

2024 - BA Stöhr Meetings

Topic

Application and experiment automatic performance evaluation pipeline:

Dateisysteme und der damit verbundene Lese- und Schreib-Durchsatz werden üblicherweise mit Microbenchmarks und Applikationen getestet. Dies findet allerdings immer manuell oder über dafür konstruierte Skripte statt. Die Aufgabe der Bachelorarbeit ist es eine Test-Pipeline zu entwerfen und zu entwickeln, die es erlaubt verschiedene Dateisysteme und Anwendungen zu testen. Diese Pipeline soll modular aufgebaut sein, so dass (definierte) Dateisysteme und Applikationen per Knopfdruck gebaut und getestet werden können. Ziel ist es also eine Pipeline per Knopfdruck vollautomatisch auf einem neuen System testen zu können. Die Testergebnisse pro Anwendung und Dateisystem sollen entsprechend aufgearbeitet werden und z.B. per Abbildungen (Bar-Plot o.ä.) präsentiert werden.

TODO

- An [Paper](#) orientieren bzgl. Applikationen + [IO500](#)
- Support über mehrere Rechner muss beachtet werden
- Dateisysteme: Kernel, User-Space
 - Kernel: Hier muss nur der Pfad verwendet werden. z.B. /lustre ist ein Lustre-Dateisystem /localscratch ist XFS usw.
 - User-Space: GekkoFS, BurstFS, UnifyFS
- Finales Ergebnis: Auf Knopfdruck Benchmarks und Apps laufen lassen
 - Visualisierung sollen erstellt werden (z.B. Matplotlib)
 - Vergleich:
 - Für ein Dateisystem Performance Visualisierung über mehrere Apps und Benchmarks (Wall-clock time)
 - Pro App und Benchmark Vergleich über mehrere Dateisysteme.
 - Modulares System: Es soll einfach sein Dateisysteme und Use cases (Apps, Benchmarks) hinzuzufügen.
 - Konfiguration und Installation
 - Ausführung und eventuelle Laufzeit-Optionen
 - Ausreichende Dokumentation

Nice to have:

- Kernel Dateisystem installieren (benötigt root für pfad): EXT4, XFS, BeeGFS...
- Weiteren Benchmark: Elbencho <https://github.com/breuner/elbencho>

Meetings

Marc todo:

- git repo anlegen
- seafile paper material. Wie werden Dateisystem Ergebnisse präsentiert

15.5.24

- Motivation besprochen
- Topic-Beschreibung. Beispielhaft gezeigt wie IO500 funktioniert und wie Applikationsausgaben aussehen können
- Fokus der Bachelorarbeit soll auf der Architektur (und der Modularisierung) als auch Visualisierung liegen. Wie können Applikationsergebnisse am Besten präsentiert werden? Insgesamt macht es Sinn sich daran zu orientieren wie Dateisystem Ergebnisse in der Literatur präsentiert werden.
- Nächste Schritte:
 - Einarbeitung
 - Entwicklung kann lokal erfolgen. Zum Beispiel IO500 auf dem Arbeitsspeicher `/tmp` ist möglich.
- Abgabe: September anpeilen
- Wöchentliches Meeting
- Kommunikation: mattermost.gitlab.rlp.net oder Mail

Revision #3

Created 15 May 2024 13:14:56 by Lefthy

Updated 15 May 2024 14:57:56 by Lefthy