Distributed Computing

A-08. Discrete Event Simulations

Simulating A System

- We’ve seen an example of the type of analysis needed to evaluate how a system will behave
- To be theoretically tractable, **simplifying assumptions** must be taken
  - E.g., memoryless/exponential distributions, single server
- We want to look at what happens when we **drop** those assumptions

Simulations

- To evaluate distributed systems:
  - Mathematical models
    - Good: “hard truths” for the modeled world
    - Bad: needs simplifications: the modeled world is not the real world
  - Experiments & measurements on real systems
    - Good: evaluating the “real thing”
    - Bad: overly focused on implementation details, expensive, limited
  - **Simulations**
    - Good: (to some extent) scalable, cheap
    - Bad: trade-off between scalability and precision

Discrete Event Simulation

- A queue of events, sorted by the time at which they happen
- A system state
- Iteratively:
  - Select the first event in the queue
  - Update the state
  - Add any new event triggered by this one to the queue
- Repeat until the queue is empty (or maybe a special STOP event is reached)

Our Code

- Plenty of stuff can be done
- SSE students will extend it and write an assignment
- `discrete_event_sim.py`: library file
- `sir.py`: example complete simulation code
- `queue_sim.py`: simulation to complete

Goals

Do what you can in the lab hours

1) Complete `discrete_event_sim.py` (check that `sir.py` works)
2) Complete `queue_sim.py`
3) Familiarize with the command line, run experiments, save the results on a CSV file
4) Check the plots
5) Think about how to reproduce the plots in the slides and compare them with the theoretical values