

Prüfungsform: (Video-)Referat mit Ausarbeitung

Thema: Lamports Algorithmus für verteilten gegenseitigen Ausschluss

Ausgabe des Themas an die Studierenden am: **26.02.2021 9:30 Uhr** in MS Teams

Abgabe der Ergebnisse zum: **31.03.2021 23:59 Uhr** als Freigabe in der HAW Cloud

Abgabe in Ordner: <https://cloud.haw-hamburg.de/index.php/s/e8PsYIHFHB4Xjb4>

Ordner Passwort: **vs2021** Zip-Archiv-Name: **2394482.zip**

Abstract

In dieser Arbeit wird ein klassisches Thema der Informatik mit dem „Wechselseitigen Ausschluss“ aufgegriffen. Für dieses klassische Problem wird eine Grundlagenarbeit von Leslie Lamport eingeführt, welche die Grundlage der weiteren Betrachtung sein soll. Darüber hinaus wird ein Anwendungsbeispiel skizziert, an dem die Leistungsfähigkeit des Algorithmus demonstriert, in eine etablierte Architektur eingebracht und bewertet werden soll. Im Abschluss werden für die Bewertung verschiedene Aspekte eingebracht.

Einleitung

Der wechselseitige Ausschluss[1] ist ein allgemein bekanntes Problem im Kontext von kooperativen Prozessen mit einer geteilten Ressource. In verteilten Systemen ist dieses Problem in einer neuen Dimension zu diskutieren, da Mechanismen der Hardware oder die Synchronisation über gemeinsame Speicherbereiche fehlen. Ein etablierter Mechanismus zur Lösung dieses klassischen Problems ist der „*Lamports Algorithmus für verteilten gegenseitigen Ausschluss*“. Dieser Algorithmus wird in [1] und in [2] vorgestellt diskutiert und insbesondere durch Code Fragmente in [2] visualisiert. Wie jeder Algorithmus, ist der der vorgestellte gewissen Charaktereigenschaften unterworfen, die auf die Ziele der verteilten Systeme einen Einfluss haben.

Problemstellung

Der eingeführte Algorithmus soll an einem Beispiel diskutiert werden, welches ohne die Hinzunahme vorhandener Bibliotheken, Frameworks oder Code-Erweiterungen auskommt, wobei die Auswahl der Sprache freigestellt ist. Der Algorithmus ist von Grund auf umzusetzen und als Teil einer eigenen Bibliothek/ eigenen Frameworks anzubieten. Dieser Algorithmus soll austauschbarer Teil einer weiteren Bibliothek sein, die eine RMI/RPC Implementierung repräsentiert. Die RPC/RMI Lösung repräsentiert die grundsätzlichen Lösungsansätze, die durch die Methoden in [2] beschrieben werden und hält die Funktionen des Experimentablaufs transparent. Ziel der Hausarbeit soll die Leistungsbewertung des Algorithmus und die Einbettung in der eigenen RMI/RPC Lösung sein.

Experiment

Wesentlicher Teil des Experiments ist die RPC/RMI Lösung. Die RPC/RMI Lösung ermöglicht die Umsetzung einer Schweißrobotersystemsimulation (Siehe Abb. 1) mit max. 16 Robotern. Ein Schweißroboter in der Simulation hat ein Minimalset von Funktionen:

- `void setStatus(int status)`
Der Schweißroboter erhält Auskunft über seine Betriebsfähigkeit. Mit dem `int` kann ein Statuscode von einer noch nicht festgelegten Liste gesetzt werden. Bisher existieren 0 = OK, 1=WORKING und -1 = NOK.
- `void welding()`
Der Schweißroboter fährt an die vorher festgelegte Koordinate und setzt den Schweißpunkt. Alle Schweißroboter haben die gleiche Zielkoordinate, die nicht mehr bestimmt werden müssen.
- `void register(int id)`
Der Schweißroboter kann sich bei allen anderen Schweißrobotern bekannt machen.

Um die RPC/RMI Lösung zu evaluieren, soll ein Experimentaufbau realisiert werden, welcher die Leistungsfähigkeit der RPC/RMI Lösung demonstriert. An dem Aufbau des Systems darf **kein zentrales System¹** beteiligt sein, insbesondere

¹ Ausgenommen sind nur Systeme, die helfen geeignete Daten der Experimente zu erfassen

keine zentrale Koordinationsinstanz. Jeder Roboter ist als Node[2] zu interpretieren, sodass auch eine Kommunikation über alternative Netzwerke möglich wäre. Die geographische Skalierung ist auf einen Rechner **mit Kommunikation über Loopback** reduziert. Die administrative Skalierung beschränkt sich auf einen Administrator, der auch gleichzeitig der Nutzer des Systems ist. Ein Fremdzugriff kann ausgeschlossen werden, somit kann auch der **Sicherheitsaspekt unbeachtet bleiben**. Ein möglichst Hoher Grad bei der Erhaltung der Transparenzziele sollte angestrebt werden. **Fokus dieses Experiments liegt auf der gemeinsamen Nutzung von Ressourcen**, in diesem Fall 3 bis 16 Schweißrobotersysteme und eine geographische Koordinate. Dies legt damit auch die maximale Problemgröße in der Skalierung fest. Größere Systeme sind nicht zu berücksichtigen.

Der Ablauf selbst ist nur über die Ausführung einer Funktion „welding“ definiert. Alle registrierten Schweißroboter **einigen sich auf einen Zyklus**. Die Zykluslänge wird vorab für den einzelnen Experimentablauf festgelegt. Jeder Roboter ist bei der Berechnung der Zykluslänge akkurat. In jedem Zyklus wird nur von **genau drei Schweißrobotern** ein Schweißpunkt auf die gleiche Koordinate gesetzt. Alle übrigen Systeme nehmen keinen Schweißvorgang in dem Zyklus vor. Die Koordinate für die drei ausgewählten Roboter stellen eine räumlich geteilte Ressource dar, somit ist die Reihenfolge über Lamports Algorithmus für verteilten gegenseitigen Ausschluss zu realisieren. Es soll beim Ablauf davon ausgegangen werden, dass jederzeit ein Roboter ausfallen kann, aber ein Schweißvorgang immer abgeschlossen sein wird. In der Simulation kann davon ausgegangen werden, dass der Roboter von 100 Schweißaufträgen 99 erfolgreich beginnt und erfüllt. In jedem Fall ist aber das Setzen der drei Schweißpunkte von genau drei Schweißroboter auf der gleichen Koordinate im Zyklus sicherzustellen. Sollte es nicht möglich sein, die Aufgabe im Zyklus zu erbringen, sollte das System in einen Fehlerzustand gehen.

Nach Ablauf des Experiments hat jeder Schweißroboter mindestens 20 Schweißpunkte gesetzt oder ist in einem Fehlerzustand. In keinem Fall sollte während eines Experimentdurchlaufs ein Schweißroboter mehr als 3 Schweißpunkte als ein anderer Schweißroboter gesetzt haben. Die Ausführungsreihenfolge ist in einer Logdatei von jedem Roboter zu protokollieren. Die RPC/RMI Lösung hat dieses Verhalten sicherzustellen. Die RPC/RMI Lösung soll in fachlich korrekter Form hergeleitet und dokumentiert werden. Diese Dokumentation, wie auch die anschließende Evaluation werden in dem Referat erläutert und diskutiert. Die Dokumentation muss den Code abdecken und der Code die Dokumentation.

Evaluation

Um die RPC/RMI vom Leistungsvermögen bewerten zu können, ist die Leistung abhängig vom Erwartungswert und der beobachteten Leistung einzuordnen. Weiter soll festgestellt werden, wie die Leistungsfähigkeit abhängig der Problemgröße in der erbrachten Lösung skaliert. Für diesen Zweck ist eine geeignete Erfassung der Daten festzulegen und deren Informationsgehalt zu illustrieren. Grundsätzlich sollte der Experimentablauf mit unterschiedlichen Problemgrößen und Zykluszeiten wiederholt werden.

Zusammenfassung

Die Umsetzung von mindestens einem verteilten Algorithmus liegt im Fokus dieser Ausarbeitung, wie die Einbettung in einer RMI/RPC Architektur. Darüber hinaus sind weitere, dem Experimentablauf geschuldete Aufgaben in die RPC/RMI Lösung einzubeziehen. Am Ende steht eine Leistungsbewertung, die als minimale Aussage die minimale Zykluslänge Pro Schweißvorgang in einem praktischen Versuchsaufbau demonstriert.

[1] Leslie Lamport. "Time, Clocks, and the Ordering of Events in a Distributed System". In: *Commun. ACM* 21.7 (Juli 1978), S. 558–565. ISSN: 0001-0782.

[2] Andrew S. Tanenbaum und Maarten van Steen. *Distributed Systems*. 3rd edition. Maarten van Steen, 2017. ISBN: 978-90-815406-2-9. S. 314ff