Short Essay on Performance Matters Talk

João Pedro Amaral Dias

December 2019

Emery Berger's "Performance Matters" it's a talk given in The Strange Loop 2019 conference where the main theme was performance, which is a subject that is gaining a lot of visibility. It is obvious that performance is an important asset for users, but for developers high performance in applications sometimes becomes difficult to deliver. This is mainly caused because the current approaches for evaluating and optimizing performance are not the most suited, and this is the main premise of this talk and this essay. In this talk are also mentioned by the speaker some tools that can be used to achieve better results in matters of performance.

In the old days performance was a subject mainly ignored by developers because it was easily solved by improving the hardware of the machine running the program. But nowadays, this is no longer the case because hardware improvements don't mean faster machines and sometimes it's not possible to obtain more processing power due to some problems like overheating. After the speaker introduced the historical view regarding how performance was handled, he proceeds by exemplifying how a normal developer handle performance fixes. This example is followed by some argumentation on why the current approaches are wrong and how should they be made. The way that this optimizations and analyzes should be made are supported by the demonstration of two tools, Stabilizer and Coz. The first being a tool that evaluate performance improvements and the second being a causal profiler that can predict the impact of optimizations.

There is no doubt that an application with a good performance is what users look for, and this is already a positive argument to prove the importance of achieve high performance. But also the way that we are handling performance topics in the present is an alert to why we should give more importance to this subject. Currently, when developers face performance issues almost all of them try to solve them by changing and rearranging their code. This may result in an increase of performance but the cause could be pure luck, due to the influence that the layout (where the code and data ends up in memory) has on performance. This influence can be both positive and negative, and a simple change in the code can disrupt the behavior of the program in matters of performance. This variance in behavior is very important and is a big point on why should we focus on different strategies to achieve performance.

It's difficult to identify negative consequences when talking about achieving a good performance, and that's why we should continue exploring arguments that are in favor of this essay's thesis. And both of the tools mentioned in the talk by the speaker are good examples on why we should improve our approach, because once applied to an application used as an example in this talk, an increase in performance, that is not due to layout influence, is observed. Stabilizer induces randomness in layout to prove that the increase in performance is real and not caused by layout variance, and Coz helps by showing to the developer where he should make the changes to improve performance, as this tool predicts the overall performance boost that will result in the program by changing each module of the application.

This talk is a great starting point to understand the changes in paradigms that needs to occur regarding high performance developments. The methodologies used now are not the most appropriate because the increase in processing power is stagnating and we should instead focus on using the resources we have. In the future we should also take a look to other tools that are out there, similar to the ones introduced in this talk, because they will help accelerating this change