Parallel Computer Architecture

Hemangee K. Kapoor
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati

Week - 01
Lecture - 05

Lec 5: Reporting Results

Hello everyone. We are doing the module 1 Introduction to Parallel Architectures. This is lecture number 5. The title is Reporting Results. In the previous lecture, we have looked at the definition of performance and the list of benchmarks or the requirement of benchmarking. In this lecture, we are going to study once we get the benchmark results, how do we go ahead and report or do the comparisons between the different machines.

Okay. So, the theme is how do I collate the results that I get after benchmarking. Alright. So performance of a set of programs is what we were looking at. So performance is measured separately for every program that is you run one program on a machine you get its execution time. So, T_i is the execution time for program i and we repeat this for all the programs in the benchmark suite.

So, suppose I have programs P_1, P_2, P_3 up to P_m in my benchmark and for every program we compute the execution time. So, T_1 is the execution time for program P_1, T_2 for P_2 and so on up to T_m. So, this is the value of execution time of every program. Okay. So once we get this, how do we report the performance because we have different execution time for different programs. So, how do I collate these results to give one final answer? So a straightforward solution is that we can take some mean to find this out. Okay.

So what type of mean values do we want to take up? So one popular one is arithmetic mean. So I will take an arithmetic mean to report a single value. So we had the execution times T_1, T_2 in the previous slide. So I have to do an arithmetic mean of all the T's. So I will do sigma of T_i where i goes from 1 to m because I had m processes in benchmark suite. Okay.

So all this and then I have to divide this by m because I have m processes. So what is going this going to be? This is going to be the arithmetic mean of execution times. If I say this is T_exe and I will also put another subscript arith. So that is the arithmetic mean. So that is how we can report a performance of a set of programs.

The problem with arithmetic mean is that if there is a program which runs longer, if one program takes 100 time units and another takes only 10, the overall arithmetic mean will be larger. So if you want to even it out, for example, program P1 executes only 10 instructions. It is a small program and program P2 is executing 1000 instructions. So naturally, this is going to take longer. So if you have more instructions, the program will take longer.

So definitely the time will be more. So here the execution time increases. So we have a longer execution time for P_2 and a smaller for P_1. So overall performance of this machine will show poor because P_2 will be taking much longer than P_1 on this particular machine because of the size of the program. So in this case, to offset such programs, we could associate a weight with the mean value.

So I can associate a weight with everything. So this weight could be a function of the instruction count. Right. So we can associate a function of the instruction count. So that I can equalize the effect of P_1 and P_2. So we will have effectively T_2 multiplied by the weight of P_1, then T_2 multiplied by the weight of P_2.

So overall the execution time of program i multiplied by the weight associated with that program. So now this is used to compute the arithmetic mean. So i going from 1 to m and then the same thing we have to do this. So what is this called? This is called the execution time arithmetic mean, but I will just put a wt that is the weighted arithmetic mean. So this is how we can compute the two means.

So the weighted arithmetic mean and the normal arithmetic mean formula are shown on this slide. All right. So the next measure of performance is speedup. But speedup when I say speedup, how fast is one machine compared to other? So for speedup we need a reference machine or a reference entity. We cannot simply say that I am x times fast, but compared to what? So that reference machine is very important when we have to report speedup. Okay.

So if I have a single program we normally take, this is the execution time of the program, it is this much on this machine and that A on this machine and B on that machine and then we can compare and say that this program runs faster on this machine compared to the second machine. But if I want to compare two programs on the same machine, so how do I say that which is faster right or if you have two enhancements, you did some improvement in the hardware and improvement one and improvement two and you want to compare which improvement is performing good. So every improvement is used on a particular benchmark program. So you run that program, then with the same improvement you run on a reference machine and then compare the execution time on your machine compared to a reference machine. Okay. So to do the speedup, we need a reference machine.

So I am using ref as the reference machine and m1 as my machine. So on this machine the execution time is T_1 and on the reference machine the execution time is T_ref. So these are the two execution times. We have seen on the previous slide that a set of execution times are average to give an overall average execution time. In terms of speedup, for every program we need to compute how fast the program runs on a given machine. Okay.

So I have these programs P1, P2, P3 and so on. So for every program we compute the execution time and then compare it with the reference machine. So I will say T_R that is execution on the reference machine divided by T_1 which is the execution time on this machine for the program P1. Similarly T_ref divided by T2 and so on. So we are going to do this.

So once we have this set, what is this fraction called? This fraction is called speedup. So I will use the term S for speedup and I will simply say 1. S1 is the speedup of program 1 with respect to reference machine. So this is S2 and so on up to Sm. So this is the respective speedup.

Again we have a set of values here. Like we had a set of values for execution time, here we have a set of values for speedup and when we have a set of values we need to take mean value. Okay. So we could use the same methods and use an arithmetic mean. So arithmetic mean is sigma of all the speedups. So I will say,

$$1/m\, \Sigma^{m}_{i=1}\ S_{i_i}$$

So if I have m programs, if I have n programs then this is 1 over m sigma 1 to n of this. So this is going to be the arithmetic mean of the speedup. So on the slide you can see that for a set of programs P1, P2, P3 up to Pm, we have computed the speedup of individual program S1, S2, Sm. For this set of speedups we need to average it out. So I have worked out an arithmetic mean but we can also do different other types of means.

So what are the options with us? We have arithmetic mean, we have harmonic mean and we have geometric mean. So this slide shows the different formulae for every mean. Arithmetic you have all seen. For the harmonic mean we simply have to take the reciprocal of the speedup. So 1 over Si, then take a sigma of this, i ranging from 1 to, I have used n here for n processes and then we put this in the denominator and then take the harmonic mean.

Okay. So this is the harmonic mean for the different speedups. The other mean is the geometric mean which is the nth root of the product of all these speedups. So we take Si, take the product of all i ranging from 1 to n and once this is done you raise it to 1 over n. So this becomes the geometric mean. So the speedup has got these 3 variations to take up the mean value. Okay.

We have seen 3 means. Now which mean to take? So there are some observations that arithmetic and harmonic mean, they have slight problems because if you change the reference machine with which you are comparing, the values tend to differ. Okay. So what I mean to say here that, if we have arithmetic mean of the speedup, so S_arithmetic for machine R1. So this is for my reference machine R1. Okay. So when the reference machine is R1, all the speedups will be with respect to R1. So it will be $T_{R1}/T_1, T_{R1}/T_2$ and so on for every program.

This is going to be my set of speedups and the arithmetic mean of this speedups is S_arithmetic on machine R1. So it will be sigma of the speedups on R1 for all the programs i ranging from 1 to n. Okay. So this one. If we change the reference machine, then what do we get? Speedups with respect to reference machine R2 for all the programs 1 over n. And now what is this value? This is arithmetic mean with respect to reference machine R2. Okay.

So the observation says that these two values, suppose I call this value V1 and this value V2, the observation is that the V1 is not always equal to V2. Okay. So not always. V1 and V2 are not always equal in arithmetic mean. The same holds for harmonic mean as well. Okay. But we do not have problem with geometric mean.

So if you recollect the geometric mean was the nth root of the products of the speedups. Okay. So if I write s geomean on reference machine R1, what would that be? Reference machine R1's speedup for program i,

$$(\pi^{n}_{i=1} S_{R1_i})^{1/N}$$

Okay. So the nth root. So this is value V1 and then we have a similar value V2 with respect to machine R2. Okay.

This is value V2. And the good property of geometric mean is that even if we change the reference machine, the overall mean remains the same. So V1 is always equal to V2 even if the reference machine is changed. One other property of the geometric mean is that it is composable. Suppose I have already run three programs P1, P2, P3, we have obtained the geometric mean for this set. So, I will say the geometric mean for set 1 and later you intend to run some more programs, but you do not want to rerun P1, P2, P3.

So you have another geometric mean value for this separate set. So another geometric mean value of 2. So once you have these two values, it is not possible in harmonic and arithmetic mean to combine it, but in a geometric mean formula we can easily combine. So the overall mean in geometric can be combined using these two values. So what is it? It is simply the product of these two means. Okay.

So the overall mean will be the product of these two geometric means. So which mean to take harmonic and arithmetic are good enough, but geometric mean is more stable. We can draw the same conclusions across different reference machines and it is also composable. Okay. Next we are going to do some examples to understand the concepts further. I will also ask you to solve the examples as you see this lecture. Okay.

So on this slide you can see a table. The first column is the list of machines. So we have machine 1, machine 2, ref 1 and ref 2 are reference machines. Then the second column is giving the execution time of program A. So program A takes 10 time units on machine 1, it takes 1 time unit on machine 2.

So that is the interpretation of the value. The third column is program B's values for every machine. Okay. So on machine 1 we have two values for program, one for program A and one for program B and we want to calculate one final value of performance on machine 1. So how do you do that? Either of the means which we have discussed, so we can compute an arithmetic mean of A and B on machine 1 and then fill the value. So I will ask you to fill it yourself also along with me. Okay. So here I am going to do the arithmetic mean of these two values on machine 1.

Then same thing here. For this, this is

$(100 + 10000)/2$, $(100 + 100)/2$,

So if you solve it, this comes to 55, this comes to 100.2, sorry 100.5 and this is 5050 and that is 550. Okay. So these are the arithmetic means of the performances on different machines.

And just for fun we can compare them and check how do they behave. So if you compare these two, which machine is better if you want to run both A and B on it with respect to arithmetic mean? And when we talk of execution time, you have to be faster, so the smaller value is good. Okay. So we want to pick this value 55 instead of 100. If I am comparing these two machines, then I am going to select 550 over 5000. Okay. So if I just write them in some order, the fastest machine to execute A and B is machine M1.

I will use the greater than sign to say that it is good. It is not that it is greater, but the greater than sign implies that M1 is better than the right hand side machine. So M1 is better than M2, then M2 is better than Ref2 and then last we have Ref1. Okay. So with respect to execution times, this is how we can draw the order. So machine 1 is better than machine 2, Ref machine 2 is better than Ref machine 1 and so on and execution time lower is better.

Alright, so we will move on to calculate the speedup of the same example on the left hand corner, I have copied the table from the previous slide. Now this orange table, what does it saying? It is going to tabulate the speedup. Speedup as we define is with reference to a reference machine, with respect to a reference machine. So the first two columns are with respect to reference machine 1 and the third and fourth row are reference machine 2. So what am I going to fill in this cell? So the first cell under program A column is going to report the speedup of program A on machine M1 with respect to reference machine 1.

Alright. And similarly program B, how fast it is compared to machine 1 with respect to reference machine 2. So we have to fill all these values. So let us calculate that in the next slide. So I will ask you to do it along with me. Okay. So that table is here for us and now we can use that table to calculate these values.

So let us do it for P_A. So program A, machine 1, reference machine R1. I am using a short form instead of REF1, I am using R1. So program A, machine 1, R1. So P_A with respect to machine 1 means, with respect to reference machine 1 that is execution time on reference machine 1 divided by execution time on machine M1.

We have to write this. So what is this? This is the speedup of program A. So the speedup of program A again on reference machine 1 and normal machine M2. So reference machine R1 and execution machine M2. So speedup of A on machine M2 with respect to R1. You calculate this value and the same thing you can do for the others.

So you can do reference sorry speedup of A with respect to reference machine R2 for M1 and then M2. Okay. So let us do the first one. T_{R1}/T_{M1}. So what was T_R1 for program A? So program A, this is the value 100, and the value takes on machine 1 is this 10. So performance of program A on machine 1 is 10 times better than that on reference machine 1.

Let us do it for machine 2. On machine 2, we have the numerator remains same 100 divided by, now the denominator is this value 1 here. Okay. So machine 1, it takes only one time unit, so the answer is 100. And if you do the others, here you get a 10 and here you get a 100. Alright. So what have we calculated? Performance of program A with respect to reference machine 1 and separately with respect to reference machine 2.

You can repeat the same thing. You can do speedup of B, reference machine R2 and M1, R2, M2 and same thing for reference machine 1 is left. Alright. So R1, M1, R2, M2. So you can calculate these respective speedups. Okay. So I have put those values here. I hope you could arrive at these similar results with the calculation. Okay.

Now let us see the last column. The last column is going to take ratio of the means. Here ratio of the means that is on the reference machine, how much time did it take? Reference machine program, overall if you see P program A and B, on reference machine 1, it took these two different timings, but the arithmetic mean is this much. So if I take the arithmetic mean on the reference machine, so, so I will take on the reference machine's mean value, the circled value and then the mean value on machine 1.

So 55, so divided by 55. So this divided by M1's mean. So mean on M1. So that is reference machine's mean divided by mean on that particular machine. So if you do this, you will get 91.8. So this is equal to. Okay. similarly if you want to fill this value, what is this value? It is machine 1 and Ref 2.

So Ref 2's average is 550 and machine 1's average is 55. Got it. So Ref 2's average divided by machine 1's average gives you this value. Okay. Moving on, we have the individual speedups. Now we want to average it out. Similarly we have three types of means, arithmetic mean, harmonic mean and geometric mean. I have already pre-populated the table because now these are very straightforward.

You have a speedup of 10 here, 100 here. So its arithmetic mean is this value. Right? So you take arithmetic mean of these two values, you get 55, harmonic mean 18 and geometric mean 31.6. Okay. So the average speedup can be calculated using three different mean values. Okay.

Now let us compare across these different means. So if I take this, let us do some comparisons using say arithmetic mean. So you concentrate on this column here. Okay. So we have reference machine 1, normal machine M1, reference machine 1, machine M2, we have four values. I am just listing them here on the side R2, M2. If you compare these four values or if I want to say with respect to reference machine 1, which machine is good? Is M1 good or M2 good compared to Ref 1? With respect to reference machine 1, is machine good, 1 good or machine 2 good? Okay. Compare these two values and remember this is not execution time, this is speedup and speedup should be higher.

Higher the speedup, it is better. So which value is higher? 75 is higher than 55. So I can say that between these two, on reference machine R1, M2 is better than M1. So I will draw this conclusion. The greater than sign implies that the better one is on the left hand side.

It is not greater always. This one says R1M2 is a better combination than R1 M1. If you repeat that for the second pair, what will you get? You will get R2M2 better than R2M1. So if we want to say it, what does it mean? That machine 2 is better than machine 1 when compared to reference machine 2. Alright. So overall arithmetic mean, using arithmetic mean, M2 turned out to be better than M1 given both the reference machines.

On both reference machines, we had the same conclusion. You can pause the video and try to do the exercise on harmonic mean. So I will do it on this slide. Harmonic mean, which one is better? You can directly start writing now that you know what are we comparing with. So I am comparing these two values.

So with respect to harmonic mean, reference machine R1, which machine is good? M2. So M2 is better than M1. Again harmonic mean, reference machine R2. Second reference machine, which is better between these two? So we have 10 and 9.5.

So M1 is better than M2. Okay. If you remember the previous slide, arithmetic mean had drawn the same conclusion that M2 was better than M1, whereas with respect to harmonic mean, we have got two different conclusions. One is saying M1 is better and here this is saying M2 is better. Okay. Now let us do the same exercise on geometric mean.

So to compare with geometric mean, geometric mean on R1, 70.7 and the other one is 31.6. So I am saying M2 is better than M1. Then geometric mean with respect to R2, which is better? Again M2 is better than M1. Okay. So with this, and we had also observed that geometric mean is more consistent. We are getting the consistent conclusion from this. Another nicer observation is, if you take ratio of the goodness, so I am I have compared the speedup, but if I take ratio of the speedup, what will happen? If you take ratios of the speedup, what does that mean? These are the two speedups.

70.7 divided by 31.6. If I take ratio of the speedups with respect to R1, right, with respect to R1, this is the ratio and I will do the same thing with respect to R2. What is the ratio? It is $22.4/10$. You could do it the other way also. I am doing it M2 by M1. Okay. This is M2 by M1. Right. So this ratio of speedups, when you do with geometric mean, if you will calculate, you are going to get almost same value for both the reference machines.

So this is also equal to 2.3, it comes something. So it is close to 2.24. So on this slide, we have summed up all the conclusions that arithmetic mean says that M2 is better than M1. Harmonic mean had different conclusions for the two different reference machines, whereas geometric mean had a more consistent result saying that M2 is better than M1. And also, the ratio of the speedups was also same across the two reference machines. So to summarize this particular lecture, we have used execution time to report the results, but when you have several execution times, you can average them to report one value, average is either arithmetic or harmonic or geometric means.

Then to report speedup, it is always in comparison with a reference machine. So you take a reference machine and then take a ratio of the execution time on the reference divided by execution time on your particular machine and then compare that average. So when I have several such speedups, you have to take average of that. Average is arithmetic mean, harmonic mean or geometric mean. And the conclusion we drew is harmonic mean and arithmetic are good enough, but geometric mean is a more stable metric. Okay. So with this, we finish this lecture. Thank you. .