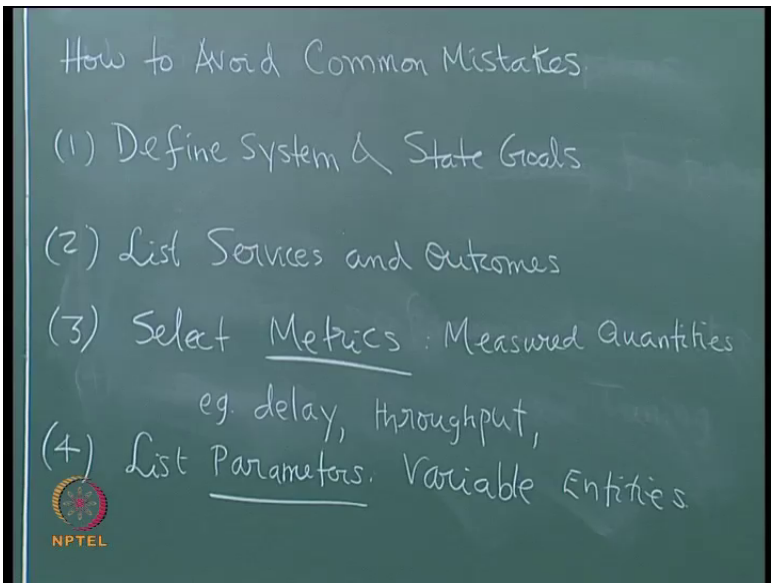
Performance Evaluation of Computer Systems

Prof. Krishna Moorthy Sivalingam
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture No. # 02
How to avoid Common mistakes

(Refer Slide Time: 00:11)



(No audio from 00:11 to 00:31)

The title of the chapter is common mistakes and how to avoid them. So, it is mostly reading stuff, but I just highlight. So, it lists about 22 common mistakes that people make, but this chapter also then presence how to avoid them. So, I just list those 5 or 10 points. So, what we need is a systematic way of how to go about this performance evaluation process. Those 6 of 6 steps that I showed, will expand on little bit more to have the specific list of step that one should follow. So, this will shown more or less like software engineering first part, because in software engineering what is the first thing that you do.

(( ))

Then

(( ))

That you do in software engineering project.

(( ))

So, you should know, what is that you want build likewise here you should first define what is that you want to build, and also the design goals of the system, or you try to design for high utilization or low response time; what is it; what is the system goal; so, that should be clearly stated. Because only based on that you will try to define your metrics, and define your experiments and so on. So, first is, define the system, and state the goals of the system. So, if you are in the banking industry, and the bank decides that it wants to automate all this processing we have more or less many systems now they do that; Let us say that that did not exist. In the bank suddenly decides let us try to give internet access to everybody. So, you can do all your transactions where the net or whereas some phone and things like that.

Then we have to decide that is the overall stated code, but what is the main (( )) system objective; specifically what you want to do, you want to reduce the time that the user spends in the bank or increase the time the user spends in front of the computer. Right now it is the other way around we spend less time in the back, but more time form the computer, especially some banks I would not name them; you sit there and keep waiting waiting for the page to load, why? Because somebody is somewhere did not do the right performance evaluation, bottleneck analysis, did not given the parameters correctly, they did not have the right number of servers in the system, then your user delay gets to be much higher. So, you should be very clear I want to reduce the time that the user spends on a particular transaction. If you go to the bank if it takes half an hour in the olden days just to withdraw 500 rupees from the bank.

Now, the goal is to make that you know much smaller than that. If you go to the a t m hopefully it is less than that, 10 minutes you can be done with the a t m transaction. So, that is what we mean by state the goals of your system itself, because the goals try to what performance objectives we have, and based on that everything else has to be defined. If do not care about response time you only care about processing large number of transactions per second or something like that then you will can define your metrics appropriately, if it is response time let us to be one of the metrics that you measure. So,

that is the first step is to design the overall goals. And then list the services and the outcomes these also similar when this kind of generic (( )) to what we say.

So, this is the system high level definition and so particular services. So, what is the particular service, if it is a; let say RMI, RMI, RPC, RMI we know that term. So, either remote procedure call or remote method invocation, that is the system that you want to define new kind of RMI technique or RPC technique. So, what is the goal here you can say, so, the outcome, the service offered is the user will be able to invoke a process on a different machine, and then get the response back. So, therefore, what is the service offered by the system, and the expected outcome is that, you give a program, you give the necessary whatever you (( )) with the code, and the parameters, other system will simply run it, and get you back the results. So, that is the expected outcome. So, these are also this the, these two fall more or less in your software engineering domain, only think is we also try to understand; what the key performance metrics are. From a performance point of view, what is important that is the customer has to come out and tell that this is more important to me than anything else.

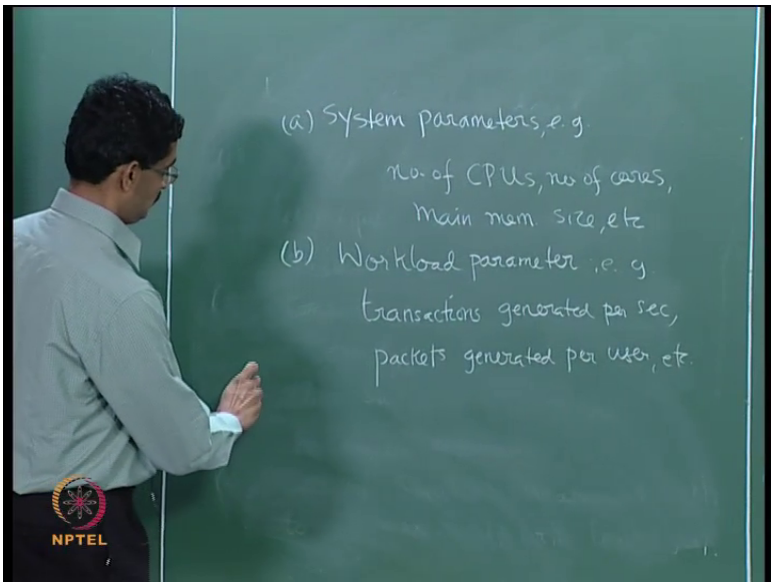
So, once I know what the system does, and what the various services are then I start defining my metrics. So, now, we start defining some terms. So, metric is a term which I find in thesis, and reports, and so on; which is very easily confuse with parameters. People talk about parameters when they mean metrics, and vice versa. So, metric is something that is measured in a system that is something, that is a measurable quantity of a system. So, delay is a measurable metric, I sent packets to this system, and I measure the delay, that each parent, each packet is experiencing, that is a metric of the system. So, metric is; these are the measured quantities.

So, for example, delay or you might say throughput, through put how many transaction per second can a database server support, that is from the server point of view that what they report, if you look at your bus specifications in the in a system it will tell you, how many transactions per second or how many transfers per second, this what able to give your g t per second and so on. So, that is something that is that you can define as many metrics as you want, but as the performance evaluation expert you are supports to know what is more important, and what will necessarily capture. The essence of these goal, so, this, these metrics have to be tied to your goals and outcomes of the system, otherwise

you random measuring the wrong things and then you not know how the system actually is behaving.

Then you list the parameters. So, parameter is something that is will control the behavior of the system that will have an impact on the systems performance. So, this parameter is the variable; the independent variable. You can have several parameters on a system if we look at a typical computer, the size of memory that is a parameter, if you want to measure how many transactions per second can this database server have then you say, come on a build a system S, but what is the memory. So, memory could be 1 G B, 2 G B, 4 G B that is a parameter; that is a controllable parameter, it is something that you have control over or the CPU to (( )) single core, 2 core, 3 core, 4 core whatever it is. That is also the parameter there is the number of cores on the system is the parameter and so on. So, here again we have two types of parameters. So, these are the varied entities. So, something that defines a system the system is several sub components, in each of the sub components you are defining in terms of these parameters.

(Refer Slide Time: 08:28)



So, the two types of parameters you can classify them as system parameters. So, for example the number of CPUs or the number of cores, main memory size, so, this is from the system perspective. These are all the parts of the system or if it is a switch then you say the switches operating at you know so many links, and so much of back plain capacity, and this is the per link through a bandwidth (( )) available. So, that is all

definition of what the system is. Then there another type of parameter, what can that be? One defines a system, one defines the

Environment

The user

User

The user; the usage environment, so, the user of the system is the next one. So, that is your work load. So, what is that, these also you would vary, so, you would try whether my system works for 10,000 transactions per second; what is the response time or 100,000 percent transactions per second; what is the response time. So, that is what is being offered to the system were the work load with the system processors, you need you can vary that too number of transactions or number of short transactions, number of long transactions or number of users, if it in the case of enough it is, let us say medium access protocol you trying to evaluate aloha, either 10 users per system or 50 users per system there is actually system level parameter or lambda, if the lambda, the traffic offer per user, the traffic generated per user, you may have 50 users that is a system parameter, the system has 50 users, but each user generates certain amount of traffic, that is variable. That is a work load that is offered by the user themselves. So, this could be; so example; so there is a database or the number of packets generated per system. So, this is have you would when you write your performance report in mtp or btp. You would clearly say that these are the metrics, these are the parameters. In fact, your introduction section, introduction chapter you normally have to say these things. These are the system parameters that I really considered important and so on.

Sir, can a parameter be both system and workload?

I would taught about that there will be workload is what is offered to the system one is trying to characterize system itself. So it, I am not able to think of an example where it can be both. We will think about it if you find something.

(( )) parameter something like what we fix initially, and metrics what the how could be want.

Yes, so parameter is what you vary. So, I want to study the system for 10 users, 50 users, 100 users, and I want to measure the average response time. Usually you have this load there is a (( )) what is the load on the system versus the delay on the; the delay that is measured. So, delay is what is measured, and load is what you control. So, one is the controlled one is the uncontrolled thing.

Sir, make a response time where will you put it the system parameter.

Response time is a metric. It is what you measure. So, what is the offered load, and what is the; how to the system handle this load, and what is the output. So, if you go to your mess, your hostel mess then the system parameter is; I am not sure how your mess operates, hopefully they have several counters, and within each counter may be sub counters, if we have north Indian, south Indian, and so on. They have a certain number of counters for each of those, that is a system parameter. The offer load is the number of students that show up per hour whatever, and that depends upon the total number of students in campus. This campus is design for half this number of students, then suddenly we have double the number of students, then we have to go and scramble, and increase the number of those counters that alerts that the serve people.

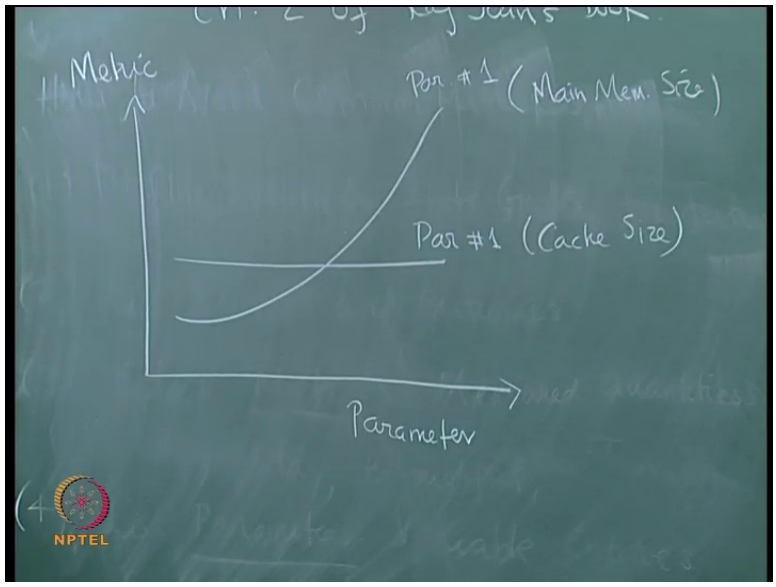
Otherwise the lines are very long which is probably happen in the hostel zone we are trying to accommodate more students, but maybe not so the banks where the number of tellers has been the same as it has been for last 20, 30 years, because that is a if you know that you have to add more the people, but various reasons prevent you from manning you know that your system performance is going to vary. So, there your number of tellers, number of counters is your system parameter that is controllable, I can change it, and if I am not changing it is because of other reasons. But delay is something that is simply measured how long they would to stand in line before we get your, you know fees paid and so on.

Well this distinction is many times lost in many pay event research papers sometimes they do not really make that, but for this course when I say metric this is what I mean, when I say parameter I mean it is a controllable parameter. So, then this is little bit more exhaustive, you would like to see what are all things that they can vary. I can vary the buffer size in each router on every link, I can vary the buffer size, and I can also look at the cache, what is the cache size that use on a particular CPU system if I measuring a

system. But then what we find out is we have to somehow have intuition or some background knowledge or simply prediction or belying faith that all these parameters, I might have several levels, I might say that each parameter I can test for you know say from 10 to 1 million, I want to test the number of users or at the size of sorting. If in sorting array I want to look at the size of number of user, number of data elements to sort very small to very large.

Then what happens is you give on multiplying for every possible number of users I want to vary the main memory size, and the CPU number of course, and the CPU number of CPU threads available all those things you slowly start getting into a (( )) explosion. Then you start saying that wait a minute this is the too many parameters to be considered and some parameters met have no impact on your system at all.

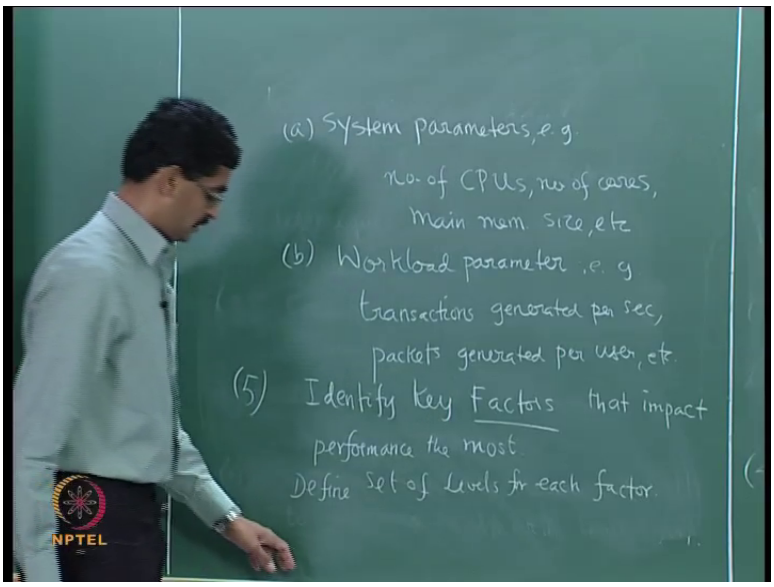
(Refer Slide Time: 15:10)



So, what you want to do is you want to have some sort of estimate, say this is your metric, and this is your parameter. So, you have to know which parameters are more important than the other ones, which ones will have significant impact compare to the other ones. And you would like to see if this is the right behavior after your experimentation after the evaluation that as this parameter value increases the metric increases like this. So, here you find the there is a reasonably substantially sharp increase in with particular parameter. So, say this is the parameter 1.

And then there is other parameter that you vary let say we can plot two different x axis, and two different two different y axis, and this is parameter number 2. Let us say this is main memory size, and this is your cache size. So, varying the cache size, your cache levels are so small that may be it makes no difference where as going memory 1 to 2 to 4 G B will make a big difference, but going from 32 kilobytes to 64 kilobytes, and what you know a bits has no real impact on your system performance. So, that is something that you need to; now that you listed all these possible variations this things that you can control and vary.

(Refer Slide Time: 16:39)

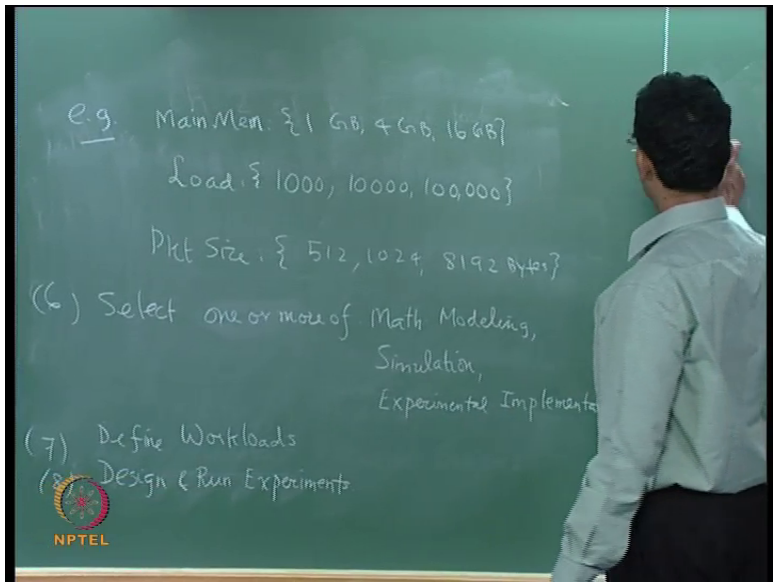


Your next step is to identify the; so, we call them as factors the most. Because that is needed to control your experimentation space, otherwise you learn up with millions of experiments which some of which give you no meaningful information at all. There is no point in running against varying cache sizes and so on. So, you should say which of these even here work load parameters, so I get a several parameters, then some parameters really may be in make no difference at all. So, then you define the set of levels for each factor.

So, level simply means the number of possible values that you want to try out. And if may these are many times continue or continuous variables. So, I could choose for example, load number of packets offered per second I can go from one packet is taken to a million packets per second increments of 1, 2, 3, 4, and so on. So that it really makes

no difference, no sense to have that many levels. So, you say 3 levels, 4 levels, low load, medium load, high load; that is adequate. So, those are the things that as part of your system performance evaluation you will decide.

(Refer Slide Time: 18:32)



So, for example, I will say that main memory, I will have three combinations 1 G B, 4 G B, 16 G B. And this is initially a guess why not 8 G B, why not 12 G B. you could have done that. But you want to keep your search space small in initially so say I am going to 1, 4, and 16. And then you would say load, let us say we are talking about packets per second. So, I will do you know 1000, 10,000, and 100,000 packets per second. So, you have now proved and all the other parameters out of the picture, you only care about what is the memory, what is the number of packets offered per user. And you only care about these three values we usually try to go may be 10 values or so. But depending on how much time is involved in each simulation run or in each performance evaluation even experiments also can take time.

So, this is again something that comes only by experience, you know that sometimes it is pointless in the internet is pointless to run 100 k packets, 100 k packet size if you look at packet size as a parameter, 100,000 packets per second is ok. But if I say I add packet size also as a parameter, then you know that there is no point looking at in a very large packet sizes. Because everything anyway gets broken down by Ethernet layer 2500 bytes or 9000 byte packets. So, you would be some more reasonable say I will do 512, 1024,

and may be 8192. And if you know more saying there only two possible packet links are there 40 and 1500 we simply try for those two packet links. So, that is all the main knowledge that comes later on.

So, this is know your spurned your search space to the set of specific parameters or then of course,

No audio from 20:35 to 21:27

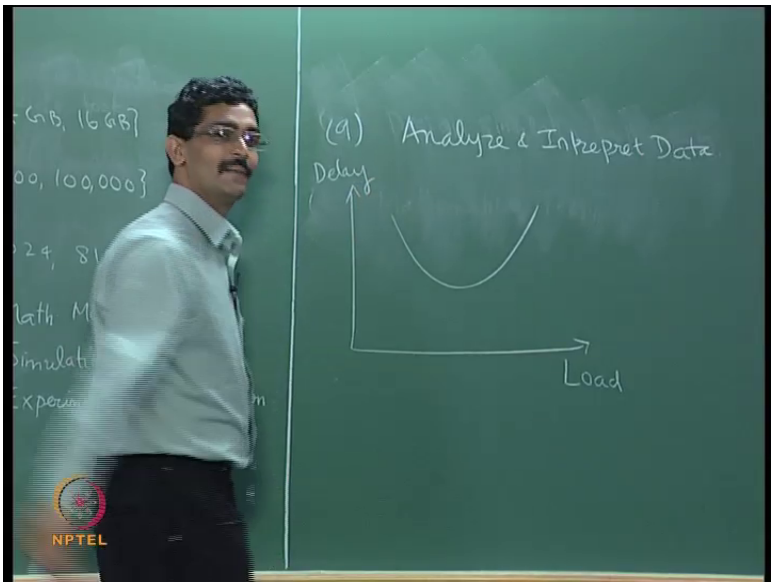
Then we choose which we have talked about this earlier on math or simulation or experimental implementation. In fact, simulation there is we are mostly talking about simulation in terms of discrete even simulation it we use to computer based simulation, but there is also analog simulation, which I remember doing a course on that several years ago, but I forget what it is all about there is book on that, I do not remember much from that at all. So, get the technique, and then you now that you have the factors, then define the workloads, I could have done define workloads before selecting the technique really does not matter. But let us say that I have designed, I have actually this is not only selecting, but also implementing. I choose the simulation platform I implemented my algorithm, TCP algorithm in there. And now I am ready to evaluate it, that time I say; what is the workload, what are the different types of combinations of workloads.

Then, so then design run experiments. So, workload is defined I can then run the experiments. And then what will be the two next steps,

(( ))

Analyze interpret data, and how many times we do not see that reports simply show a table, and then say here is my table that is all enough. I am supposed to figure out what this table is supposed to do. I varied all these parameters; I see this nice graph done. We see this in a (( )) I ask you do that in 6 0 4 assignments to when you write those various combinations this is.

(Refer Slide Time: 23:29)



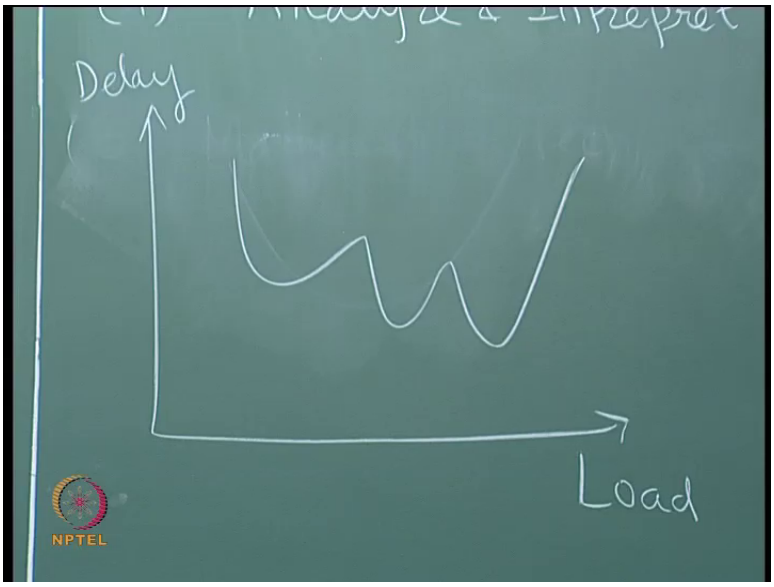
So, Analysis is more important getting the data, and then interpreting data. You should either things behave intuitively increasing memory, increase to reduce the latency; increase the number of transaction per second, therefore this is the expected. If we does not happen then what; who is that fault; systems fault, your implementations fault or is unexpected behavior we never know. You are expected to know, and something goes out of order that is that is not as that is that is something beyond what you thought, what happen. That is not intuitively agreeable then you have to explain. In many times you will find graph set look like this load or whatever it is it will go like this. This is some say delay. Increasing load decreases the delay, and then it increases again. And then people are quite comfortable look this is what I measure whatever this is an oracle this would not tell lies this is it, just take it, and go understand that this is indeed the case.

So, if you working for a company, and the company says tell me how the system behavior, and increase the number of users online, increase the number of users delay will go down. Client will not be very happy, if tell them that kind of suggestion, let us say what when try either you did your measurements of delay incorrectly. So, the common measurement of delay that one of the reasons so you have to go back, and see how do I measure delay let us say that you are talking of packets coming to a router in the measuring delay for those packets. Then you have this policy of dropping packets. So, you only count the delay of successful packets. So, what happens is when the; for some reason when the load is light many packets got drop because of errors. Let us say

simply link, channel errors and so on. So, lot of packets got dropped therefore, only two packets got through therefore, delay is very high, because the two delay is might be very large in therefore you (( )) of having large delays.

Then some of the system was better behaving, when your load was different. Some change in your parameter model that you did not notice that. Therefore, large number of packets got selected for transmission were successful, therefore delay goes down then when this is the expected, as delay as the load goes up this delay is; therefore is as the load increases delay is also increasing. But that is where you have to combine, and figure out what is going on. Simply presenting the graph, and not explaining the graph. One is to explain what is going on the graph, and the second is explaining discrepancies the graph.

(Refer Slide Time: 26:13)



The other set of draw that you are sometimes see is like this, we will see those also. And it may not be thus, but you find those specific know very clear dips ups and downs in the performance. So, then will have to say ok. Now what is going on either we have able to explain that or do more experiments. So, many times why something like this happens is when you guess, sometimes you are not running this simulation for long enough. And therefore, the system is not reach steady state. So, depending on the load, depending on the random variable; your seat for the random variable you are therefore, here the system was stable, here the system is still not it steady state. Therefore, a delay valley is will can

go up and down, this is not a smooth curve. You do not always get these smooth expectations.

But you would still have to be able to for load and delay you know that that is got to be there some expect behavior is there and your job is to find out what happened or it could be simply an implementation error for some reasons some where you are having some packets you know running out of memory those packets are not getting created. But you are simply still proceeding without you know (( )) of the system. So, that is interpretation of data is really really critical.

One more point I would like to say they find your set of levels of each factor. And then defining your work load need some guidelines definitely, I cannot really go ahead in this sets on arbitrary value let say this this this; say buffer size If (( )) either the 10 buffer or 50 buffer or 100 buffers

Yes

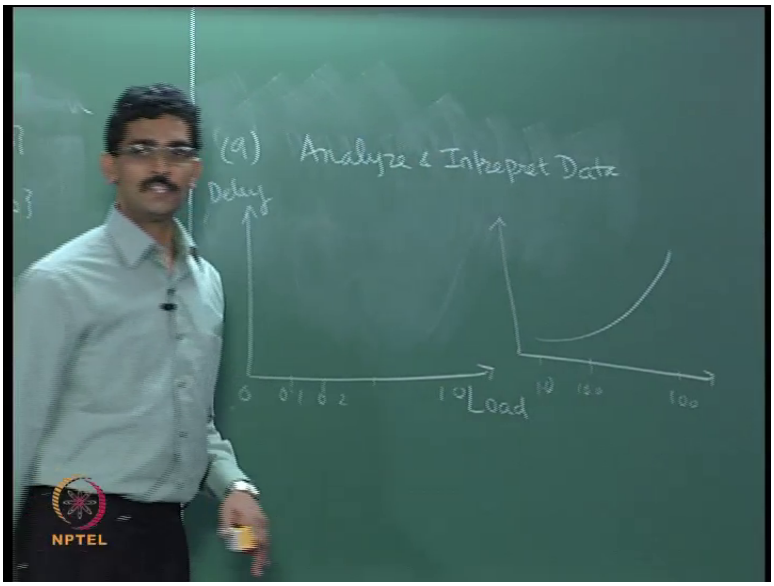
Just to analyze the (( ))

So, that that will come from (( )) that is that would be more or less or we simply blindly try out you will simply say I am going to fix all other system parameters constant I simply try varying this for say ten different values then you find that these three are adequately representative I need not go more than three such values. And that is simply that would come from experience itself.

At the interleaving between these two levels

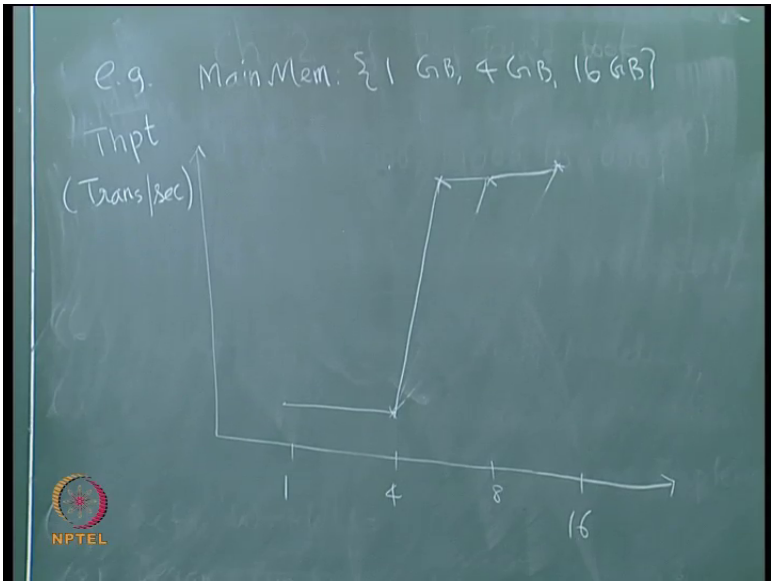
Yes, that is also important.

(Refer Slide Time: 28:22)



So, some times what we find is that people will have plots that look like this. They will try to do a log scale or there say 0 to 1.0. This will be increments of 0.1 and so on. That is one kind of where you want to zoom on a particular section of your graph because you know that lots of going on in this 0 to 1.0, and then you have the same thing where you simply go from say 1.0 to 100 in orders of whatever 10 and so on. So, that is again if you want to know something more detail in more detail about a particular section of all the work load is affecting your system you do this zooming in and then show that that you should be able to figure out.

(Refer Slide Time: 29:14)



So, then and then let us look at this 1, 4, 16 as an example let see what happens there.

No audio from 29:06 to 29:24

So, I have only these three values I thought those are good enough for my system, and I measuring throughput. So, this is throughput of the system. So, this is transactions per second. So, this will have increasing or decreasing which higher is better, lower is better; higher is better. So, I would expect that this curve. So, from here as you like this and then I suddenly say like this, then you find that is this necessarily a straight line between 4, and 16. If I put 8 where will I if I have 8 gigabyte of memory, it will be you will be should be curious know what be the transactions per second can we simply linearly interpolate, and say because 4 is this and 6 is therefore, 8 is this you cannot really say.

So, you might want to say let me see, if the 4 is here and 16 is there, in case I had with 4 gigabyte so many transactions, 16 also same number of transactions, then there is no need for further interpolation. Because you say that 4 itself I am getting the same performance of 16, this happens with parallel processing as we increase the number of CPU is you find that beyond the point there is not much improvement in your performance therefore, you simply is stabilize out. So, you want to see at 4 and 16 as a big jump, then you should definitely go back and do some more interpolation with real experiments not simply predicting. Because only you have two values it is not enough to predict. So, in this case you might want to look at 8 and 8 might also be very closely that they are, so, you get a graph like that. So, at 8 you are almost matching 16 gigabytes worth of performance.

So, you might tell the system provider or your manager that 8 GB is actually good enough for this kind of workload, that we are trying to do there is no need in going or buying 16 gigabytes of memory. But the manager might come and say why even 8 what is what happen to 6. So, where do then your question is there is start looking back then on my binary search. So, where between 4 and 8 do I find then the performance is maxing. And at that point I will draw my threshold. So, you want to find out say some where here at 6 I am more or less maxing out if you find it that is that is when that only comes by experimentation intuition. So, you start off with the intuition is saying that these are probable values then you find that I have to begin further to get more details.

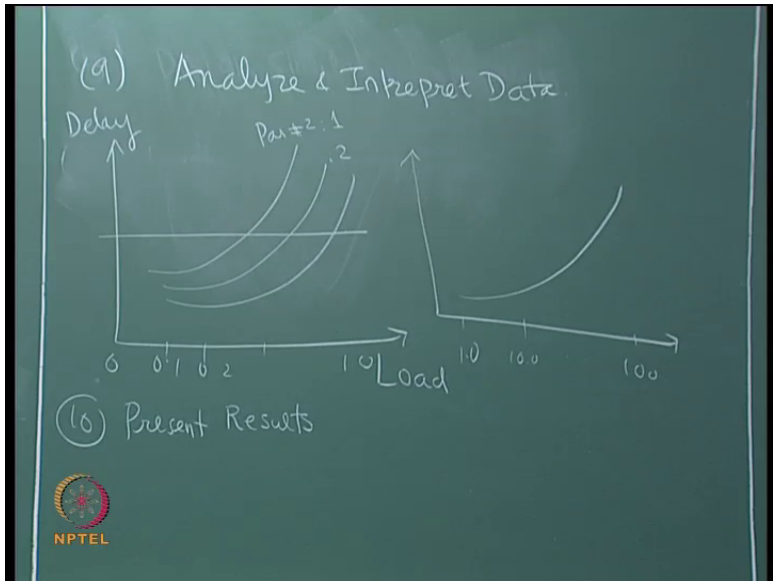
Sir, sometimes one or more of these parameters can (( ))

There is a combination of parameters.

Yeah combination of them might through to be the one which we have to experiment on that. So, how to how do we generally characterize to take that?

So, what we will try to do that is say if it as say you will sometimes of absorb these plots whether a two parameters varying on the same graph. Either you do 3 D graph which is very hard to look at visually interpret.

(Refer Slide Time: 32:19)



So, what you try to do is you say you know there is one more parameter. So, this is parameter 1 this is parameter 2. So, this is your second parameter which is a one value is this is the first value, this is second value, and so on. So, then you look for combinations of at which load, and which what parameter value to a find the cut of that I am looking for. So, let us say that you want derive to be kept somewhere here, that is the upper limit of delay that you want to tolerate, and you want to find out what should be the load as well as if I am operating at x load then I should be setting this other parameter to this corresponding value. So, that will come one of the chapters this multi factorial experimental design were are two or three in factors simultaneously affect by impact of performance.

So, in the end what you also try to do is other way what you will try to do is you fix one set of default values then vary this system against those. So, there is a lot of

experimentation that has to be done. But once you narrow down to the most important parameters then you can only play around with those two or three parameters. And then the last step is what we (( )) so for is present the results. And presenting results is also and not by itself. And there too then the default is put the values in excel, excels picks out one bar graphs put it out. And you do not worry about how it looks, does it really convey, what I am trying to show none of those are clearly. Some cases bar graphs are good, some cases you know just line plots are good, sometimes pie charts are good, there is a several variations we can will talk about very briefly at the end.

But one has to be really experiment with those, and then come out, and say also a line types, color types, so many things are there that one can vary. It is the default thing is whatever excels gives may excel really gives the lousiest like we can erase this excel really does not give you the best of graphs there are several other tools what did you, you guys used r, did use is r better than anything else. So, that is short of the template which one expected to follow (( )) template just follow this do this in a methodical manner. So, that the end your report you know that whenever you are saying a system is a better you are saying that with confidence you are saying that I have done all due diligence I have followed all these steps, I have consider all the parameters and so on.

And that that should happen it does not happen because of lack of time usually the one week before the viva is due a reports are written in which time we tell them to vary x number of parameters than draw all nice plots else it nothing doing. Viva is due tomorrow, so, do not let viva be the reason for your deadline for how you operate. Make sure that your report has this is not just for your M tech, B tech viva. You go beyond this you go to industry they are going ask you to do this. If any have to design a system they will expect you to do some more regress performance evaluation. So, more informed way in which you can tell which of these choices are better, it is always going to come up whatever system you go to, which our company that you go to, even if you start your own company you want to certificate on web service the first question is how many servers to we put or what kind of service to like buy from, what kind of parameters to what kind of service parameters do they have, what is the metric do they, which of these guys, if we have choose cloud and now we have I cloud coming into the competition.

Why should I go to I cloud, versus amazons cloud or some other cloud, how to we decide that we can just took it get multiple accounts, and then do our own testing for a while,

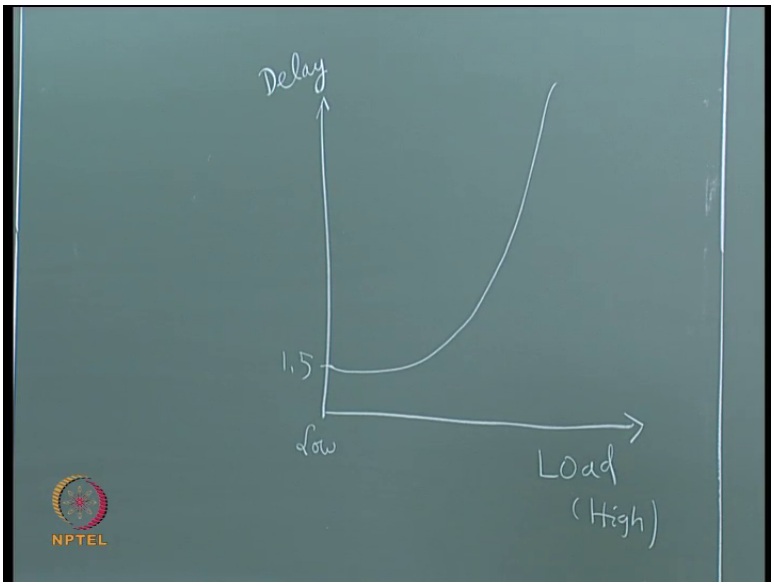
and then say yes this is better than the other. So, that is a it is more of a guess an engineering set of steps. So, now, we will talk little bit more about several of these things as we go along and some of those things it is a more of just telling you what should be done and in every domain the way you choose parameters, the we choose your metrics is going to be different that depends on what you going to be implementing, and what is more important than the other part.

We were talking about chapter 2; the common mistakes and performance evaluation, how to avoid them. Yesterday, we simply went over the list of what to or how to avoid. So, we have a list of to do list of setting state has to a goals of the system, and coming up with a right performance metrics parameters and so on. There are couples of items in that common mistakes list which I thought we should look at. So, one of them there are as a said about 22, this is still from chapter 2, about 22 items that are there in the long list not all are, will relevant to discuss now you can still go through that the (( )) later on, but couple of things we will see what we mean by that.

So, one is a this is item number 6 its unrepresentative work load, so, which means choosing a work load that does not really depict the real expected behavior of the system and what conditions will the system be used, and so you are let say choosing a load which does not capture your typical day to day expectations in terms of load of that system. So, one classic example for that is let say we are looking at medium access control protocol, all of you remember mac protocols from networks class. So, we have 2 choices one is random access based other is time division based or scheduling based. So, for random access your classic example is c s m a, aloha you can even go to aloha, csma. These are your classic random access protocols. And for scheduling based or t d m based protocols we have typical tdm.

So; if we where to compare this system of in terms of delay let us look at the response time of this system. So, have send, I have a packet to send. And let say there are 50 users in this system, and I am comparing delay for these two systems. So, what kind of load should I choose high load, low load; will it make a difference, what is the expected performance for these 2 protocols as I increase the load on the system, load is simply and let say have 50 users, each user can generate certain number of packets per second 0.1 packets per second or 50 packets per second. So, that is your load.

(Refer Slide Time: 38:45)



So, load is your x axis, this is your factor. We talked about factors yesterday. So, this is load, and the metric or measuring is the average packet delay the time it takes for a packet to be received at the receiving center. Once the sender sends the packet how long does it take before the receiver gets it on the others side. If it is aloha or slotted aloha what will be the expected behavior in terms of delay. Let say each packet is 1 unit law, so, 1 unit per packet, there are 50 users in the system then the load is very light, when each user is badly generating any, let say 0.1 packets per second. So, what will be the expected delay?

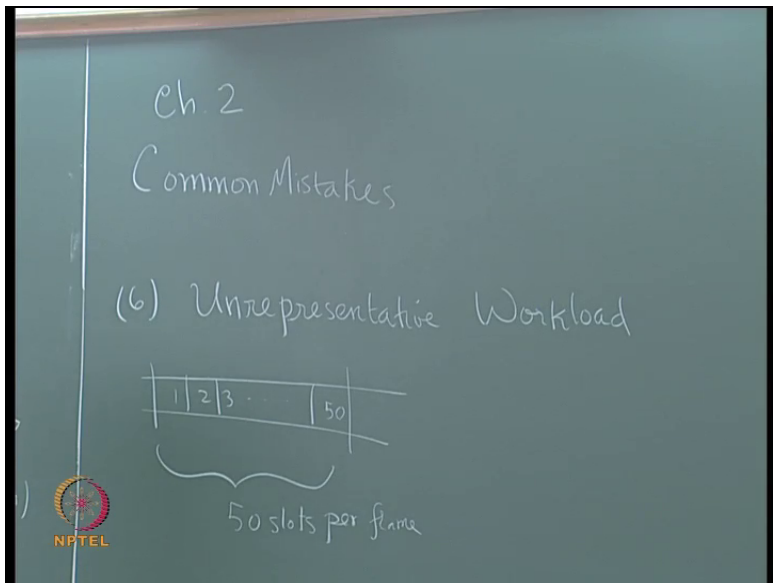
Very less

Very less, so, how less what is there a specific number you can think of load is very very light 0.001. What will be the expected delay, just one, you maximum 0.5, 1.5, if it is slotted aloha what you do; you wait until beginning of a slot comes then you send the packet. So, worst case you may have delay of 0.5 before you get the channel access, and then you send the packet, that is simply one unit law. And there are no then you assume that there are no errors in the system, no error channel errors. A channel is perfect therefore, there is no channel error only collisions lead to packets getting draw.

So, if you measure it let say something like you know 1.5 that is the average starting initial lowest possible delay. So, this is very low load, and it goes on to very high load on this access. So, for slotted aloha I will the start of with 1.5 then I say increase the load

what will happen to the delay will keep on increasing then stabilizing some point then you don't actually stabilize you want to keep going for ever. Because as collusions increase, and you start retransmissions taken then you will find that you are always colliding, retransmitting, nobody ever gets through. Therefore, your delay just keeps on escalate. This is the access delay itself. Now what about the case of tdm; there are two delays in this system, one is the delay when you get to the top of the queue, before you can actually gets it. The second is waiting time in the queue itself. So, let us look at only the time, the final access time when how long should a wait before a top of the packet, the top of the queue packet gets selected for transmission. So, what will that in the case of tdm, what will be the average delay.

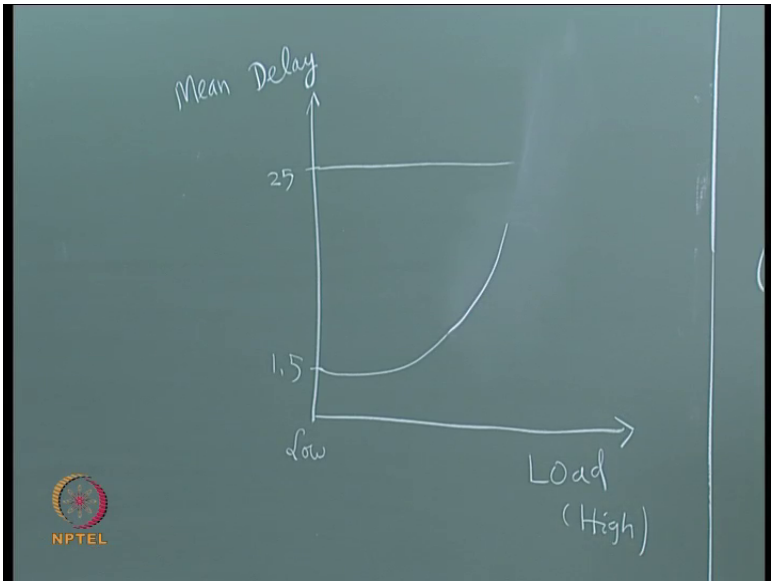
(Refer Slide Time: 41:29)



So, t d m is there are 50 if you do not remember the t d m definition. In a tdm based Mac, if I have 50 users I will have 50 slots. And we have the standard slot 1, slot 2, and so on. Remember that every packet is one unit law. So, therefore, there are 50 slots because of 50 users I have to give every user one slot. This is your classical time division multiplexing. So, this and then this frame is repeats purposively whether there is user has a packet or not, user always as a slot assign into that user. So, slot 1 is always for user 1, slot 2 is for user 2, and so on, that is a default definition of tdm. In this case what will be the delay once a packet gets to the head of the queue, Average delay, worst case its 49 best case it is 0.

So, average would be may n by 2. If have n users there n slots in a frame, so, the average what happens is, if let say 50 is your slot and your packet got to the head of the queue let say just before fiftieth slot then your time waiting time is 0 almost negligible or in the worst case your packet arrive to this queue just after your slot started or slot finished that case you have to wait for the entire cycle. So, if it arrives just before your slot; waiting time is almost 0 if it just arrives just after you have 50 unit of waiting. Therefore, the average you will see is 25 this is the guess, is there your performance evaluation starts. The first step is to look at some sort of mathematical intuition as to what should be the performance of the system. So, that is where these your classical then what you do.

(Refer Slide Time: 43:08)



So, you would find that the delay is 25, this is average delay; mean delay per user depending on when the packet come to the queue you will find that, sometimes it comes early, sometimes it comes late, and therefore there is this little bit of variation in the system. So, this is our guess, we can verify this by Simulation. Very simple mathematical computation we are saying that it should be about 25. So now, I what will you find is this going to vary with the load in the system?

No sir, no

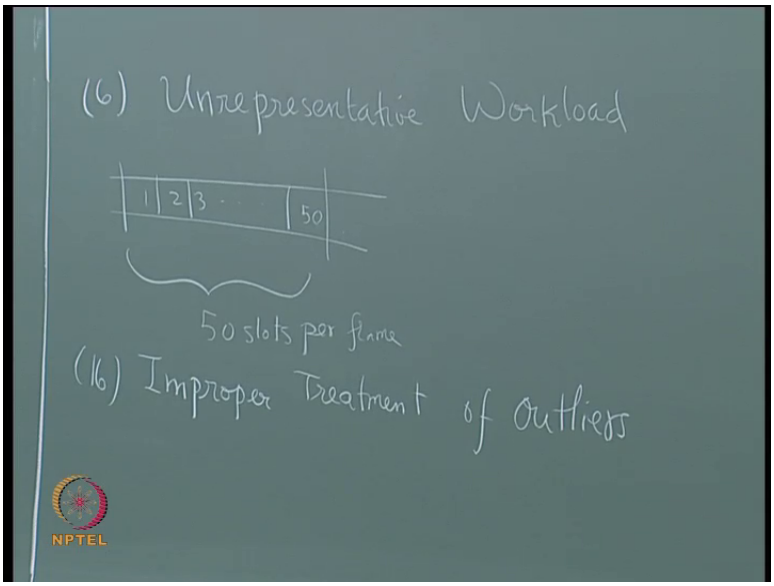
Because is independent of the number of users whether 50 users or 100 users are there you always have your slot. Only thing is you have to wait for your slot to array. Therefore, I would say that in terms of the waiting time or the channel access delay, it is

more or less constant. If I add queuing then it will become depended on the queue size. Then it will vary with load. But this is excluding queuing. If I add queues these are also become like this it will sort of go like this some point this I am not looking at the queuing, just the time taken. So, the bottom line is said light loads then the delay of started aloha will be less and the delay of the t d m will be more. So, if we want to say that I have got a better algorithm the best to do that is not even bother showing your load stops there somewhere here you say that I have simulated for from here to here like system is always better than tdm. Therefore, stop at that point and then close those then finishes the thesis say I am done when the algorithm is definitely better than this existing algorithm.

Assume that only tdm was known there was no aloha you will propose this and say I am better. So, that is what is the called as an unrepresentative work load. Your work load does not really capture the factor; you could also be operating in higher loads. But you may be in a system there are cases where this actually adequate where your number of users and I generating traffic is fairly low in which case actually aloha is better. But that is something that you should know and predict. So, when you give your report to the system for back to your over wants to know about particular system. You should look at loads that construct the entire spectrum. So in fact, aloha was one of the early protocols this was design for slot light access and they knew that this is not the best protocol. We will see later on time permitting 18 percent, 36 percent, maximum throughput in aloha slot aloha we can actually derive those things.

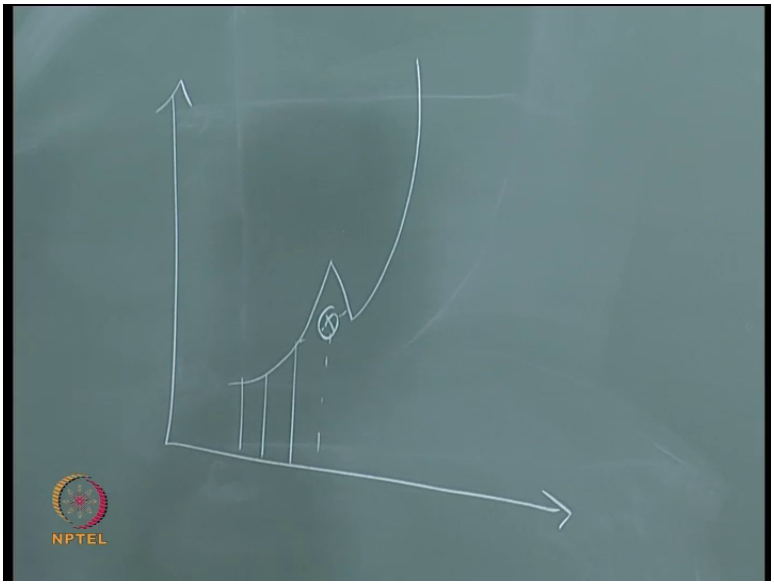
But they knew that the load of the system is very light for that light load lightly loaded system aloha is better. But if you are going to say I am doing the comprehensive evaluation of two systems you should look at the entire all those possible combinations of work load. So, that is I want classic example. Then the second is over looking important parameters all those things we can look at later.

(Refer Slide Time: 46:00)



Then we look at 16. This is with respect to out layers in the data, improper treatment of out layers. So, out layers are; when I am taking a sample of some metric whole different load values and so on. You suddenly find that there is some numbers which do not really fit the trend. You know some expect trend is there as load increases, delay increases, and so on. But sometimes there is you now ups and downs in the graph. We saw that briefly yesterday.

(Refer Slide Time: 46:39)

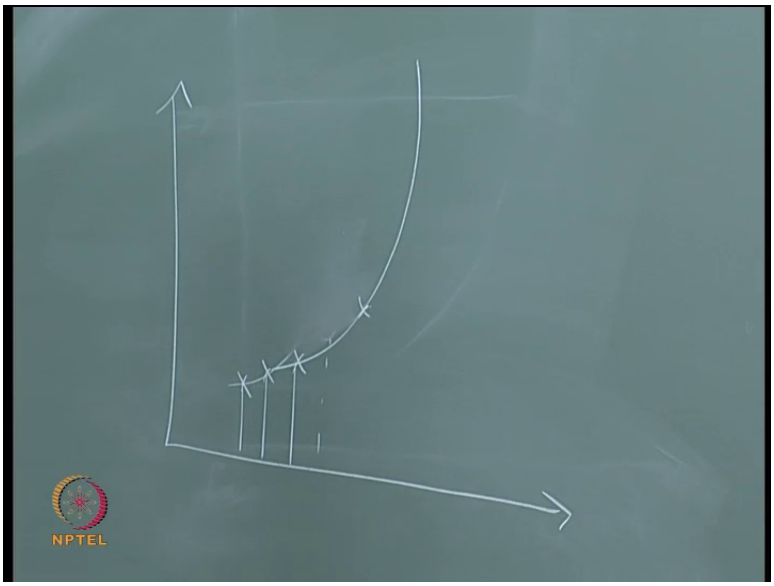


So, out layer could be; so again the trend is normally like this it goes up, it goes up, but it suddenly comes down and it goes up. There is one value where this expected trend is broken, and after that it again starts catching up. So, this is an out layer with respect to the rest of the data. It is breaking the trend that we expect to see. And the normal thing is go back and see something wrong with the code, something wrong with the load, for this particular value of load value why is the system behaving. So, your expected to go back and check your; because theory met show you that this actually the smooth curve. But in simulation you finding that there some problem. Even experimentation also there is some particular problem. So, this is in not only simulation you could be running experiments on a particular system.

You are comparing two sorting algorithm, algorithm a, algorithm b, and you find that algorithm a is generally giving you these you know consistently increasing values, and when increase the size of your list sort, the list should be sorted, and therefore, you suddenly find that there is this one jump. A one jump here because what should be what happened is let says that these are the different value. So, where is the out layer here, I expected the value to be somewhat here. And the value is gone there, then you can need best thing to do is ignorant.

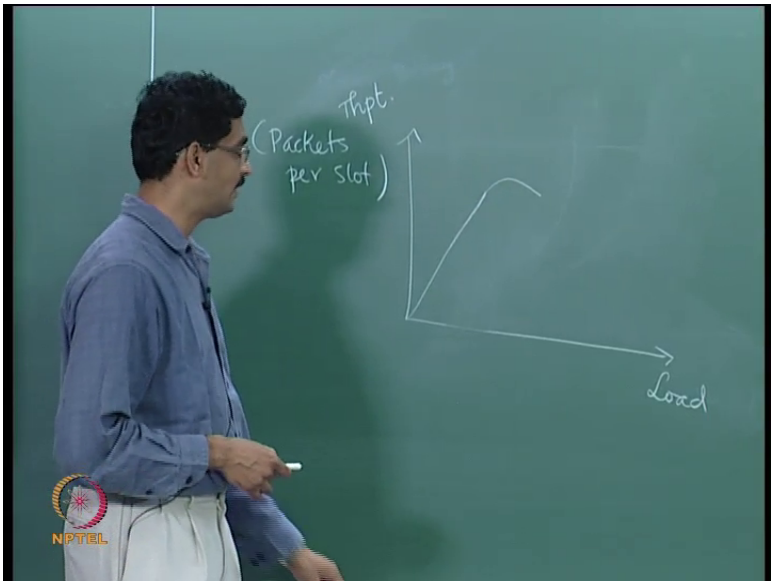
And just there is this one out layer I do not know what happen and otherwise you find out if you can actually replicate that that is sometimes hard with experimentation. you run the same sorting algorithm again. Say so, let me see if I get the same set of results, the same trend show of again; second time if you does not show then you know that there some transient behavior in the system, just when you are running this particular data set you are something start at downloading right a 10 o clock it got turned on started downloading, your CPU got busy therefore, your searching time or sorting time for that particular data set was back. So, you try to repeat the experiment first, and then see if you can same trend is noticeable, and if it does not then great. Otherwise you have to go back, and look at the code. There is something wrong in your code when a particular value of that load value is set.

(Refer Slide Time: 48:44)



So, there we should ignore or is should avoid the tendency is to simply do not plot the value at all; plot this, plot this, plot that, then you are done. But that it does happen just delete that value from your file and keep going on. So, you have to go back, and look at out layers. Some of it is (( )) the ethics class, but anyway what these are things that do occur, but you should have the intuition as to why your system behaves that way. Either explain your particular behavior or do not explain at all.

(Refer Slide Time: 49:20)

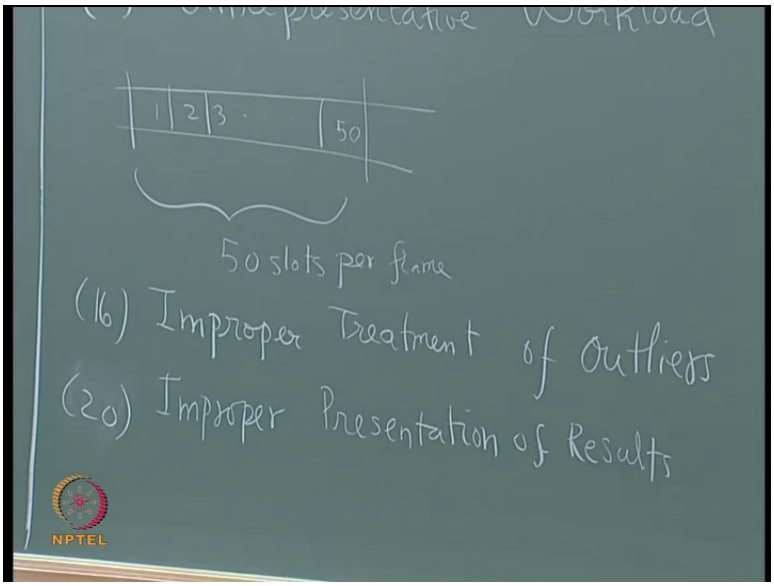


So, in another case where you would see this is if I look at again aloha or slotted aloha is the example and I vary that load versus. So, this is increased offered load, increasing offer load to the system. And here I am measuring the throughput, through put is in term of number of packets per second that are being generated. So, in aloha if I have only one packet maximum I can send one packet per slot. So, this is throughput in terms of number of packets per slot. So, throughput will be proportional to load if I offer let us say 0.5 packets per slot that is the total offered load then the number of packets sent per slot is also 0.5 per slot on. So, at after some point you will, till some point your load and the measure throughput should match.

So, you will see some sort of behavior like this it goes up like this it goes up like this it can of more or less linearly keeps increasing with the load offer to the system. And then you find that now your suddenly some out layer beginnings you find that it starts (( )) like this. That is expected behavior in aloha yes or no why that is when you are 36 percent limit comes. So, if you did not know that there is somebody else has told you about the 36 percent or if you do not think that you know what beyond some point collisions will happens so much, that you cannot really send any more packets. So, this is starting to show the beginning of some, some sort of out layers. So, you can is this just say that well I will stop my simulation at this point. There is something wired going on there may be my simulation is incorrect or my modal is incorrect. So, here is where you should have intuition about why the system is behaving in this particular way. And in a case of aloha it is very easy to figure out intuition, but in some cases it is not when you go to more complex systems lot of sub systems are involved.

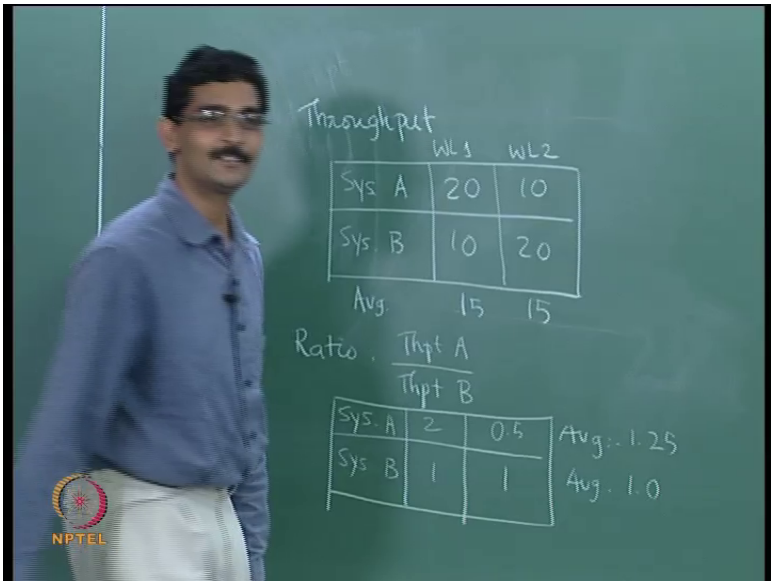
Then it is very hard to predict as to why right some of these are happen have some of these out layers are occurring. So, you do your best to explain and if not just state in your thesis report that you know there is this wired out layer I could not explain this and just leave it at that then just straight to hide that if it is repetitively occurring out layer. So, that is the improper treatment of out layers that is another common mistake that should be avoided.

(Refer Slide Time: 51:44)



Then one more example is improper presentation of results right or even analysis of results. It is not 17 actually according to the book number it is 20. So your first chapter that is this comparison of 2 different systems. So, I am let say measuring throughput of two different router implementations or the protocol implementation.

(Refer Slide Time: 52:31)



And, so, I am measuring throughput. There are two systems A and B and I have two workloads workload 1 workload 2. Throughput in some unit does not matter what the unit is. Now there are these two units 20, 10; 10, 20. So, in 1 work load system one gave

20 throughput; 20 units of throughput whereas, system B gave 10 units of throughput, and reversed in the other case. So, which of these two you are suppose to compare these two systems only these two values are there. So, which is better this is again what you will see some times in thesis, I have done two values I am done I am compare two different systems. So, which is better?

(( )) we get have two cases in this case we say the system A is better when the work load is WL2 when the work load is move of say then system A is better when the work load is (( )) system B is better.

So, this could be say two different workloads and therefore, so this could be high load, low load. So, in that case you might come out like in the case of t d m where we said that one is better. But let say that this is two different random seats when we generate traffic in a system there is some randomization in the generation. So, I could always we have to actually do multiple random seats initially the random seat differently each time I get you know. So, that the same lambda or number of you know request per second is arriving. But the way in which they arrive is there if some randomness in there therefore, this leads to these two different values. So, let us assume that same load; work load, but differences in the way the traffic is generated. In that case what happens can you conclusively say there one is better than the other if I simply I will say mean, I will compute the mean throughput. So, the average throughput is 15 you find that this is not satisfactory in you want finish of writing your report. So, I need to find a constructive way of showing that system A is yours and you want to show the system A is actually better than system B. So, how can we player on with this in terms to see if I can make system A better than system B.

Can we introduce some more points?

Now let say that you are running out of time due this you tonight, no time to take more point, either you cook a points or you have to use these two points.

Ignore at the workload.

The work load to the one is to simply show only workload. But your advisor is not happy say no no two workloads are needed one is I simply contact up to just one workload. That is when you start playing so called ratio games. So, let say I compute instead of

throughput I say Ratio of throughput of system A to system B that is my metric, now I am change my metric from simply throughput to ratio of throughput of these two systems. So, this is ratio defined as throughput A by throughput B I can do that. This actually course supposes to issue you good practices in performance evaluation. But this is not a; does not fall in that category. So, if I do that ratio, now what will happen? So, this 2, this is 1, actually this average should come here not under here. 2 1 and this is 2 1 and this is 0.5 and 1.

So, both systems system B is compare to system B therefore, the ratio is 1. System A is compares to therefore ratio is 2 1 0.5. Then I compute; I say my metric is this ratio I will compute the average improvement of system A with respect to system B. So, I compute the average here average of 2 1 055 is 1.25, average is 1. Therefore, the system A is better than system B. That is you know either you can call this analysis or presentation whatever it is. But this is the way you twist the result to show that this is a better way to do that. But this is not what one should do, but when you probably might see this in papers then you might say this is not correct, this is in not only papers even when you go to companies you have be asked to compare two different systems.

Then when you run out of time, your boss says the report is to yesterday, then you say fine I will show you this system is actually better. Because you want to compare your implementation to some other competitors implementation and then you have to somehow make the boss happy for that day that your implementation is actually better. So, we can; so, we can use also (( )) to come up with different metrics. So, that is an example of the simple offer analysis representation of results. So, the list goes on and you can look at that later on, that this is our conclusion of chapter 2.