Lecture 25
Chapter # 7
SIMULATION MODELING

Simulation

The technique of using a computer to imitate the operation of an entire process or system

• ***Purpose:*** It provides a way to predict the performance of a computer system or compare several alternatives.
• ***When to use simulation:*** for the stochastic system involved is too complex to be analyzed satisfactorily by the kinds of mathematical models (e.g., queuing models).

We should keep it in mind that it actually require a precise formulation of the system to be studied & we need to develop the models of the systems & put them into action to conduct various experiments & it actually needs an appropriate abstraction of the workload as well & once the workload is defined & the model is developed & we have conducted the experiments as well then we should be able to interpret the simulation results as well.

Advantages

• Simulation is much *less expensive* than actually building a machine.
For example; in order to study the effectiveness of pipelining in CPU, simulation is a very good tool.
• It is much *more flexible* than measuring the performance of a real machine.
In a simulated system, we can quickly change important parameters such as the size or associativity of the cash in a cash simulator. & for instance, if we need to determine the overall system's performance then we may change the no. of memory module of that particular system. This is not exactly possible in case of the real system.
• Simulation allows a system to be studied in *more detail* than analytical modeling.
Because analytical modeling requires several simplifications & assumptions.
• Simulation enables to *study dynamic systems* in real, compressed or expanded time.

Terminology

We will consider the problem of simulating various scheduling techniques for given CPU, demand characteristic of job. For now, the other components of the system such as Disk & terminals will be ignored for the moment.
• ***State Variables:*** The variables whose values define the state of the system. In the CPU scheduling simulation, the state variable is the *length of the job queue.*

• ***Event***: A change in the system state. In the CPU scheduling simulation, there are two events: *arrival* of *a job* and *departure of a job*.
When a job arrives, the state of the job queue changes & similarly when a job departs, the state of the job queue or the length of the job would be changed.

• ***Continuous-Time and Discrete-Time Models***

- CT models: A model in which the system state is defined at all times.
  e.g the CPU scheduling model. i.e. the state of the system or the length of the job queue would be defined for all the time i.e. we will be measuring it for the entire period of time.
- DT models: The system state is defined only at particular instants in time.
  e.g. a process that activates every 5-msec and records the number of blocked processes in the system. It could be a demon or a background process i.e. it is only activated after certain time interval or it has got some specific or dedicated task. That particular process can be modeled as a discrete time model.

• ***Continuous-State and Discrete-State Models:***
The continuity of time does not imply the continuity of state & vice versa.

- CS models: State variables are continuous e.g. pressure of gas in thermodynamic systems. Also called a continuous-event model.
- DS models: state variables are discrete. it means that at every time instant the state variable has got a certain value a specific or discrete value for e.g. length of job queue. Also called a discrete-event model.

We can conclude that:
Thus, there are four possible combinations:

- discrete state/discrete time,
- discrete state/continuous time,
- continuous state/discrete time, and
- continuous state/continuous time models.

• ***Open and Closed Models***
They are exactly as same as we have discussed in an open & closed queueing network.

- Open model: The input is external to the model and is independent of it. The input to the system is purely independent of the system. For e.g.
  new jobs enter the model from outside.
- Closed model: There is no external input. e.g. the same jobs keep circulating in the model. A job departing the second queue reenters the first queue. This is therefore a closed model.
  When the communication from the outside model is not allowed.

Simulation Efficiency Considerations

1) An important limitation of simulation: the simplifying assumptions restrict its ability to exactly duplicate the behavior of a real system. There is in general a tradeoff between the accuracy of the simulation & the time required to write & execute the simulator. So a extremely detailed simulator can take very long time to write & execute but a less detailed simulator may be not be as accurate but it might allow sufficiently detailed results to be produced more quickly.

2) Determining the level of detail necessary when writing a new simulator depends on the level of detail necessary to solve the problem, and the consequences of being wrong.

• E.g., the consequences of being wrong when using a simulator to determine the best cache size for a new system are relatively small. That's totally ok either the cache specified will be too large resulting in slightly higher cost or the cache will be too small which will produce a degradation in the system's performance but
• The incorrect operation of a microprocessor used to control a heart pacemaker, though, can have much more serious consequences. Thus, in this case, a very detailed simulation may be warranted.

Types of Simulations

• Emulation
• Static (or Monte-Carlo) simulation
• Discrete-event simulation
• Continuous-event simulation

1) Emulation

Already discussed it very briefly in our very first lecture.
• An emulator is hardware or software that enables one computer system (called the host) to behave like another computer system (called the guest). An emulator typically enables the host system to run software or use peripheral devices designed for the guest system.
• Emulation allows program to run on the platform other than the one for which they were originally developed.
• For Example
  • Android Emulator
  • DOSBox emulates the command-line interface of DOS
  • A terminal emulator program which can be executed on a computer to make that computer appear to be a specific type of terminal to a remote system.

- Also, JVM Java Virtual Machine is an example of processor's emulator & it executes on a wide variety of different computer systems it simulates the execution of a virtual processor whose instruction set is defined in the JVM specification & Java Virtual Machine allows program written in Java to be executed on any processor for which a JVM has been written.

2) Static (Monte Carlo) Simulation

• No time parameter.

• The simulation is run until further refinement of the state of the simulated system is no longer useful or possible. Simulation is executed under some specific considerations & it can be executed until some refinement of the state of the system is achieved & its further refinement is no longer useful or possible.

• Static simulations are often used to evaluate some physical phenomenon (probabilistically), or to numerically estimate the solution of some mathematical expression, such as a complex integral.

Example
• Consider the problem of numerically determining the value of π.
• We begin with a geometric description as shown in Figure 1.
• Since the area of a circle with a radius of 1 is $\pi(1)^2 = \pi$, the area of the quarter-circle within the first quadrant is $\pi/4$.
• The area contained within the unit square in this quadrant is simply 1.
• Thus, the ratio of the area of the quarter-circle to the area of the square, which we denote R, is $R=\pi/4$.

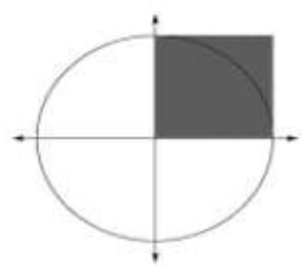

Figure 1: A simple Monte Carlo simulation, to estimate the numerical value of π.

• The numerical value of π then can be found from $\pi=4R$.
• The problem of computing the value of π has been transformed into the equivalent geometric problem of determining the ratio of the two areas, R.
• A Monte Carlo simulation can be used to find R by modeling an equivalent physical system.

• Imagine throwing a large number of darts randomly at figure 1 such that every dart hits within the unit square.
• Then we count the number of times a dart hit within the quarter-circle, n_{circ}, and the total number of darts thrown, n_{total}.
• Then the desired ratio of the two areas is $R = n_{circ}/n_{total}$.
• We can simulate this dart-throwing experiment by generating two random numbers, u_1 and u_2, for each dart thrown, such that u_1 and u_2 are both uniformly distributed between 0 and 1.
• If the distance from the origin of the point defined by $(x,y) = (u_1,u_2)$ is smaller than the radius of the circle, that is, $u_1^2 + u_2^2 \leq 1$, then the simulated dart has hit within the quarter-circle.
• By repeating this process, a large number of times, we can theoretically compute the value of π to any level of precision desired.
• The key to this type of Monte Carlo simulation is identifying an appropriate physical model for the system being studied.

We can recall the complete procedure we can actually initialize the circle points, square points to 0 & we can actually generate the random points u_1 & u_2. We will actually then calculate the distance & this distance can be measured by this formula & if it is less than equal to 1, we can increment the count of circle point.

The count square count or the total number of counts will also be incremented & we an actually repeat it for certain number of intervals.

Let's suppose we are performing this experiment for 20 points, so we need to repeat this experiment for 20 times; each time we will first generate the random number than we will calculate the value of d or distance, we will compare it with 1 if it's less than or equal to 1 we will increment the circle count. The total count will always be incremented & we will keep track of the no. of iterations as well.

Once all the iterations are completed or suppose we have generated results for 20 points we can use our formula for the estimation of value of pie & i.e. $\pi=4R$

Where R can be replaced by previous slide formula.

But accuracy can only be obtained by repeating this process a large no. of times.

Try this for 10 random numbers for u_1 & u_2

Keep in mind u_1 & u_2 should be between this number.

The more you increase the number of samples the better result you can obtained.

Example Problem

Monte Carlo Simulation can be used in several application areas including forecasting models, financial & cost analysis, project management etc.

Choose any application area & provide a detailed example of Monte Carlo Simulation.