



## **Diplomarbeit: Synchronisierungstechniken für replizierte Objekte**

Im Rahmen des EU-Projekts XtremOS wird hier in Ulm eine Bibliothek zur Unterstützung *Virtueller Knoten (Virtual Nodes, VN)* entwickelt. Ein Virtueller Knoten repräsentiert dabei eine Gruppe von Prozessen, die jeweils die gleiche Anwendung repliziert ausführen. Durch diesen Mechanismus wird die Anwendung fehlertolerant, da der Ausfall eines oder mehrerer Knoten nicht zum Ausfall der Anwendung führt. Eines der Hauptprobleme von Replikationsmechanismen ist die im Vergleich zur nicht-replizierten Ausführung entstehende Latenz - zumeist bedingt durch Konsistenzanforderungen. Um die Latenz zu verringern, wurden für VN sogenannte deterministische Scheduler entwickelt, die in gewissen Situationen Nebenläufigkeit erlauben. Diese Scheduler sind mit dem Fokus auf das in Java übliche *synchronized* Konstrukt entwickelt worden.

Seit Java 5 unterstützt die Java Spezifikation allerdings auch andere Arten von Synchronisierungsmechanismen, wie zum Beispiel explizite *lock* und *unlock* Aufrufe. Daneben werden Synchronisierungsmechanismen ohne Locks immer wichtiger.

Ziel der Arbeit ist es die in Java 5 neu hinzugekommen Synchronisierungsmechanismen auszuwerten und die Tauglichkeit der bisher implementierten Scheduler daraufhin zu untersuchen. Die existierenden Mängel sollen dabei aufgezeigt und Erweiterungen der Schedulingalgorithmen bzw. ein neuer Algorithmus vorgeschlagen werden. Im weiteren Verlauf der Arbeit sollen diese implementiert werden. Zusätzlich soll untersucht werden, ob wie speziell für Java entwickelte lock-frei Synchronisationsmechanismen in einer replizierten Umgebung eingesetzt werden können. Ein Evaluierung der neuen Strategien und ein Vergleich mit den bisher existierenden Algorithmen rundet die Arbeit ab.