Lecture # 23 CS 695

the experiments story

motivate. formulation. design. build. deploy. measure. reason. report

Q — question
A — accuracy
R — reproducable
C — completeness

1. question
2. setup
3. modality
4. execution & reporting

(*) **setup**

hardware
software
workloads

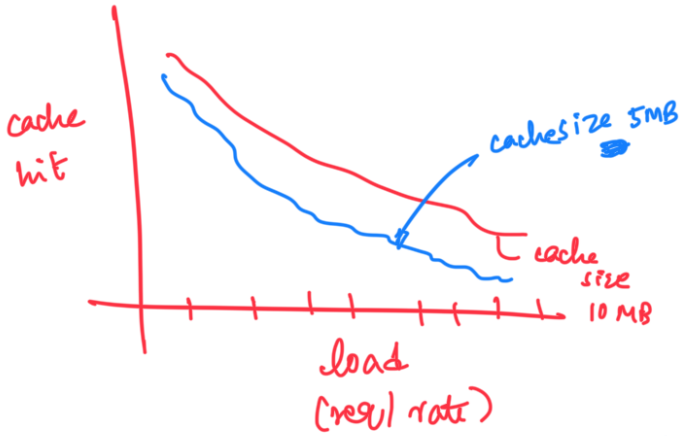
} (i) mimic / representative of "real" conditions of interest.

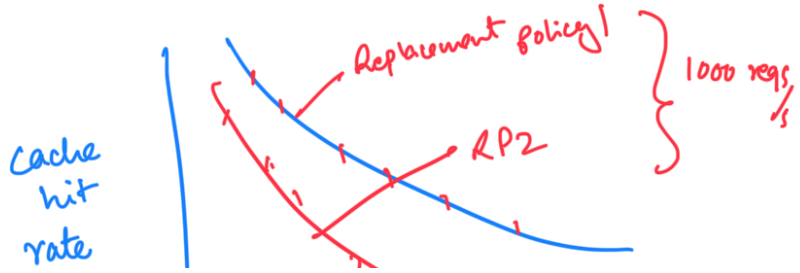
(ii) reproducability

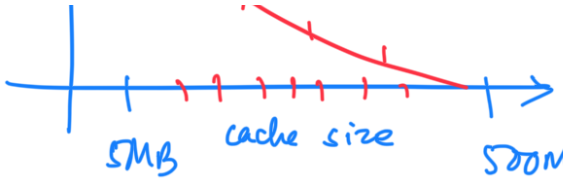
(iii) apples vs apples.

caching system

- cache size
- cache replacement policy
- object size dist^n.
- req. dist^n.
- IO blw to storage







(3) modality

(how to setup the system for expr).

(i) model (ii) simulators (iii) emulators

(iv) prototypes

(v) production systems.

- analytical / theoritical representation of a system.
- no physicality (program n/w)

eg: (i) queuing model

char. of arrival

char. of the server

'μ' ← service rate

- expected probabilistic analysis

eg. what is expected queue length given λ & μ?

(ii) setting up optimization formulations.

eg: LP — linear program

ILP - integer LP

objective fn. min # servers for VMs

constraints. $R_{ij} < C_{jk}$ ← resource req. i of VMj should be less than capacity of resource

(i) simulation

(ii) Simulators

~ time / events management loop

~ Keeping track of important system state.

on node K

– no "actual" work.

eg: webserver

– there need not be actual disk / web pages / HTTP responses

basic simulator loop

```
while (1) {
    e = getnextevent (Q);   (*)
    processevent (e);   (*)
}
```

Q is a queue of time-ordered items (event: requests)

processevent: dequeuing & possibly adding events to the queue — ordered

is system logic + update to system state.

fn: (*) addEvent (e, Q); — ordered addition

eg:

```
if (cachehit)
    cachehit ++;
else {
    cachemiss ++;
    t = readdisk (e0);   (*)
    addevent (e', Q, ct + t)
}
insertcache (e0);
```

e0 — object

ct — current time

t — service time of disk IO

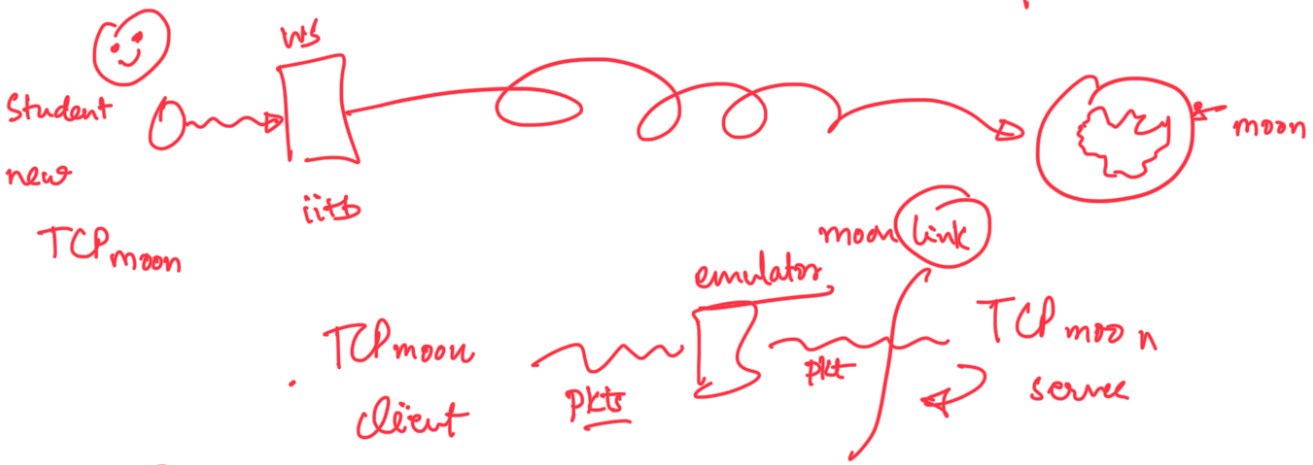
(*) this based on a model / profile / lookup table.

insertcache (e0); ← system state maintenance

(iii) emulators

~ emulate system behaviour via a set of different

operations.



PoC

(iv) prototype

(v) production

② execution / types of exp^n.

~ types ——— correctness, comparison, causality,

~ independent vs dependent variables

parameters — metrics

* only one parameter should change across experiments.

instrumentation overheads.

~ ovhds. to measure system behaviour.

~ cannot overtly influence the system itself.

④ Observations & inferences

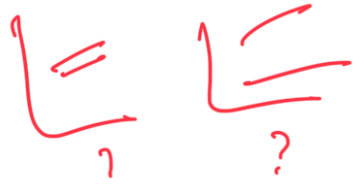
- go back to QARC
- if question not answered or needs further analysis

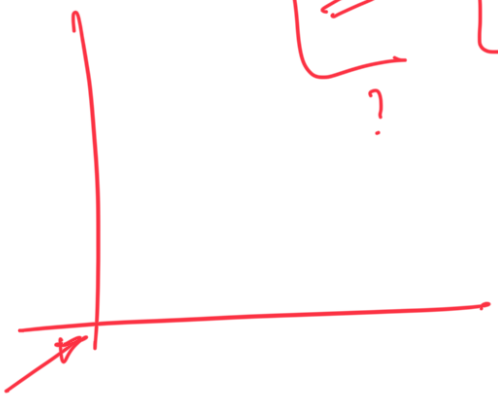
go back to step 0.

⑤ representation

~ tables, figures, graphs

- cdf
- pdf
- Scatter plot.
- 3D





always start at y=0