunicare con il server è, sfortunatamente, **la stessa porta utilizzata da quest'ultimo per ricevere nuove richieste.**




Soluzione

I thread incaricati di soddisfare le richieste hanno il compito di **creare un'apposita socket di supporto** eseguendo il **binding su una porta effimera** e utilizzando quel socket per inviare e ricevere i successivi messaggi con un client.

Il client deve guardare **il numero di porta della prima risposta del server** e dunque inviare i pacchetti successivi a quella porta.

Da un punto di vista implementativo ciò è reso possibile dalla funzione `_receive_datagram` la quale provvede ad **aggiornare il contenuto della struttura** `DataNetwork` dopo la ricezione della prima risposta. Tutto ciò permette al client di **ignorare tutti quei pacchetti non provenienti dal processo server con cui interagisce**.



An abstract graphic consisting of multiple flowing, curved lines in shades of light blue and white. The lines originate from the left and curve towards the right, creating a sense of movement and fluidity. Some lines have a slight gradient, and there are small, faint white dots scattered along them.

Grazie per l'attenzione!