Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Next Event Simulation

Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering

1

Next-Event Simulation

Next-event simulation is a more general approach to discrete-event simulation

- system state
- events
- simulation clock
- event scheduling
- event list

2

Definitions and Terminology – State

The *state* of a system is a complete characterization of the system at an instance in time

3

Algorithm 1.1: how to develop a model

1. Goals and objectives
2. *Conceptual* model (cm)
   - very high level
   - which are the state variables, how they are related, which can be ignored and which not
3. Convert cm into a *specification* model (sm)
   - important: collecting and statistically analyzing data to provide the input models that drive the simulation
4. Convert sm into a *computational* model (cptm)
5. Verification
   - Is cptm consistent with sm?
6. Validation
   - Is cptm consistent with the system being analyzed?
   - Can an expert distinguish simulation output from system output?

4

Definitions and Terminology – State

- Conceptual model:
  abstract collection of variables and how they evolve over time
- Specification model:
  collection of mathematical variables together with logic and equations
- Computational model:
  collection of program variables systematically updated

- Example *ssq*: the state is number of jobs in the node
- Example *inventory system*: the state is current inventory level

5

Definitions and Terminology - Events

An *event* is an occurrence that may change the state of the system.
By definition, state cannot change except at an event time.
Each event has an associated *event type*.

- We can define artificial events (do not change system state)
  - Statistically sample the state of the system
  - Schedule an event at a prescribed time

  (block arrival flow into the node, an inventory review without orders etc.)

6

Definitions and Terminology - Simulation Clock

The *simulation clock* represents the current value of simulated time

- Discrete-event simulations lack definitive simulated time
  As a result, it is difficult to generalize or embellish models

7

Definitions and Terminology - Event Scheduling & Event List

scheduler

- a *time-advance mechanism*
  to guarantee that events occur in the correct order
- *next-event* time advance is typically used in discrete-event simulation

event list

- the data structure containing the time of next occurrence for each event type

To build a *next-event* simulation:

- construct a set of state variables
- identify the event types
- construct a set of algorithms that define state changes for each event type

8

Next-Event Simulation

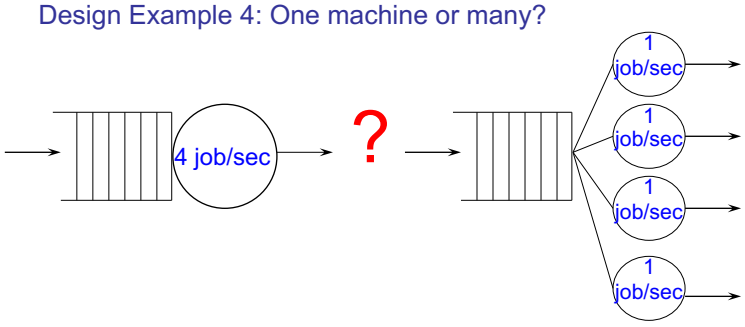
Algorithm 1

1. **Initialize** - set simulation clock and first time of occurrence for each event type
2. **Process current event** - scan event list to determine most imminent event; advance simulation clock; update state
3. **Schedule new events -** new events (if any) are placed in the event list
4. **Terminate -** Continue advancing the clock and handling events until termination condition is satisfied

Note that the simulation clock runs asynchronously; inactive periods are ignored

Model extensions

Design Example 4: One machine or many?



A Multi-Server Service Node

Conceptual model: MSQ

- Conceptual model: abstract collection of variables and how they evolve over time

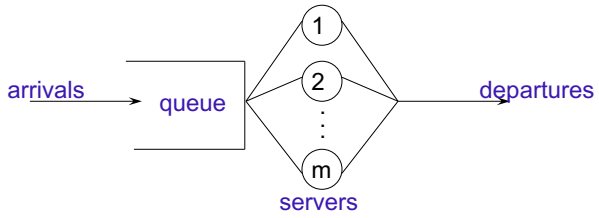
Definition 1
A *multi-server* service node consists of

- A single queue (if any)
- Two or more servers operating *in parallel*

At any instant in time,

- Each server is either *busy* or *idle*
- The queue is either *empty* or *not empty*
- If one or more servers is idle, the queue must be empty
- If the queue is not empty, all servers must be busy

Conceptual model: MSQ



When a job arrives:

- If all servers are busy, the job enters the queue
  Else an idle server is selected and the job enters service

When a job departs:

- If the queue is empty, the server becomes idle
  Else a job is removed from the queue, served by server

Servers process jobs independently

Conceptual model: Server Selection Rule

Definition 2

The algorithm used to select an idle server is called the *server selection rule*

Common selection rules:

- *random*: at random from the idle servers
- *in order*: lowest-numbered idle server
- *cyclic*: first available, starting after last selected (circular search may be required)
- *equity*: use longest-idle or lowest-utilized
- *priority*: choose the "best" idle server (modeler specifies how to dermine "best")