

PIOsimHD

Julian M. Kunkel

Institute for Computer Science
Parallel and Distributed Systems
Ruprecht-Karls-Universität Heidelberg

18.11.2008

Ziele

- MPI & MPI-IO Simulierbar
- Einsatz/Nutzbarmachung von Standard-Tools zur Analyse der Ausgabe
- Validierung ist wichtig (Glaubwürdigkeit des Simulators)
- Skalierbarkeit (bedingt realisierbar, sequentieller Simulator)
- Neue Algorithmen/Verhalten schnell und reproduzierbar testen
- (Varianten der einzelnen Funktionsimplementierungen auswählbar)
- Was wäre wenn der Supercomputer (oder das Cluster) diese und jene Hardware hätte?
- Wie ist die Last-Balancierung? Welchen Sequenziellen Anteil hat die Anwendung?

Komponenten

- GUI um die Komponenten zu spezifizieren
- Simulator-Kern (Java), nimmt Komponenten-Spezifikation und mapped eine oder mehrere Applikation-Spezifikationen hinzu.
- PMPI-Wrapper erzeugt Eingabedatei/Trace von realen Programmläufen für den Simulator
- Benchmark-Tool erzeugt Konfiguration für Komponenten Spezifikation (erste Überlegungen)
- Tau-Trace-Generator - Konvertierung für Jumpshot möglich, profiling mit paraprof möglich
- Integration in Scalasca, (Cube3) wäre möglich um Expertensystem zur Analyse zu nutzen (Late-Sender)

Status

- Simulator-Kern simuliert Client Operationen mit wählbaren Varianten:
 - Rechnen(Zyklen)
 - Send/Receive (Tags, Wildcards) mit Eager + Rendezvous Protokoll
 - Barrier (Virtual)
 - Allreduce (Virtual, One-Gather-Scatter, Binary-tree)
 - File-Write/Read (PVFS-like)
 - Weitere schnell implementierbar (Werden in Schritte zerlegt, vgl. simpler Endlicher-Automat)...
- Switchtopologie möglich (im Moment nur eine Route).
- Server simuliert I/O-Read und Write
 - Mit simplem Write-Behind, verwaltet Cachespeicher und gibt diesen sofort wieder frei.
 - (Hier gibts noch viel zu tun)

Status (2)

- GUI erlaubt Komponenten zu platzieren, Eigenschaften auszuwählen (Praktikum)
- Simulator erstellt für Client-Operationen TAU-Trace
- Experimenteller PMPI Wrapper erstellt für wichtigste MPI-Funktionen XML-Trace Datei
- Benchmark bestimmt Latency für Netzwerk und ermittelt Zeiten für Allreduce, Barrier und Max-Größe der Nachricht für Eager-Protokoll (Validierung)

- Modellierte Komponenten erklären, XML zeigen
- Modellierung von Programm zeigen.
- PMPI Wrapper (kurz)
- Details zur Simulation: Wie werden Komponenten simuliert.
- Wie werden Netzwerke trotz Latenzvariation ausgenutzt? Wie erhält man dennoch eine halbwegs “realistische“ Simulation?

- Jumpshot Demo :-)