Performance evaluation:

Performance evaluation suggested reading: heidelberger and lavenberg, "computer performance evaluation methodology"

Performance modeling and measurement is needed through the entire life cycle of the system: design, debugging, installation, tuning, data collection for next system.

performance evaluation coverse these areas:

- measurement
- analytic modeling
- simulation modeling
- tuning design
- improvement

good work in performance evaluation requires a good substantive understanding of the system under study. you can't just arrive with a bag of tricks (e.g. queueing theory) and do something useful

Measurement:

- advantage of dealing with something "real" has all interactions, which would tend to escape the model
- Disadvantage of time and effort to get data (inc. facilities needed).
- Not that many published measurement studies

Hardware Monitoring:

- use some sort of hardware monitor (like a logic analyzer) to collect and partially reduce data
- data is pulled off system with logic probes
  - note that signals have to be available. (Not from the middle of the chip). May have to design in probe points.
- Signals can be used to count events, generate traces, sample, etc.
- Hardware monitors typically have some hard logic (such as counters and gates) backed up by some programmable control (such as a minicomputer).
- Hardware monitors are difficult to get, expensive, and hard to use. They are seldom used except by vendors and large computer centers.
- Samples should be taken at random times, not regular intervals. The latter will fail to get correct measures of events which occur at regular intervals which are multiples or submultiples of the sampling interval.

Software Measurements:

- Code running on system is instrumented.
- Can put counters on signallers in OS code, compile it into source code to be studied. Can sample at timer interrupts
  - Note that sampling wont sample something which isnt interruptable
- can use built in profiling facilities provided by some compilers
- can instrument the microcode to collect data, if machine is microcoded.
- Automatic facilities like compilers and profiles can do a lot of this
- can generate significant overhead, like 20% or more.

Hardware Counters

- Modern CPUs have counters built into the hardware. Can be set to count various things: branches, misses, cycles, instructions, etc.

Multics Measurements:

- Sampled using timer interrupts
- Using hardware counter for memory cycles.
- Had external I/O channel which could be externally driven to do useful things like read certain regions of memory.
- Used *remote terminal emulator* to drive system.

Diamond:

- Diamond is an interal DEC measurement tool - a hybrid monitor. Hardware probes read the PC. CPU mode, channel and I/O device activity, and system task ID. Software gets the user ID. A minicomputer reads the data and does real time and later analysis. It can also generate traces.

IBM's GTF (General Trace Facility)

- IBM - generates trace records of any system events, such as I/O interrupts, SIO's, opens, traps, supervisor calls, dispatches, closes, etc.
- Designed for debugging; generates lots of useless data and not as much useful data as you would like.
- IBM's SMF (Software Management Facility)
  - Generates records for assorted events - jobs, tasks, opens, closes, etc.
    - Designed for accounting and management; Also generates lots of useless data and not as much useful data as you would like.

Some mainframes have console monitors which generate real time load information and measurements (queue lenghts, channel utilization, etc)

Workload Characterization:

- Important part of any workload study - you must know what the workload is.
- Three types of workloads for Performance Experiments:
  - Live workloads - good for measurement but poor for experiments (it's uncontrolled)
  - Executable workloads consisting of real samples
  - Synthetic executable workloads - can be parameterized versions of real workloads
    * A synthetic workload may be needed as a projection of a future workload
- To characterize a workload:
  - Decide what to characterize. This is not easy, many published papers are not interesting because the authors tried to characterize *everything*
    * Figure out how to characterize those items
    * Figure out how to get the data
    * Get the data
  - Exploratory data analysis
    * Cluster analysis (find a vector)
  - Statistical Methods
    * Means, variances, distributions
    * Techniques such as linear regression and factor analysis quite useful.
    * Can do statistical analysis on data to see if various models fit.
    * Seldom used - little intersection between class of competent experimental performance analysis and competent statisticians

Analytic Performance Modeling:

- Build analytic model of some type of system of interest.
- Calculate factors of interest as function of parameters
- Models tend to be queueing models, stochastic process models.
  - Most of progress in queueing theory in last 30-40 years is due to computer system modeling

    * Advances are:
        · Class of queueing network models which are easily solved
        · efficient computational algorithm for these solutions
        · good approximation methods for systems not easily solved

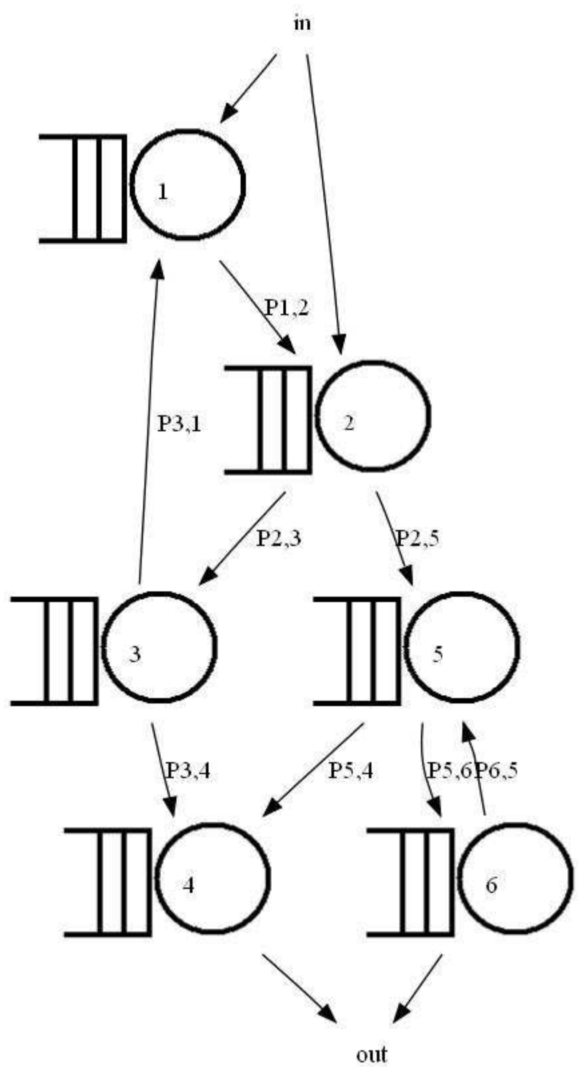
Pros/Cons of Analytic Modeling:

- They are good for capacity planning, I/O subsystem modelling, preliminary design aid.
- Capacity planning is a big area of application - measure and analyze current system, set up validated model of current system and workload, project changes in workload and see what sort of system design will handle it.
- Analytic models do not apture the fine level of detail needed for some things, such as hardware design and analysis (e.g. caches, CPU pipelines.)

Queueing network is a powerful technique in analytic modeling

- Major Components of Queueing network are:
- Servers
  - Customers
  - Routing

Graph of BCMP:



Many types of queueing networks can be easily solved, such as the following (called BCMP - Baskett Chandy Munts & Palacios):

- Customers can have a type, T
- Routing can be of the form p(i, t1, j, t2) where i and j are servers, t and t2 are types.
  - Servers can be:
    - FCFS exponential, with service rate a function of queue length
    - Processor sharing

    - Infinite server
    - LCFSPR - last come, first serve, preemtive resume, any distribution
  - Solution is a product of terms for each service station

- If network is not of BCMP type, it probably can't be solved exactly. There are approximation methods that can be used in these cases, though.

Simulation

- Types

- discrete event simulation

  - Trace Driven
    - Random Number (Stochastic)
  - continuous simulation (e.g. a differential equation) (not used for computer system modeling)
  - Monte carlo models (e.g. sampling)

- Simulation model has these components:

  - A model of the system which has a state
  - A set of events which cause changes in the state
  - A method for generating such a sequence of events.
  - A measurement component, which records the statistics of interest

- Discrete Event general simulation model:

  - Events come from *event list*. (Events can come from trace).
  - Next event is taken off list
  - System state is updated
  - Event list is updated. (events added, deleted, their times changed)
  - Statistics are accumulated
  - Next event is obtained from event list

- Example: simple discrete event simulation of M/G/1 queue

- Events come from trace and/or random number generator.

  - There are special languages for simulation (GPSS, SIM-SCRIPT, GLASP, SLAM, SIMULA)
  - There are simulation modeling packages: RESQ (IBM), PAW(UT Austin), QNAP (INRIA - Potier)

- Analysis of simulation output:
  - Regenerative simulation - find regeneration point, do standard IID statistics
  - Time Series Analysis - do time series analysis (e.g. auto-correlation analysis) - also called spectral method
  - Repeat entire simulation run
  - Very long runs
  - Batch means - take long samples and conider them IID (Independent, Identically Distributed

Back of the Envelope methods:

- Very often, you can get a pretty good number on a piece of paper no bigger than an envelope.
- Examples: Calulating the volume of water leaving the missisipi per hour.