UNDERSTANDING HPC APPLICATION I/O BEHAVIOUR USING SYSTEM LEVEL STATISTICS

SEMINAR SUPERCOMPUTER

ARBEITSBEREICH WISSENSCHAFTLICHES RECHNEN
FACHBEREICH INFORMATIK
FAKULTÄT FÜR MATHEMATIK, INFORMATIK UND NATURWISSENSCHAFTEN
UNIVERSITÄT HAMBURG

VORTRAG: LEAH KNAACK

07.12.2021

ÜBERSICHT VORTRAG

- Einführung
- Hintergrund/Aufbau
- Datenerhebung
- Analyse
- Zusammenfassung

EINFÜHRUNG: ALLGEMEINE MOTIVATION

- Was tun Supercomputer?
  -> Berechnung umfangreicher Rechenaufgaben
- Viel Hauptspeicher benötigt
- Eingabe/Ausgabe Daten (I/O) evtl. notwendig
- Dateisysteme für Dateispeicherung und I/O Operationen
- Zugriff Daten auf Festplatten

EINFÜHRUNG: ALLGEMEINE MOTIVATION

- Hochleistungsrechner entwickeln sich immer weiter
- Verbesserung Rechenleistung schneller, als Festplattenzugriffsgeschwindigkeit
- Resultat: I/O Operationen schränken Effizienz ein (Flaschenhals)
- Lösung: Nutzung und Verbesserung paralleler verteilter Dateisysteme [Lustre]

  -> da besonders für (hohen) I/O-Bedarf geeignet
- Verständnis I/O-Verhalten aktueller HLR

EINFÜHRUNG: UNTERSUCHTES PROBLEM

- Frühere Studien Fokus auf Statistiken aus Anwendungssicht
  - Ergebnisse spezifisch für bestimmte Anwendungen
- Abgrenzung andere Studien: Analyse Statistiken direkt aus Dateisystem
  - unabhängig Nutzeranwendungen -> allgemein gültig
- Ziel: Weiterentwicklung paralleler Dateisysteme

EINFÜHRUNG:
RELEVANZ DER STUDIE (& DER MÖGLICHEN ERGEBNISSE)

- Allgemeines Verständnis von I/O Verhalten
- Auswirkung Workloads auf Leistung Dateisystem
- Flaschenhals vermindern, Effizienz vergrößern
  - Verständnis im Vordergrund, um allgemeine HLR Arbeit zu verbessern

EINFÜHRUNG:
WIE IST DIE STUDIE AUFGEBAUT

- Ganz allgemein: Statistiken sammeln und auswerten
- Fokus auf Lustre Dateisystem -> interne Statistiken
- 2 Cluster (Quartz, Cab)
- Datenarten gesammelt:
  -> Aggregate Job Statistics
  -> Time-Series job Statistics
- Kein Wissen über Arten der Anwendungen, die gelaufen sind!

HINTERGRUND/AUFBAU: LUSTRE DISTRIBUTED FILE SYSTEM

- Lustre = Linux und Cluster
- Paralleles, verteiltes Dateisystem
- Vorteil gegenüber anderer verteilter Dateisysteme
- Hochperformant für kleine und große verteilte Cluster -> Skalierbarkeit

A. Lustre Distributed File System

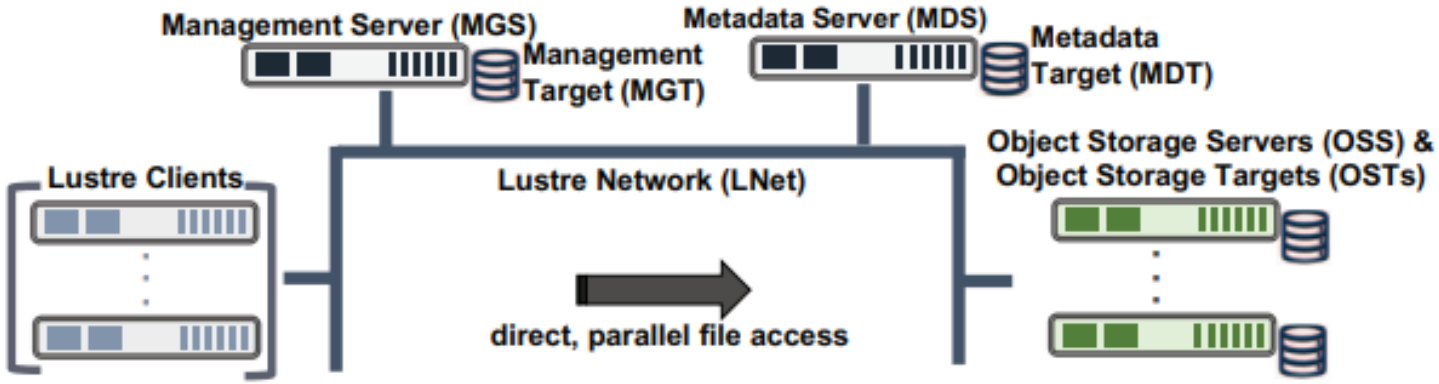


Fig. 1: An overview of Lustre architecture.

Quelle: Literaturverzeichnis 1)

HINTERGRUND/AUFBAU: CLUSTER

- Genutzte Cluster: Cab & Quartz
- Datenerhebung in Lawrence Livermore National Labratory (LLNL)
- Dauer Datenerhebung:

  -> Cabs: 3 Jahre Aggregate Job Statistics

  -> Quartz: 1 Jahr Aggregate Job Statistics, 3 Jahre Time-Series Job Statistics

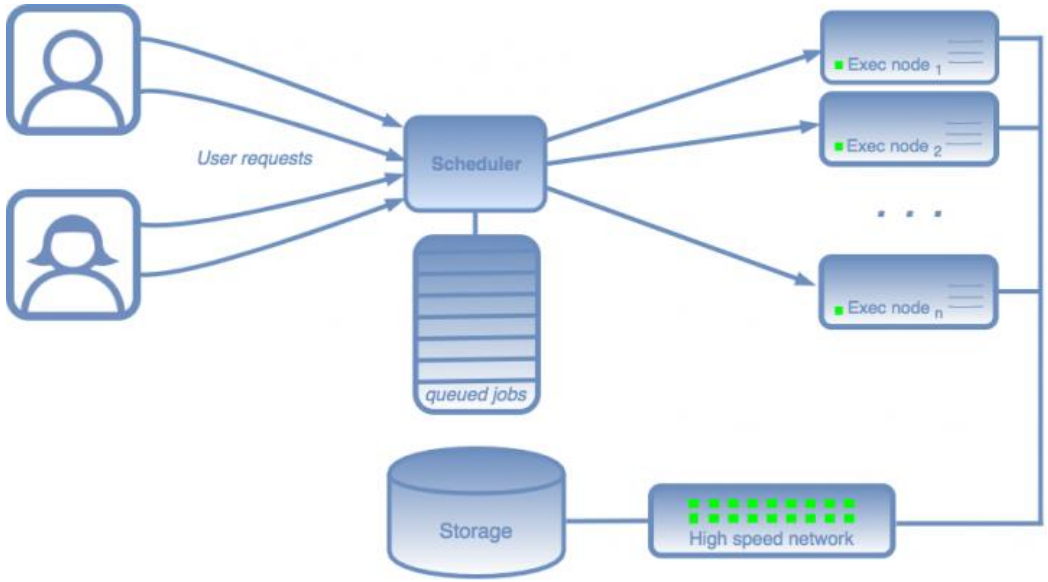
	Cab	Quartz
Processor Architecture	Xeon 8-core E5-2670	Xeon 18-core E5-2695
Operating System	TOSS 2	TOSS 3
Processor Clock Rate	2.6 GHz	2.1 GHz
Nodes	1,296	2,634
Cores per node	16	36
Total Cores	20,736	96,768
Memory per node	32 GB	128 GB
Total Memory	41.5 TB	344.06 TB
Interconnect	QDR Infiniband	Intel Omni-Path 100 Gb/s
Tflops	426.0	3,251.4

TABLE I: Cluster Configurations.

Quelle: Literaturverzeichnis 1)

HINTERGRUND/AUFBAU: SLURM JOB SCHEDULER

- Job Scheduler = managet Jobanfragen, teilt Ressourcen ein
- SLURM Job Scheduler
  - Hauptfunktionen: 1. Zugang zu Knoten zuteilen
    2. Framework zum Starten, Ausführen, Überwachen
    3. Warteschlange regeln
  - Open-Source
  - Hohe Skalierbarkeit
  - Linux Cluster



Quelle: Literaturverzeichnis 3)

DATENERHEBUNG: AGGREGATE JOB STATISTICS

- Lustre Client Knoten
- Pro Lustre Dateisystem
- Counter bei Lustre start (Job-übergreifend)
- Linux VFS Schnittstelle zu Lustre Dateisystem
- Anwendung greift auf VFS zu, VFS greift auf Lustre Client zu
  - Counter Systemaufrufe (lese-/ schreib-)
- Counter spiegelt Übergabe Anfrage von VFS zu Lustre wieder

DATENERHEBUNG: AGGREGATE JOB STATISTICS

- Lustre procfile
  - Gesamtzahl geschriebener/gelesener Bytes
  - Startzeit/Endzeit, Dauer, ID, Knoten
  - Metadaten
- SLURM Prolog und Epilog Script
- Differenz Lustre procfile Daten bevor und nach Jobablauf

DATENERHEBUNG: TIME-SERIES JOB STATISTICS

- Lustre JobStats feature
- Lustre Server Knoten
- Time-series = wiederholte Messungen über Zeit (1 sample alle 60 sek.)
- Aufzeichnung erhaltender Anfragen per Job-ID, pro Server
- Anzahl Bytes, Minimum/Maximum Bytezahl, Summe Bytes
- Insgesamtes I/O = Gesamtsumme der Werte gemeldet von (allen) Servern

DATENERHEBUNG: ZU BEANTWORTENDE FRAGEN („VORBLICK“)

- Typische Merkmale I/O-umfangreicher Rechenaufgaben (Jobs) ?
- Beziehung zwischen I/O-Anfragen und Metadaten(Server) ?
- Zeitliches Verhalten von I/O Verkehr ?
- Zusammenhang Anzahl geschriebener Bytes und Arbeitsspeicher ?
- I/O Contention in Bezug auf Zeit ?

ANALYSE:
A. LAUFZEIT JOBS UND ANZAHL GENUTZTER KNOTEN

- Verteilungsfunktion genutzt für Cab & Quartz
- Ergebnis:
  - 90% Jobs laufen kürzer als zwei Stunden
  - 90% Jobs nutzen weniger, als 100 Knoten
- Erkenntnis:
  - Großteil Jobs kurze Laufzeit -> Ansatz Optimierung

ANALYSE:
B. VERTEILUNG LESE- UND SCHREIB-INTENSIVE JOBS

- Jobs nach Nutzer gruppiert
- Lese- & Schreib-Anteil der Jobs analysiert
- Ergebnis:
  - Manche Nutzer Großteil Schreib- bzw. Leseanfragen
  - Ungefähre Gleichverteilung
- Erkenntnis:
  - Zuvor eher Schreibintensive Jobs -> Fokus Optimierung
  - Machine Learning führt zu häufigeren Leseanfragen

ANALYSE:
C. IN- BZW. EFFIZIENTE SCHREIBZUGRIFFE

- Anwendung kann mehrere Schreibzugriffe beinhalten
- Ineffizient = + Datenmenge - Bytes pro Schreibaufruf
  Effizient = + Datenmenge + Bytes pro Schreibaufruf
- Ergebnis:
  - Cab: 64,6% Jobs und 46,9% User ineffiziente Schreibaufrufe
  - Quartz: 69% Jobs und 66% User ineffiziente Schreibaufrufe
- Erkenntnis:
  - kaum effiziente Schreibanfragen
  - Training Nutzer

ANALYSE:
D. BEZIEHUNG ZWISCHEN METADATEN & I/O

- Metadaten-Operationen
  - Öffnen/schließen Dateien
  - Umbenennen
  - Datei erstellen
- Aufsummieren aller Metadaten-Operationen pro Job
  - Vergleich mit geschriebenen Bytes

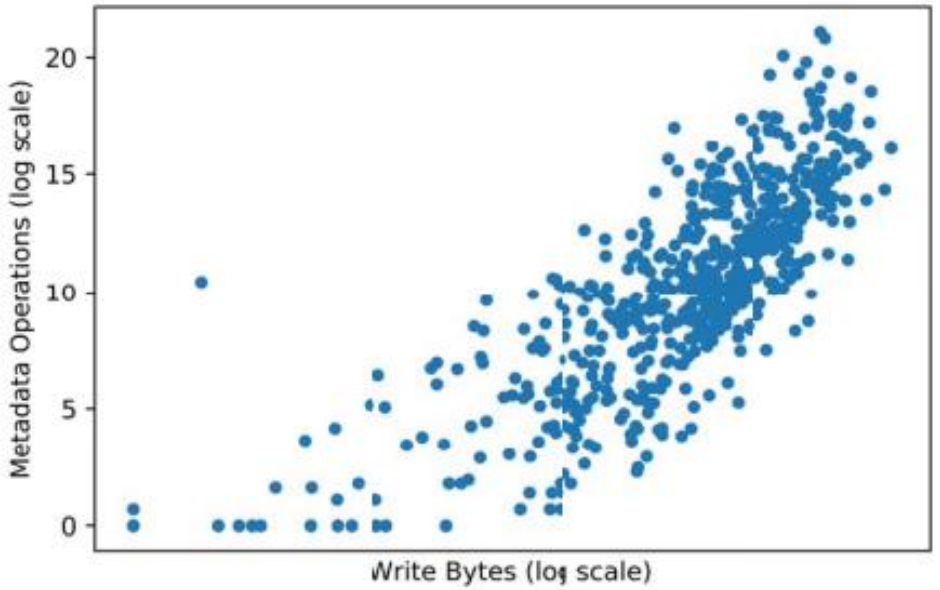


Fig. 4: #Metadata operations vs #Write operations.

Quelle: Literaturverzeichnis 1)

ANALYSE:
D. BEZIEHUNG ZWISCHEN METADATEN & I/O

- Ergebnis:
  - Je mehr geschriebene Bytes, desto mehr Metadaten-Operationen
  - Korrelation vor allem bei Datei-Erstellung
- Erkenntnis:
  - Metadaten großen Einfluss auf I/O Performanz
  - Fokus auf Verarbeitung Metadaten legen

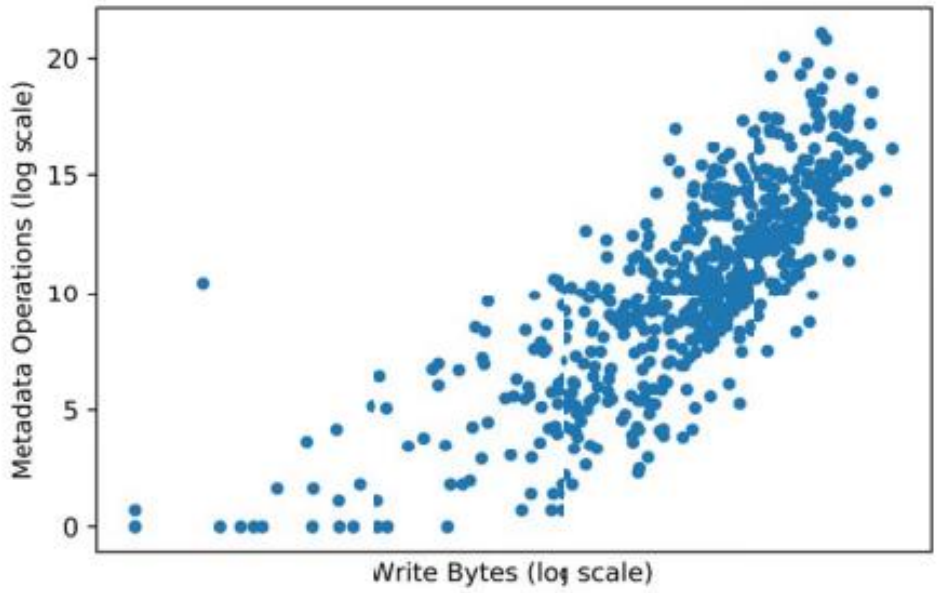


Fig. 4: #Metadata operations vs #Write operations.

Quelle: Literaturverzeichnis 1)

ANALYSE:
E. VERHALTEN VON METADATEN SERVERN

- Lustre besitzt 14 MDTs (da Zuwachs in Metadaten)
- Analyse Anzahl Dateiöffnungen und Schließungen verschiedener MDTs
- Ergebnis:
  - Anzahl Dateiöffnungen konsistent mit Anzahl Jobs
  - Nicht alle geöffnete Dateien wieder geschlossen
  - Anzahl Dateiöffnungen verschiedener MDT unterschiedlich
- Erkenntnis:
  - Ungeschlossene Dateien = schlechtere Leistung
  - Ausnutzung aller MDTs für bessere Leistung

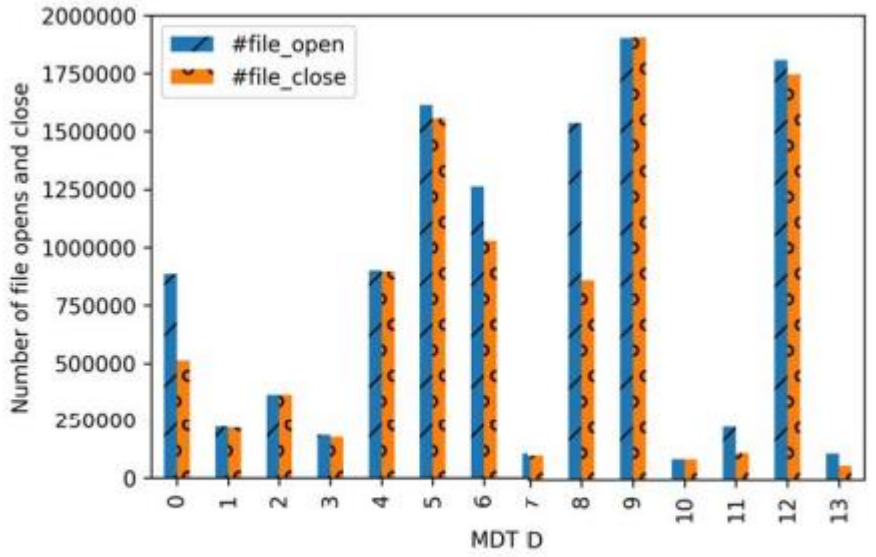


Fig. 5: #File opens and close handled by different MDTs

Quelle: Literaturverzeichnis 1)

ANALYSE:
F. ZEITLICHE ANALYSE I/O VERKEHR

A) I/O Verkehr besonders hoch an bestimmten Tagen?

- Heatmaps für I/O Verkehr
- Ergebnis:
  - kein bestimmter Trend

B) Einfluss Langzeit-Jobs auf Gesamtverkehr I/O?

- Berechnung Anteil Langzeit-Jobs an Gesamtverkehr I/O
- Ergebnis:
  - kein signifikant größerer Anteil

ANALYSE:
F. ZEITLICHE ANALYSE I/O VERKEHR

C) Einfluss Nutzer mit höchster Bytezahl auf Gesamtverkehr I/O?

- Berechnung Anteil Nutzer mit höchster Bytezahl an Gesamt-Verkehr I/O an diesem Tag
- Ergebnis: sehr signifikanter Einfluss

Erkenntnis:

- Wochentage keinen Einfluss auf I/O Verkehr
- Länge des Jobs nicht relevant, dafür aber Menge an Bytes

ANALYSE:
G. ZEITLICHE VERTEILUNG I/O INNERHALB JOB-AUSFÜHRUNGEN

- I/O Share = Prozentanteil an Laufzeit Job in dem I/O-Anfragen ausgeführt werden
  - Minutenweise, sobald 1Byte geschrieben/gelesen
- Ergebnis:
  - Großteil von I/O Share sind Schreibanfragen (80,4%)
  - Mittelwert I/O Share = 78,8%
- Erkenntnis:
  - I/O Aktivität über Gesamtlaufzeit Job verteilt (kein bestimmter Zeitpunkt)
  - Optimierungsansätze sollten sich auf Gesamtlaufzeit beziehen

ANALYSE:
H. BEZIEHUNG ZWISCHEN SCHREIB-BYTES UND ARBEITSSPEICHER

- Schreib-Bytes im Laufe der Zeit periodisch
  - gilt für die meisten I/O Jobs
- Bursts = Auf einzelne Datei schreiben
  - Größe der Datei = Summe geschriebener Bytes
- Checkpoint Status Anwendung
  - Datensicherung

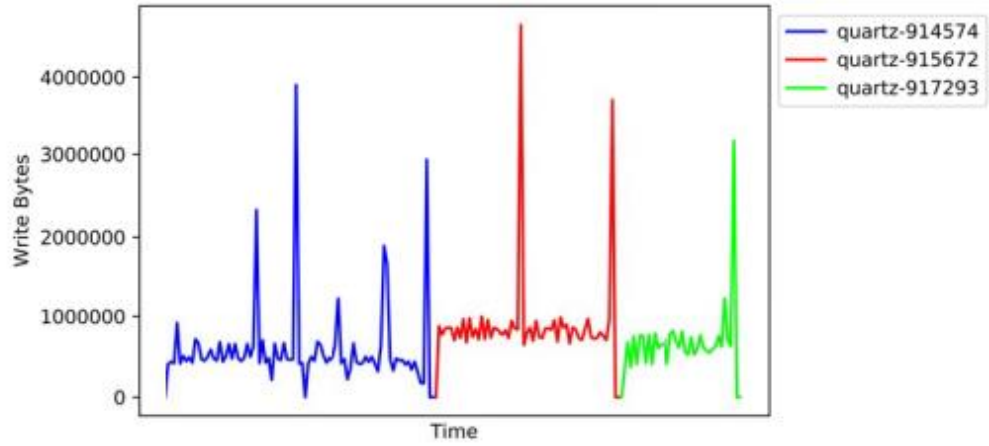


Fig. 11: Write Bytes written over time by 3 random jobs in Quartz.

Quelle: Literaturverzeichnis 1)

ANALYSE:
H. BEZIEHUNG ZWISCHEN SCHREIB-BYTES UND ARBEITSSPEICHER

- Schreib-Pattern aller Jobs betrachtet
- 1. Vergleich Schreib-Bursts mit Arbeitsspeicher
  2. Betrachtung I/O Laufzeit (pro Job)
- Ergebnis:
  - Großteil Jobs nutzen in 100% der I/O Zeit <1% Arbeitsspeicher
- Erkenntnis:
  - Arbeitsspeicher bei Schreib-Bursts nicht ausgelastet
  - Arbeitsspeicher für Prefetching nutzen

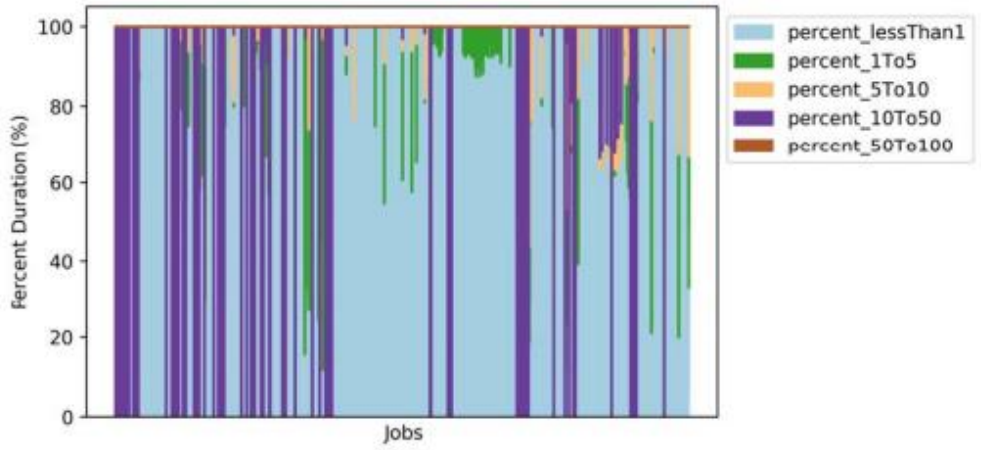


Fig. 12: % I/O duration vs. write burst size as % of memory.

Quelle: Literaturverzeichnis 1)

ANALYSE:
H. BEZIEHUNG ZWISCHEN SCHREIB-BYTES UND ARBEITSSPEICHER

- Betrachtung genaue Byteanzahl der Schreib-Bursts
- Ergebnis:
  - viele Jobs nutzen >90% Zeit für Schreib-Bursts 1MB-1GB
  - 73% Jobs nutzen 50% I/O Zeit für Schreib-Bursts 1KB-1MB
- Erkenntnis:
  - Signifikanter Anteil kleine Checkpoint-Dateien
  - Optimierung für Schreib-Bursts mit wenig Bytes

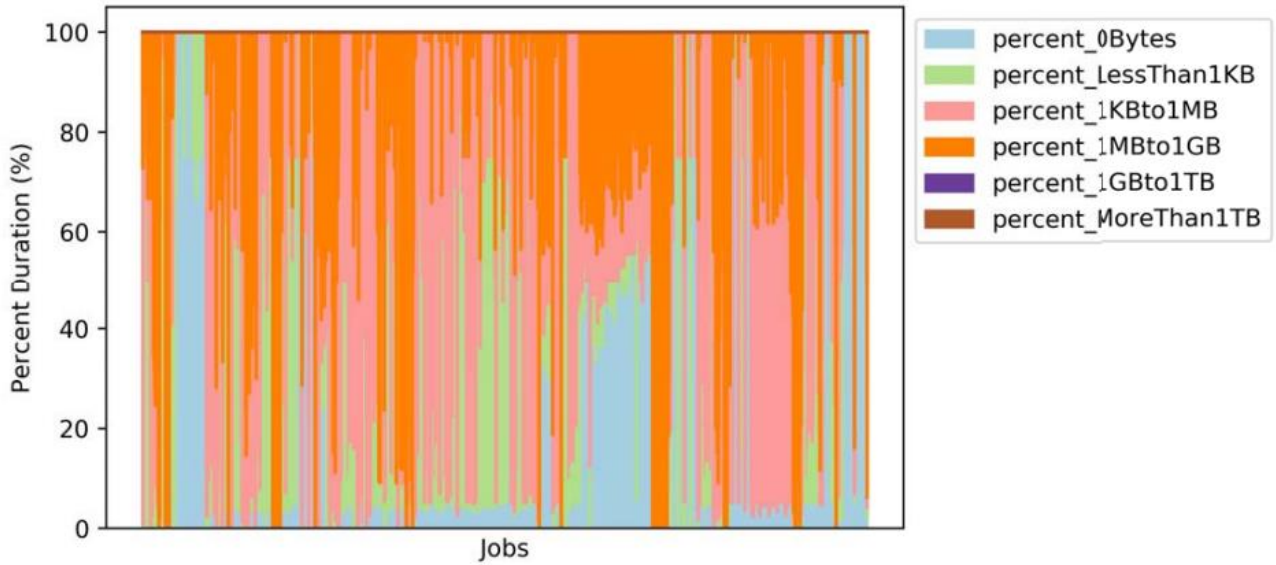


Fig. 13: Percent I/O duration vs size of write bursts.

Quelle: Literaturverzeichnis 1)

ANALYSE:
I. I/O CONTENTION HÖHEPUNKTE

- I/O Contention = Abnahme Leistung durch Wettbewerb I/O Ressourcen
  - Problematik bei parallelen Dateisystemen
- I/O Aktivität hinsichtlich Tageszeitpunkten betrachtet
- Ergebnis:
  - Höchste I/O Aktivität bei Jobs die zwischen 5AM und 11AM starten
- Erkenntnis:
  - Mehrere I/O-lastige Jobs zum selben Zeitpunkt führt zu Leistungsabnahme
    -> Ressourcenkonflikt
  - Entlastung durch Verteilung I/O Jobs

ZUSAMMENFASSUNG („RÜCKBLICK“)

- Verständnis von I/O Verhalten bekommen
- Typische Merkmale I/O-umfangreiche Rechenaufgaben (Jobs)
  - Gleichverteilung Lese- und Schreibzugriffe
  - Großteil Jobs kurze Laufzeit und wenige Knoten
  - Viele ineffiziente Schreibanfragen
- Beziehung zwischen I/O Anfragen und Metadaten(Server)
  - Je mehr geschriebene Bytes, desto mehr Metadaten-Operationen
  - Metadaten-Server nicht effizient ausgenutzt

ZUSAMMENFASSUNG („RÜCKBLICK“)

- Zeitliches Verhalten von I/O Verkehr
  - Nicht auf bestimmte Wochentage konzentriert
  - Byteanzahl Einfluss auf I/O Verkehr statt Job-Laufzeit
- Zusammenhang Anzahl geschriebener Bytes und Arbeitsspeicher
  - Arbeitsspeicher kaum ausgenutzt -> kein Prädiktor für Größe Schreib-Bursts
  - Kleine Checkpoint-Daten
- I/O Contention in Bezug auf Zeit
  - Bestimmte Tageszeitpunkte sehr hoher I/O Aktivität

LITERATURVERZEICHNIS

- 1) 2020 IEEE 27th International Conference on High Performance Computing, Data, and Analytics (HiPC): Understanding HPC Application I/O Behavior Using System Level Statistics (Arnab K. Paul et al.)
- 2) Introduction to Lustre Archictecture: https://wiki.lustre.org/images/6/64/LustreArchitecture-v4.pdf (aufgerufen 24.11.2021)
- 3) SLURM Workload Manager Documentation: https://slurm.schedmd.com/documentation.html (aufgerufen am 24.11.2021)
- 4) SLURM Manual (Bild): https://rdlab.cs.upc.edu/wp-content/uploads/documentation/manuals/html/manual_rdlab_hpc/index.html
- 5) Vorlesung Hochleistungsrechnen: PDF-Folien aus Cloud von Vorlesung 2021 (Prof. Dr. Thomas Ludwig)