

Deconstructing Complex Distributed Platforms: A Report From the Trenches

François Taïani

Université de Rennes 1 / IRISA, Rennes, France

francois.taiani@irisa.fr

1 Wading in unfamiliar software in modern distributed systems

Distributed services play an increasing role in our daily lives and our economy. Unfortunately, as their importance grows, so does their complexity, and the difficulty to analyse, verify, and validate them. Popular distributed applications such as social networks (Facebook, Twitter) or collaborative websites (Wikipedia, reddit) typically involve millions of users, support external developers through publicly available APIs (involving REST and JSON), and run in the cloud (e.g. Netflix and Foursquare). None of these technological feats would be possible without the use of complex software stacks, usually constructed from reusable third party components, many of which are available in open source.

Integrating these third party components is however challenging, and often requires developers to modify or analyse the behaviour of software they know little about. This happens for instance when hardening a piece of software (e.g. an RPC engine, or a message-oriented middleware) to use it in a mission-critical environment. This can also happen when diagnosing non-functional issues such as performance.

2 How to analyse 6,466,652 Java method calls

Analysing distributed platforms for hardening or diagnosing is a hard and multifaceted problem. One key challenge arises from the fact that the software stacks at the heart of many distributed applications usually combine many pieces of legacy and third-party software developed by different teams, in different projects, hosted by different organisations. Because of their size and complexity, these stacks typically cannot be fully analysed with entirely automated approaches. In this talk, I provide an overview of our experience with two real life platforms (a C++ RPC middleware [2], and Java web-service grid platform [1]) using two different kinds of strategy (graph manipulation, and interactive visualisation). Our work shows the importance of *pragmatic approaches* to analyse the dynamic behaviour of real platforms, and the potential for strategies based on *incremental abstraction* to construct representative models of said platforms that are intuitively understandable by practitioners.

References

- [1] Shen Lin, François Taïani, Thomas C. Ormerod & Linden J. Ball (2010): *Towards anomaly comprehension: using structural compression to navigate profiling call-trees*. In: SOFTVIS '10, ACM, pp. 103–112. Available at <http://doi.acm.org/10.1145/1879211.1879228>.
- [2] François Taïani, Matti Hiltunen & Rick Schlichting (2005): *The Impact of Web Service Integration on Grid Performance*. In: HPDC-14, IEEE, pp. 14–23. Available at <http://doi.ieeecomputer.org/10.1109/HPDC.2005.1520929>.