

---

It's not up to you to assign CPUs. You just spawn off the threads you want run, and the operating system will schedule them on whatever CPUs are available. That is its job. Anything you do will reduce performance, by limiting the opportunities for your threads to run. – [Tim Roberts](#) Oct 24, 2022 at 0:57



---

Hi @TimRoberts, yeah I know that, but this is strictly what I want to implement in my code. I know this is possible but I am just not sure how to implement it correctly. – [Gie Grajo](#) Oct 24, 2022 at 1:03

---

That is not exactly true tho. OS wouldn't know how to distribute threads on units sharing some cache, depending on their affinities. So, sure, I share your belief that Gie should not try to bother with that, and let the OS balance the threads on CPU, since whatever they would do would probably reduce performance. But one they will really know what they do, you can, on the contrary, increase performance, by taking into account the cpu/cores topology, and trying to make it match the amount of memory exchange between threads. – [chrslg](#) Oct 24, 2022 at 1:05

---

Hi @chrslg, thank you for your confirmation. The reason I want to do it is (1) because it is a strict requirement in my project (2) I noticed in the runs that when  $t = p$  the performance is optimal but as  $t$  increases the runtime also increases. I want to compare them with the instance that what if these threads are equally distributed in each available CPU. Also, by the way, I will only be using  $t - 1$  CPU to give separate the process from the OS processes. – [Gie Grajo](#) Oct 24, 2022 at 1:20

---

You misunderstand. Using affinity does not mean "I own CPU 1". It means "I cannot be scheduled on anything OTHER than CPU 1." The operating system will keep sharing the CPUs among all ready-to-run threads (and typically there are hundreds), but yours will not be able to participate fully. With very few exceptions, managing affinity results in a decrease in performance. – [Tim Roberts](#) Oct 24, 2022 at 3:37

---

[Show 2 more comments](#)