Cloud Computing

Quality of Service (QoS)

- There are many, many ways to measure “quality” of software
- Some are:
  - Code
  - Design
  - Requirements, Specification
  - Test
  - Security
  - Performance
  - Reliability

Quality of Service (QoS)

- There are many, many ways to measure “quality” of software
- Traditional Software Engineering:
  - Code – number of lines (LoC), defects (bugs) per KLOC, number of modules (or functions), size of functions, many more
  - Design – inflow, outflows, state
  - Requirements, Specification – number, conflicts, missing
  - Test – test coverage, cases

Quality of Service (QoS)

- There are many, many ways to measure “quality” of software
- Also interesting:
    - Security – difficult, use tools such as penetration test, static code analyzers, design analysis
    - Performance – Set goals, measure over time, vary loads
    - Reliability

Quality of Service (QoS)

- There are many, many ways to measure “quality” of software
- Reliability

- How can one “measure”?
- Good? Pretty reliable? Stable?

- Verbal descriptions are not “quantitative”

Quality of Service (QoS)

- There are many, many ways to measure “quality” of software
- Reliability

- Run over some time, collect “stability” or similar information

Quality of Service (QoS)

- There are many, many ways to measure “quality” of software
- Reliability

- When is a system not available? (Crashed?)
- When it doesn’t respond “in time”
- (could be 5 minutes, could be 10 seconds)

Quality of Service (QoS)

- Reliability

- For example, for a system we are watching:

- Where UP means the system is available, in time (what ever we set)
- DOWN means either the system is not available or too slow

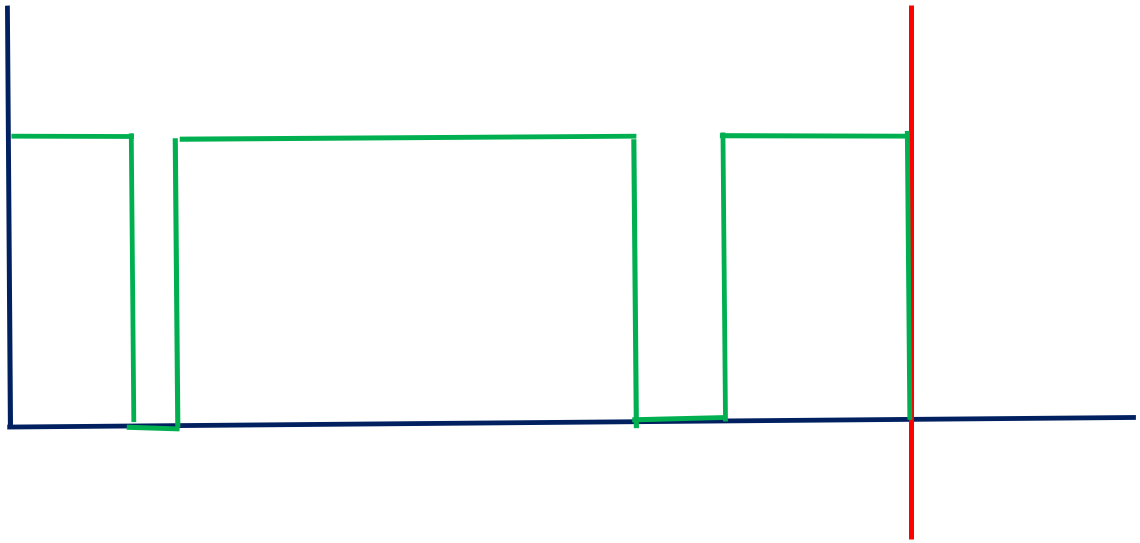
Quality of Service (QoS)

- Reliability
- For example, for a system we are watching:

Hours	Status	Amount of Time at Status
0 to 10	UP	10
10 to 11	DOWN	1
11 to 61	UP	60
61 to 64	DOWN	3
64 to 100	UP	26
100	DOWN	Then Stop Measuring

Quality of Service (QoS)

- Reliability
- For example, for a system we are watching:



Quality of Service (QoS)

- Reliability
- For example, for a system we are watching:

Hours	Status	Amount of Time at Status
0 to 10	UP	10
10 to 11	DOWN	1
11 to 61	UP	60
61 to 64	DOWN	3
64 to 100	UP	26
100	DOWN	--

Quality of Service (QoS)

- Reliability
- For this example
- “Mean” times (arithmetic averages)

- Mean Time To Fail
- MTTF = (10+60+26)/3

Quality of Service (QoS)

- Reliability
- For this example
- “Mean” times (arithmetic averages)

- Mean Time To Fail
- MTTF = (10+60+26)/3
- Mean Time To Repair
- MTTR = (1+3)/2

Quality of Service (QoS)

- Reliability
- For this example
- “Mean” times (arithmetic averages)

- Mean Time Between Failures
- MTBF = ((10+1)+(60+3)+(26+0)/3

Quality of Service (QoS)

- Reliability
- For this example
- “Mean” times (arithmetic averages)
- These are “means” measured in hours
- Could be measured in days, months, minutes, seconds, etc

Quality of Service (QoS)

- Reliability
- “Mean” times (arithmetic averages)
- These are “means” measured in time units
- May offer insight, often does not

Quality of Service (QoS)

- Reliability
- Availability (as a fraction, or percent)

- Out of time measuring, what percent of time is system “available”?
- Usable?
- Availability = (100-(1+3))/100

Quality of Service (QoS)

- Reliability

- But even availability doesn’t mean the system is actually used during that time
- What if we know what the “need” or “demand” for that system is, over time
- Perhaps lower demand at night, than during the day

Quality of Service (QoS)

- Reliability
- For example, for a system we are watching:

Hours	Status	Time	Demand
0 to 10	UP	10	100
10 to 11	DOWN	1	100
11 to 61	UP	60	500
61 to 64	DOWN	3	100
64 to 100	UP	26	200
100	DOWN	--	

Quality of Service (QoS)

- Reliability
- For example, for a system we are watching:

- Total demand = 1000

Quality of Service (QoS)

- Reliability
- Availability on Demand (as a fraction, or percent)
- AoD = (100+500+200)/1000

Quality of Service (QoS)

- Reliability

- Availability = (100-(1+3))/100
- = 0.96 or 96%

- AoD = (100+500+200)/1000
- = 0.80 or 80%

Quality of Service (QoS)

- Reliability

- Frequently, services (such as Amazon or Google) will either give
- Uptime (part of MTTF)
- Availability (but not AoD)

- But other metrics (MTBF, AoD) can be estimated externally
- And frequently are

Quality of Service (QoS)

- Reliability

- Can’t get better unless you know how good you are now

- Set some goals: AoD or MTBF
  - Try to meet goals   and/or
  - Try to have steady improvement

Last

-