Using CPU Stubs to Achieve Realtime Behaviour

A Specialisation of Dynamic Performance Stubs

Peter Trapp[1], Christian Facchi[1] and Markus Meyer[1]

Ingolstadt University of Applied Sciences

Extended Abstract

In realtime systems the runtime behavior of applications is a central issue. For such systems there exist a set of timing constraints which have to be fulfilled. A lot of systems have to be optimized to achieve these timing constraints.

Dynamic performance stubs have been introduced in [1]. They can be used for a "hidden bottleneck" detection. Knowing the level of optimization potential, a cost-benefit analysis can be done, too. This leads to more gain oriented performance optimizations. The idea beyond *dynamic performance stubs* is a combination of performance improvements [2,3] of already existing modules or functions and the stubbing mechanism from software testing [4,5]. The performance behavior of the component under study (CUS) will be determined and replaced by a software stub. This stub can be used to simulate different performance behaviors. The optimization expert can use this set up to analyze the performance of the complete system under test (SUT).

Based on *dynamic performance stubs* we will present *CPU Stubs*, which simulate the performance behavior regarding CPU usage, more in detail. A simple implementation of *CPU Stubs* as well as a validation of their proper working is given. As a real optimization example we have chosen the telecommunication software of the LTE (Long Term Evolution the successor of the UMTS) network element NodeB. This network element has some timing constraints, which have to be guaranteed and validated in a deterministic way. If some of these key performance indicators cannot be achieved, the complete system is worthless. The software is also very complex and in addition fully automated tests are available, which is required for our measurements, so it is an ideal candidate for the validation of *CPU Stubs*.

We show with several measurements that our implementation of *CPU Stubs* is proper working and that the runtime behavior of these stubs can be deterministically modified.

Based on the implementation of *CPU Stubs* we introduce a methodology for the optimization of CPU bound systems: First a component under study, which is suspected to be a bottleneck, has to be selected. Then it has to be substituted by a *CPU Stub*. The proper working of the stub has to be validated and supported by measurements. As next step some measurements with different performance parameters of the *CPU Stubs* have to be done. Based on the results,

it can be seen whether the component under study is a real bottleneck or if a hidden bottleneck exists. In the first case it can be determined, which factor of optimization is necessary. Based on this value the required optimization effort can be evaluated and such an over optimization can be avoided. In the second case a hidden bottleneck can be determined and optimized at first. These steps can be repeated until the optimization leads to the required result or no further optimization is feasible in a technical or business sense.

To validate our methodology we present a case study which is based on the software of the NodeB-Element of LTE at Nokia Siemens Networks. We have proven that our method can be used in a large scale industrial environment and have found significant improvement possibilities of the runtime behavior of an application. So some required real time properties can be met now.

To summarize, the paper provides *CPU Stubs* as part of the *dynamic performance stubs* framework. Therefore, the following items are presented in the final paper:

- A possible implementation of *CPU Stubs*.
- A methodology on "How to create *CPU Stubs*".
- A methodology on "How to use *CPU Stubs*".
- A case study done for LTE from Nokia Siemens Networks.

Using the approach of *dynamic performance stubs* CPU bound systems, such as often exist in realtime environments, can be optimized.

We present *CPU Stubs* which provide a framework for the simulation of the performance behavior of software modules and functions. They can be used to realize a gain oriented performance improvement. It is additionally possible to identify "hidden" bottlenecks and the most relevant optimization candidates. We show how *CPU Stubs* are defined in a real system and validate their proper working. Furthermore, we present a methodology for the optimization of the runtime behavior of CPU bound systems. This methodology has been validated in a case study at Nokia Siemens Networks.

References

1. Trapp P, Facchi C: Performance Improvement Using Dynamic Performance Stubs. Technical Report 14, Fachhochschule Ingolstadt, August 2007.
2. Fortier PJ, Michel HE: Computer Systems Performance Evaluation and Prediction, volume 1. Digital Press, Burlington, 2003.
3. Lilja DJ: Measuring Computer Performance: a practioner's guide. Cambridge University Press, New York, 2000.
4. Bertolino A, Marchetti E: Software Engineering: The Development Process - A Brief Essay on Software Testing, volume 1, chapter 7, pages 393–411. John Wiley & Sons, Inc., 3rd edition, 2005.
5. Sommerville I: Software Engineering. Addison-Wesley, 6th edition, 2001.