Scale, The Missing Link from Academia to Industry

Tae Eun Kim

2021. 09. 10

Abstract

Scalability is one of the biggest challenges for the tools derived from the software engineering research field. Unfortunately, most tools fail to scale, therefore leaving a gap between Academy and Industry. However, Facebook managed to tap a bridge between these two grounds when applying a static analyzer introduced in the Academic field to their development process. At the core of this success lies their effort to make the tool scale. Discovering how Facebook enabled their tool to scale can lead to some applicable points for research in diverse software engineering tools.

Scalability is one of the biggest challenges for the tools that sprouted from the software engineering research field. Research literature often flaunts their success, usually presented in the form of improved performance in the known problem. However, it is a common practice to test the performance in a controlled experimental environment. Many times, sophisticated tools are validated with a so-said real-world, yet refined and limited benchmark. Sadly, it is rare for a tool presented by research literature to successfully reproduce its performance in real-life settings.

The tools that fail to scale, in other words, not applicable when given the real-life settings, tend to separate Academy and Industry into two distinct fields. Not being able to impact beyond academic society is a demotivating situation for a researcher. Even for students who are interested in research with big dreams of changing the world, it will be a disappointing aspect. The separation of Academia and Industry is quite an issue because it is the loss of chance that could lead to great progress on both sides.

Facebook, however, has managed to tap a bridge between Academia and Industry when they applied a static analyzer to their development cycle[1]. The effectiveness and the scalability are often in a trade-off relationship. Scaling up a sophisticated technique usually results in reduced effectiveness. What makes Facebook's recent work significant here is that their static analyzer comes from the front-lines of Academic studies, differing from simple, primitive analyzers, and yet still being effective.

The key to Facebook's success lies in three domains. First, they considered "when" to analyze. There was a shocking result when analysis results were handed out to developers after being collected overnight. Almost none of the alarms were handled. Thus, deploying the analysis results as close as possible to the related action time was crucial. Second, they focused on "what" to analyze. Not all bugs are the same and thus should be treated differently. They selectively chose the bugs that mattered the most. Third, they carefully chose "how" to analyze. The theory they were grounded on was the compositional semantics. By exploiting the nature of compositional semantics, they were able to scale.

The lessons learned from the case of Facebook can apply to research scenes of diverse software engineering tools. First, let the tool have the possibility of scaling from the first place. It is important to base the tool on a theoretical concept that can scale. Despite all the engineering efforts, if the nature of the fundamental theory can't scale, neither will the tool that has rooted in it. Additionally, keep in mind the concept of parallel processing. The remarkable advance in computer hardware has enabled us to do multi-programming. Exploit the provided hardware because speed is one of the reasons to achieve scalability. Second, carefully imagine how the tool would be used. Factors such as the step in the development process in which the tool will be used, or the behavior of the user group will matter when scaling the tool. For example, if the tool is the static

[1]Distefano, D., Fähndrich, M., Logozzo, F., & O'Hearn, P. W. (2019). Scaling static analyses at Facebook. Communications of the ACM, 62(8), 62-70.

analyzer and the users are the general developers, then the tool should provide only minimal and actionable alarms. In contrast, if the users are security expert teams, the tool should provide thorough analysis results. Thus, by enabling the control of the degree of information it emits, the tool will achieve at least some level of scalability. Lastly, for the students and the junior researchers, do not be discouraged. Despite the numerous studies failing to scale, research work in Academia can still scale to an Industry level. The fundamental theories and concepts do contribute to society. Our research can make an impact and help people. All we need is the thrive to scale.

In conclusion, scalability is what bridges Academia and Industry. Facebook did a successful job by building that bridge when they adopted a static analyzer to their development process. Many lessons learned from this case study can apply to other software engineering tools as well. Just remember, your tool can scale, with the right amount and the right direction of effort.