

Sound Dynamic Deadlock Prediction in Linear Time

Florian Rudaj

Hochschule Karlsruhe
Matrikel-Nummer: 65106
`ruf11020@h-ka.de`

1 Einleitung

Sei es in Betriebssystemen, Web-Servern oder Echtzeitsystemen – in fast jeder modernen Applikation ist die Nebenläufigkeit (engl. „Concurrency“) von Tasks von großer Bedeutung. Beispielsweise könnte ohne Nebenläufigkeit ein Betriebssystem nicht mehr als ein Programm gleichzeitig laufen lassen. Ein IoT-Gerät müsste für die Netzwerkkommunikation die Aufnahme von Sensordaten stoppen. Allein diese Beispiele zeigen die Relevanz von Nebenläufigkeit in heutigen Systemen.

Nichtsdestotrotz gibt es neben den Vorteilen der Nebenläufigkeit, wie z.B. dem Performancegewinn durch das Verteilen von Tasks auf mehreren Prozessorkernen, auch Probleme, die mit ihr einhergehen – sog. „Concurrency Bugs.“ Grundlage dieser Concurrency Bugs sind die nebenläufigen Zugriffe verschiedener Tasks auf dieselben Ressourcen. Diese Zugriffe können dazu führen, dass ein Programm, welches die Tasks nebenläufig ausführt, abstürzt oder sogar in einer Endlosschleife verweilt, ohne tatsächlich Arbeit zu verrichten.

Es gibt mehrere Typen von Concurrency Bugs. Der wohl bekannteste von allen ist der sog. „Deadlock“. Dieser kommt zustande, wenn zwei Threads bereits eine Ressource reserviert haben, die der jeweils andere zum gleichen Zeitpunkt ebenso reservieren will. Wenn ein Deadlock vorkommt stürzt das Programm in Folge ab. Die Forschung beschäftigt sich schon sehr lange damit, Deadlocks zuverlässig und effizient zu erkennen bzw. vorherzusagen. Dabei wurden statische und dynamische Lösungsansätze entwickelt. Statische Lösungsansätze versuchen einen Deadlock anhand des Quellcodes zu erkennen, wohingegen dynamische Ansätze die Ausführung des Programms analysieren. Probleme dieser Lösungsansätze waren jedoch häufig, dass sie entweder zu viele false positives (Deadlocks, die aber keine sind) angezeigt oder eine zu schlechte Laufzeit haben.

Die vorliegende Arbeit beschäftigt sich jedoch mit einem kürzlich erbrachten und bemerkenswerten Fortschritt in der Vorhersage von Deadlocks. Nachdem in den nachfolgenden Kapiteln zuerst auf Concurrency Bugs und Deadlocks im Besonderen eingegangen wird, werden Methoden vorgestellt, die es ermöglichen, in linearer Laufzeit sowie mit sehr hoher Präzision Deadlocks vorherzusagen.

hier noch Quelle zur Einleitung im Paper hinterlegen, da das mit den statischen und dynamischen Lösungsansätzen daraus kommt

- 2 Nebenläufige Programme**
- 3 Die Vorhersage von Deadlocks**
- 4 Dynamische Deadlock-Analyse**
- 5 Bestehende Ansätze**
 - 5.1 Deadlock-Muster**
- 6 Sync-preserving Deadlocks**
- 7 Fazit**

8 Beispiele

Dieses Kapitel enthält Beispiele wie Bilder, Tabellen und Fußnoten verwendet werden können.

8.1 Fußnote mit Link

The website *Google*¹ is a search engine.

8.2 Referenz auf Bild

Bilder können gut benutzt werden um das Forschungsmodell zu zeigen (vgl. Abb. 1).

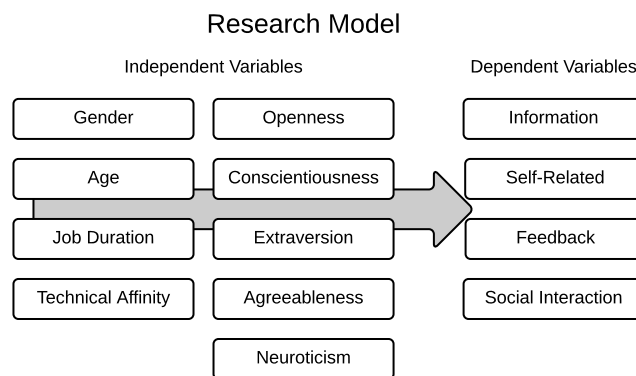


Abb. 1. Our case study investigating explanations for differences in usage motivation.

8.3 Beispiel Tabelle

8.4 Referenzen

The usage of social networking sites (SNS) for business purposes seems to be a promising approach for enhanced connectivity and communication among employees independent from space, time and position [1]. Since social media services like Facebook, Twitter and other SNS are part of our daily private lives [2], their implementation as a business support tool spread with amazing rapidity [3].

Die Referenzen finden sich in der Datei: references.bib.

¹ <https://www.google.com>

I use the software because,...	Scale	Loading
I can access information more easily.	Information	.825
I can access information that is relevant for me.	Information	.817
I will get informed about activities in my department.	Information	.775
I can present my ideas.	Information	.697

Tabelle 1. Dependent variables: Item texts and scales. Loading refers to the factor-loading of the principal component analysis after varimax rotation with Kaiser-Normalization.

Literatur

1. DiMicco, J., Millen, D.R., Geyer, W., Dugan, C., Brownholtz, B., Muller, M.: Motivations for social networking at work. In: Proceedings of the 2008 ACM conference on Computer supported cooperative work, ACM (2008) 711–720
2. Stocker, A., Müller, J.: Exploring factual and perceived use and benefits of a web 2.0-based knowledge management application: The siemens case references+. In: Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies, ACM (2013) 18
3. Koch, M., Richter, A.: Enterprise 2.0: Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen. Oldenbourg Verlag (2009)