

Discrete Event Simulation

These slides are created by Dr. Yih Huang of George Mason University. Students registered in Dr. Huang's courses at GMU can make a single machine-readable copy and print a single copy of each slide for their own reference, so long as each slide contains the copyright statement, and GMU facilities are not used to produce paper copies. Permission for any other use, either in machine-readable or printed form, must be obtained from the author in writing.

Motive

- ❑ Many real-world systems are too difficult to analyze.
- ❑ For such systems, one can create a “*virtual reality*” where the system can be tested and its performance measured.
- ❑ How “*real*” should it be ?
 - A good simulator reproduces the essence of the timing behaviors of the system.
 - More than that wastes processor cycles
 - Less than that produces inaccurate results.

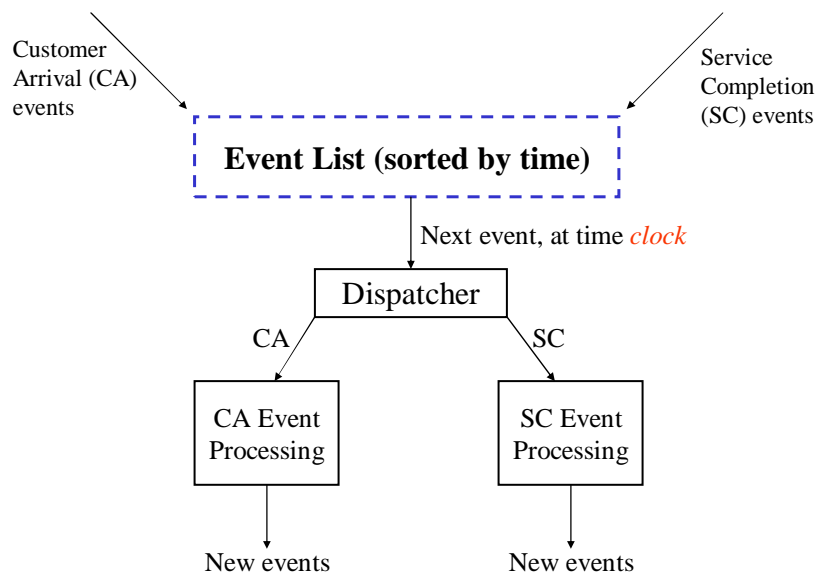
The Time

- ❑ Obviously, we need a clock in the simulation.
- ❑ However, this is not the system clock of the machine running the simulation.
 - Image simulating on a single workstation a 1,000-router network connecting 5,000 hosts.
 - It may take hours to simulate just 1 minute of activities of the entire network.
- ❑ We must create a “virtual clock” and arrange events in the simulated network accordingly

CS 656

3

Simulating M/M/1



CS 656

4

Processing CA Events

- ❑ Generate an exponential random number E .
- ❑ Schedule the next CA event at time $clock+E$.
- ❑ Increase $occupied_q_slot$ by 1.
- ❑ If $server-busy == FALSE$, then
 - Generate an exponential random variable S .
 - Schedule a Service Completion (SC) event at time $clock+S$.
 - Decrease $occupied_q_slot$ by 1.

CS 656

5

Processing SC Events

- ❑ If $occupied_q_slot > 0$, then
 - Generate an exponential random variable S .
 - Schedule a SC event at time $clock+S$.
 - Decrease $occupied_q_slot$ by 1.
- ❑ Otherwise, set $server-busy$ to FALSE.

CS 656

6

Example

- ❑ Consider 4 customers whose arrival times are 1, 5, 6, and 8, and whose service times are 5, 3, 10, 1.
- ❑ Show the contents of the event list during the course of the simulation:
 - Clock=0: (CA,1)
 - Clock= :
 - Clock= :
 - Clock= :
 - Clock= :
 - Clock= :
 - Clock= :
- ❑ As seen, the clock makes quantum jumps; hence the name discrete time simulation.

CS 656

7

More Sophisticated Events

- ❑ At a minimum, an event contains a type and occurrence time.
- ❑ It may contain additional information.
 - if customers arrive in batch (say, 3 customers arrive at the same time), then an CA event needs to contain the number of arriving customers.
- ❑ Some simulation packages give different names and use different APIs for more sophisticated events.
 - In CSIM, for example, a “mail” is an event with user-defined extra information.

CS 656

8

Maintaining The Event List

- ❑ Why the big deal ?
- ❑ Sorted array/linked-list ?
 - Using it, you will be fired on spot.
- ❑ Heap (Priority Queue)
 - $O(\log N)$ time event addition/removal, where N is the number of events in the list.
- ❑ Many other tree/heap variations.
- ❑ Calendar Queue
 - $O(1)$ time event addition/removal
 - This is according to empirical studies, confirmed by experiences

CS 656

9

The Calendar Queue --- Basic Idea

- ❑ One schedules an event on a desk calendar by writing it on the appropriate page, one page for each day.
- ❑ Events can be scheduled more than a year in advance if one writes the date besides each event.
- ❑ To find the next event, start from the page of today and search for the event with the smallest date.
- ❑ The calendar is made circular and used indefinitely.

CS 656

10

Implementation

- ❑ Each page is a sorted linked list; list nodes are events with their occurrence times.
- ❑ The calendar is simply an array of pointers (to list nodes).
- ❑ The definition of a “day” is flexible.
 - You can make a day N time units.
- ❑ The definition of a “year” is flexible.
 - You can make a year M days.

CS 656

11

Example

- ❑ 1 day = 5 time units, and 1 year = 8 days.

Page 0	(SC, 122) (CA, 123)
Page 1	
Page 2	(CA, 130)
Page 3	(CA, 135.6)
Page 4	
→ Page 5	(CA, 106) (CA, 107.2) (SC, 108) (SC, 425)
Page 6	(CA, 111) (CA, 113.5)
Page 7	(SC, 157.5)

CS 656

12

Adjustment of Calendar Size

- ❑ What happens when the number of events in the calendar is much smaller than the number of pages ?

- ❑ What happens when the number of events in the calendar is much larger than the number of pages ?

- ❑ Copy events onto another calendar whenever the # of events exceeds twice the # of days.
 - The new calendar will have twice the days too.
 - This occurs only with the **enqueue** operation (adding new event to the calendar).
- ❑ Copy events onto another calendar whenever the # of events is less than half the # of days.
 - The new calendar will have only half the days too.
 - This occurs only with the **dequeue** operation (finding and removing the next event).

Cost Analysis

- ❑ Isn't copying the entire calendar costly ?
- ❑ Consider a calendar with N days and N events in it.
- ❑ The next N enqueues will not cause copying.
- ❑ The $(N+1)$ -th enqueue will cause copying, with complexity $O(N)$.
- ❑ Thus, the **amortized cost** of copying per enqueue operation is only $O(1)$.
- ❑ By the same token, the amortized cost of copying per dequeue operation is also $O(1)$.

CS 656

15

Adjusting Day Length

- ❑ If the day length is much greater than the average separation of adjacent events, the events will be clustered in a small number of days near today and the rest will be empty.
 - Consequence ?
- ❑ If the day length is too small, most events will not be in the current year.
 - Consequence ?
- ❑ The length of a day is adjusted each time the event queue is copied to a new calendar.

CS 656

16

□ How ?

- Just find the next several events and calculate the average separation.
- This is where we cannot guarantee performance with proof.
- However, empirical studies show that the heuristic works extremely well.

A Special Case

- It may happen that the next event is N years away, forcing the dequeue operation to scan the entire calendar N times.
 - What happens when $N=10,000$?
- **Solution:** If a dequeue operation cycles through all days of a year and fails to locate the next event, it must scan the entire calendar again, looking for the next event without regarding to the year.

Examples of Simulation Packages

- **Csim**: a set of extensions to the C language to enable (virtual) parallel processes and events handling.
 - It is not only a library. The C runtime environment has to be modified too.
- **Ns2**: a combination of C++ and OTCL (object TCL) extensions.
 - Use C++ in performance critical parts/modules
 - Use a high-level script language, OCTL, to facilitate putting modules together
 - Flexible and extensible.
- **OPNET**: a commercial simulation environment