

Forum:[Hack Forums](#)

Thread: [Opinions about Backend Benchmarks](#)

happygonutty - June 9, 2024, 8:38 am

In the past few years, bench marking a programming language to prove its superiority has always been a norm. A good example would be which is the fastest, C, C++, Go, Rust and etc. Then a lot of would bang each other out that this is superior, not this isn't, no this is crap, and etc.

Now, I want to ask you guys in this thread on what are your opinions in regards with bench marking? Is it necessary or not? And why? I want to know your opinions, and not something out of ChatGPT. ?

secret_code - June 9, 2024, 10:01 am

(Jun 9, 2024 08:38 AM)happygonutty Wrote: [\[CITE\]](#) In the past few years, bench marking a programming language to prove its superiority has always been a norm. A good example would be which is the fastest, C, C++, Go, Rust and etc. Then a lot of would bang each other out that this is superior, not this isn't, no this is crap, and etc.

Now, I want to ask you guys in this thread on what are your opinions in regards with bench marking? Is it necessary or not? And why? I want to know your opinions, and not something out of ChatGPT. ?

there is no such thing as "superior" language, every language has its pro and cons, you might not be able to write system specific code, micro-services or advanced malware using python, its a fact, just like you would not be able to write a website using assembly, see the point? there are no superior languages, there are only superior programmers, programmers using assembly and c++ i can assure you they know better what they're doing then they're respective python or JS buddies, some languages require a deeper understanding of operating systems, memory management, and all sort of low-level operations compared to others, just like someone writing full C2 servers, ransomware, APTs and other advanced stuff is more highly probable to be more efficient than someone writing websites....

webdev77 - June 9, 2024, 10:34 am

A few milliseconds difference really depends on what the use case is. High frequency trading? Your app is worthless or could even cause you to lose millions of dollars if it's that much slower. A desktop or mobile app? Probably doesn't matter at all.

For the majority of software, it makes little difference. That said though, the more abstractions you add, the less the devs actually have any idea what's happening. Some day in the future there will be a massive number of security issues caused by insecure code that devs had no idea about. Maybe that day is already here for some older C and C++ tools. Rust will be particularly bad because people think it's safe by design but it's not.

secret_code - June 9, 2024, 10:43 am

(Jun 9, 2024 10:34 AM)webdev77 Wrote: [\[CITE\]](#) A few milliseconds difference really depends on what the use case is. High frequency trading? Your app is worthless or could even cause you to lose millions of dollars if it's that much slower. A desktop or mobile app? Probably doesn't matter at all.

For the majority of software, it makes little difference. That said though, the more abstractions you

add, the less the devs actually have any idea what's happening. Some day in the future there will be a massive number of security issues caused by insecure code that devs had no idea about. Maybe that day is already here for some older C and C++ tools. Rust will be particularly bad because people think it's safe by design but it's not.

true, rust is too over-estimated, there are already breaches made to rust infrastructures but people insist that is the most secure one... rust also has its issues, one of the most probable one is dependency vulnerabilities :), or another one which gives shell access

SIM-5457022 - June 9, 2024, 10:51 am

(Jun 9, 2024 08:38 AM)happygonutty Wrote: [\[CITE\]](#) In the past few years, bench marking a programming language to prove its superiority has always been a norm. A good example would be which is the fastest, C, C++, Go, Rust and etc. Then a lot of would bang each other out that this is superior, not this isn't, no this is crap, and etc.

Now, I want to ask you guys in this thread on what are your opinions in regards with bench marking? Is it necessary or not? And why? I want to know your opinions, and not something out of ChatGPT. ?

Benchmarking is only needed when you need to use a language that is light weight. If we are talking about servers and a service that a server provides. Benchmarking is needed why?

programming code is like building a restaurant. When the customer walks in they need to know what to do and how to do to place an order. If you make this over complicated then it needs to be you will lose this customer they will go else where. Design is key for this. Programming languages and the programmer needs to know to use just the code needed when it's needed.

If you provide a menu of your restaurants foods and then also add other foods that other restaurants in the area has. You just added unnecessary data. This wastes resources. The process from placing an order to having your employees make the food and then delivering the food to the customer and collecting payments is a process. Programmers and software engineers are paid to figure out what is the most efficient way to do this.

The programming languages takes care of the actual real code that runs on a machine. The programming language is just syntax on how you want your program to run. It takes that data and then makes that program in the most efficient way possible. The problem lies with the languages on what you can and cannot do.

Some languages have you load in libraries that won't be used. In C and C++ it's a powerful and complex language.

The default settings will include the actual library into the program. You can make it static linking or dynamic linking.

Anyways, some programming languages allow you to import a whole library into your program or you can actually just import the code that you need just the functions you need to use. Most people if they learn these languages from a book and are newbies they won't know about this. Some people doing benchmarks won't know this. However, some programming languages even checks if you have the same function declared in to locations at your source code. It will just ignore then one and checks other common mistakes. This is to avoid having bugs and wasting resources.

C++ and object oriented programming language allows you to easily make larger projects and large codes to be reusable. I feel the more you know a language the more you can make it run more efficiently. However some languages are not as efficient and may lack such options.

C? C++ and languages over time find new and new problems and issues programmers have or later learn better ways that would make the program run quicker and use less resources then

before. Those type of methods are more complex and is why they didn't easily think of it before hand.

So, performance is only key if it's really needed. Server based programs you want to make it run only the code that is needed since you will serve a ton of people and you want to be able to get the most out of the server.

If the program runs on a machine you might want to do this over time but it's not going to be something that is going to make or break it. It's more focus on what it can do rather than how it can perform normally at later versions you would always want to improve performance over time.

happygonutty - June 9, 2024, 11:21 pm

After reading all of your replies to my thread. I have no further questions from you guys and all I can say is that each and everyone of has his/her own personal opinion in regards with bench marking and its importance.

In my own personal perspective, there is no such thing as superior programming language and bench marking should only be used during "critical" situations that needs to be resolved in the real world especially when gaining or losing profit (e.g. liabilities, assets and money) is at stake.