Hessisches Kompetenzzentrum für Hochleistungsrechnen

Untersuchungen zu PGAS-Systemen: Programmierbarkeit, Synchronisation, Taskpools und Fehlertoleranz



Researchers
J. Breitbart, M. Bungart, J. Posner and C. Fohry

Project Term
2015 - 2015

Project Areas
Electrical Engineering and Information Technology

Clusters
Linux Cluster Kassel

Institute
Fachbereich Elektrotechnik und Informatik

University
University Kassel

Introduction

Die Verbreitung und zunehmende Größe von Parallelrechnern stellt die parallele Programmierung vor neue Herausforderungen, zu denen neben Geschwindigkeit und Skalierbarkeit auch die Steigerung der Produktivität der Programmierung sowie die Reaktion auf Hardwarefehler gehören. Innovative parallele Programmiersysteme setzen unter anderem auf das Konzept des Partitioned Global Address Space, bei dem alle Daten in einem gemeinsamen Adressraum untergebracht sind und der Programmierer dennoch zwischen schnellen lokalen und zeitaufwändigeren entfernten Zugriffen unterscheiden kann. In diesem Kontext wurden mehrere Arbeiten durchgeführt und folgende Ergebnisse erzielt:

Results

- Konzeption des Partitioned Single Assignment Memory (PSAM) Modells: Im Mittelpunkt steht ein Synchronisationskonzept, das auf der Weitergabe von Datenpaketen nach Abschluss ihrer Berechnung basiert. Das Modell wurde prototypisch implementiert und an Beispielprogrammen getestet.

- Bewertung und Vergleich der PGAS-Sprachen Chapel und X10 anhand von Erfahrungen bei der Implementierung von Taskpools und dem Unbalanced Tree Search (UTS) Benchmark: Im Vordergrund stand die Untersuchung von Taskparallelität und

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN · TECHNISCHE UNIVERSITÄT DARMSTADT · UNIKASSEL VERSITÄT · Philipps Universität Marburg

des Zusammenspiels zwischen Objektorientierung und Parallelität/ Verteilung. So wurden Varianten der Realisierung von Objekten, die intern verteilte Arrays enthalten, betrachtet und der Einsatz von Records mit dem von Objekten verglichen. Außerdem wurden Verbesserungsvorschläge für einzelne Sprachkonstrukte erarbeitet (z. B. Reduktion, knoten- und aktivitätsintern verwaltete Daten). Die entwickelten Programme wurden getestet und Laufzeitmessungen mit mehreren Knoten durchgeführt.

- Entwicklung eines fehlertoleranten Frameworks für globale Lastenbalancierung: In dieser Arbeit wurde das Global Load Balancing (GLB) Framework der Sprache X10 um Fehlertoleranz erweitert. GLB ergänzt die knotenintern verwalteten Aktivitäten der Sprache um global verschiebbare Tasks, die aufwändiger zu spezifizieren sind. Ausgangspunkt unserer Arbeit war die relativ neue Sprachausprägung Resilient X10, bei der im Falle eines permanenten Knotenausfalls eine Ausnahme ausgelöst wird, die vom Programm abgefangen werden kann. GLB verwendet für seinen Taskpool ein kooperatives Workstealing-Schema, bei dem arbeitslose Aktivitäten zunächst zufällige Stehlversuche unternehmen und später vordefinierte Partner kontaktieren, welche ein geeignetes Netzwerk bilden. Fehlertoleranz wird durch regelmäßig erstellte Sicherungskopien erreicht, die auch während eines Stehlvorgangs konsistent gehalten werden müssen. Dazu wurde ein konservatives Protokoll entwickelt und seine Korrektheit nachgewiesen. Tests und Laufzeitmessungen wurden mit UTS und dem Betweenness Centrality (BC) Benchmark durchgeführt.

Outlook

Für die Zukunft sind Arbeiten zur Erhöhung der Effizienz, Robustheit und Flexibilität des fehlertoleranten GLB-Frameworks geplant, außerdem zu RessourcenElastizität, also dem durch Elastic X10 unterstützten Hinzufügen von Knoten während der Laufzeit, sowie zur Implementierung eines GLB-ähnlichen Frameworks in Chapel.

Reference

[1] J. Breitbart (2012), Dataflow-like synchronization in a PGAS programming model, Proc. IEEE Int. Parallel and Distributed Processing Symposium Workshops (IPDPSW), Shanghai, China.

[2] J. Breitbart (2012), Programming models for next generation HPC systems, Euro-Par Workshop on UnConventional High Performance Computing (UCHPC), Rhodes Island, Greece, Eingeladener Vortrag.

[3] J. Breitbart (2013), A dataflow-like programming model for future hybrid clusters, Int. Journal of Networking and Computing 3 (1): 15-36.

[4] J. Breitbart (2013), Dissertation: Programming Models for Next Generation HPC Systems, Universität Kassel.

[5] C. Fohry und J. Breitbart (2014), Experiences with Implementing Task Pools in Chapel and X10, Proc. Int. Conf. on Parallel Processing and Applied Mathematics (PPAM 2013), Part II, Springer LNCS 8385, 75–85.

[6] C. Fohry und J. Breitbart (2014), User Experiences with a Chapel Implementation of UTS; Chapel Implementers and Users Workshop.

[7] J. Posner (2014), Bachelor thesis: Fehlertolerante Taskpools in der parallelen Programmiersprache X10, Universität Kassel.

[8] M. Bungart, C. Fohry und J. Posner (2015), Fault Tolerance Schemes for Global Load Balancing in X10 ; Proc. SYNASC2014 Workshop on HPC research services. https://doi.org/10.12694/scpe.v16i2.1088

Last Update: 2022-07-07 17:44