
Kurzfassung

Die Auswertung von Programmspuren ist eine bewährte Methode der Leistungsanalyse von parallelen Anwendungen und Rechnersystemen. Insbesondere eine grafische Analyse des Programmablaufs kann beim Verständnis von komplexen Zusammenhängen helfen. Um die Aussagekraft weiter zu erhöhen, werden neben den Programm-Ereignissen auch Leistungsmetriken in Programmspuren eingebettet. Es können unterschiedliche Metriken, von Cache-Hits bis zum Energieverbrauch, erfasst werden. Ein Teil der möglichen Messwerte fällt dabei auf externen Systemen an, z.B. SAN-Systemen. Programmspuren werden hingegen auf dem zu analysierenden System aufgezeichnet. Es bieten sich verschiedene Herangehensweisen an, um die externen Messwerte in Programmspuren zu integrieren. In dieser Diplomarbeit wird eine Architektur zur Erfassung, Übertragung und Aufzeichnung von externen Leistungsmetriken vorgestellt. Diese arbeitet mit den etablierten Leistungsanalyse-Werkzeugen VampirTrace und Vampir zusammen. Weiterhin wird anhand einer Fallstudie gezeigt, dass die Einbindung der Daten eines Energiemessgeräts Ansätze zur Optimierung des Energieverbrauchs einer Anwendung liefert. In der Fallstudie werden zudem verschiedene Implementierungsmöglichkeiten verglichen.

Abstract

Using program traces is a well-established practice for performance analysis of parallel applications and computer systems. Especially the graphical display of a program timeline helps to understand the cause and effect of a performance problem. Performance metrics are used in combination with the program events to further enhance the expressiveness of a trace. Datasources range from simple cache-hit counters on the CPU to external power measuring devices. However, some types of counters are only available on external systems such as a SAN, while the event traces are generated on the analyzed computing system. There are different approaches to integrate those external metrics into program traces. This diploma thesis presents an architecture for collecting, transferring and recording external performance metrics. The introduced architecture interacts with the existing performance analysis tool VampirTrace to leverage the well-established visual analysis with Vampir. Furthermore, a case study shows that the display of energy consumption measurements in the context of a program trace reveals the impact of MPI-Communication on the system's power consumption. The case study also evaluates and compares the different possible solutions that have been implemented.