

GWDG NACHRICHTEN 03|18

GitLab

.....
GWDG ownCloud

.....
HLRN-IV

.....
Spectre und Meltdown

.....
Plagiatspräventionsdienste

.....
Thread Scheduling on
Multi-core Systems

ZEITSCHRIFT FÜR DIE KUNDEN DER GWDG



 GWDG
Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen



GWDG NACHRICHTEN

03|18 Inhalt

.....

4 Kollaborative Sourcecode-Verwaltung mit GitLab **8 GWDG ownCloud: Upgrade auf Version X** **10 Mehr Rechenleistung für Norddeutschland** **12 Prozessor-Schwachstellen „Spectre“ und „Meltdown“** **16 Steigende Nutzung der Plagiatspräventionsdienste der GWDG** **17 Kurz & knapp** **18 Thread Scheduling on Multi-core Systems** **26 Stellenangebot** **28 Personalia** **29 Kurse**

Impressum

.....

Zeitschrift für die Kunden der GWDG

ISSN 0940-4686

41. Jahrgang

Ausgabe 3/2018

Erscheinungsweise:

monatlich

www.gwdg.de/gwdg-nr

Auflage:

550

Fotos:

© Maxim_Kazmin - Fotolia.com (1)

© teguhjatipras - Fotolia.com (4)

© momius - Fotolia.com (11)

© fotogestoeber - Fotolia.com (15)

© pterwort - Fotolia.com (25)

© nito - Fotolia.com (26, 27)

© MPIbpc-Medienservice (3, 28)

© Universität Göttingen / Jan Vetter (10, 11)

© GWDG (2, 29)

Herausgeber:

Gesellschaft für wissenschaftliche

Datenverarbeitung mbH Göttingen

Am Faßberg 11

37077 Göttingen

Tel.: 0551 201-1510

Fax: 0551 201-2150

Redaktion:

Dr. Thomas Otto

E-Mail: thomas.otto@gwdg.de

Herstellung:

Franziska Schimek

E-Mail: franziska.schimek@gwdg.de

Druck:

Kreationszeit GmbH, Rosdorf



Prof. Dr. Ramin Yahyapour
ramin.yahyapour@gwdg.de
0551 201-1545

Liebe Kunden und Freunde der GWDG,

die Verfügbarkeit von und der Zugang zu leistungsfähigen Rechenressourcen ist für viele Wissenschaftsdisziplinen erfolgskritisch. Aus diesem Grund gibt es den kontinuierlich steigenden Bedarf an High-Performance-Computing-Clustern. Wie in einer früheren Ausgabe der GWDG-Nachrichten berichtet und auch in der Presse zu lesen, wird Göttingen einer der beiden Betriebsstandorte für die vierte Generation des Hochleistungsrechners Nord (HLRN-IV). Zusammen mit den Kollegen vom Zuse-Institut Berlin wurden in den letzten Monaten die zugehörige Ausschreibung und Vergabe betreut. Diese sind nun erfolgreich zum Abschluss gekommen, so dass wir endlich über das Ergebnis berichten können.

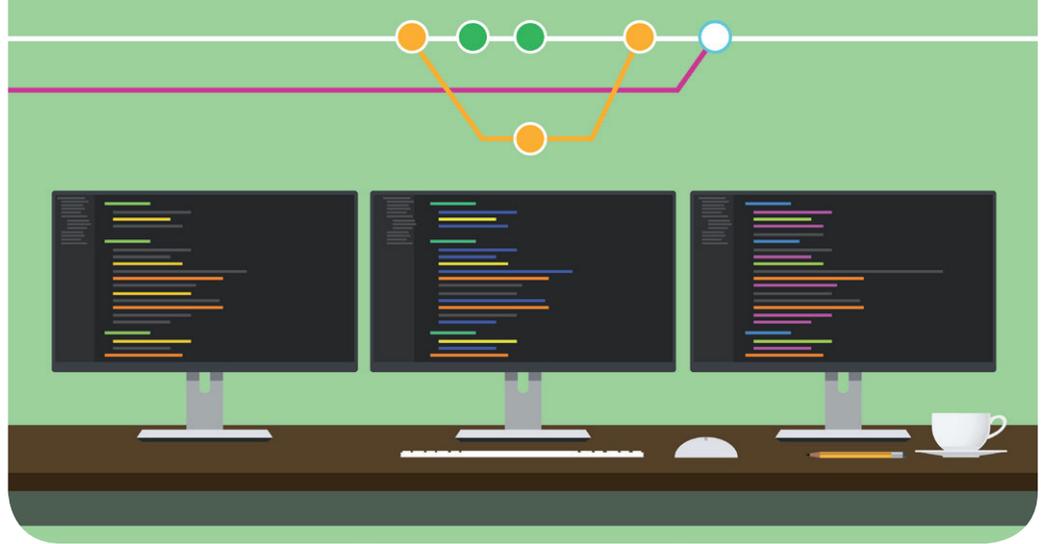
Der neue Supercomputer wird von der Firma Atos/Bull geliefert und über fast 250.000 Intel-Prozessorkerne verfügen. Das Rechnersystem wird dabei in zwei Phasen aufgebaut, wobei die erste Phase im September 2018 in Göttingen ihren Betrieb aufnehmen wird. Die zweite Phase wird dann im dritten Quartal 2019 aufgebaut. Die beiden Sites werden dann mit einer Gesamtleistung von etwa 16 PetaFlop/s etwa sechsmal schneller sein als das bisherige HLRN-III-System und damit insbesondere den Forschenden in Norddeutschland dringend benötigte neue Rechenkapazitäten liefern.

In den kommenden GWDG-Nachrichten werden wir weitere Informationen hierzu liefern und über den Fortschritt berichten.

Ich wünsche Ihnen viel Freude beim Lesen dieser Ausgabe.

Ramin Yahyapour

GWDG – IT in der Wissenschaft



Kollaborative Sourcecode-Verwaltung mit GitLab

Text und Kontakt:

Ralph Krimmel
ralph.krimmel@gwdg.de
0551 201-1821

Martina Brücher
martina.bruecher@gwdg.de
0551 201-2113

Mit GitLab stellt die GWDG einen Dienst zur Verfolgung von Änderungen zur Verfügung, die im Laufe der Zeit an einer Reihe von Dateien vorgenommen wurden. GitLab ist ein Versionsverwaltungssystem, das darüber hinaus weitere Funktionalitäten beinhaltet. Es ist primär dazu gedacht, Textdateien wie Quellcode, LaTeX-Dateien oder Skripte zu verwalten. Deshalb wird Git – die dem Dienst zugrunde liegende Versionierungs-Software – bevorzugt von Programmierern zur Koordination von Änderungen in Software-Quellcodes eingesetzt. In diesem Bereich wird sie auch GWDG-intern genutzt. Darüber hinaus ist GitLab für die Versionsverwaltung von Inhalten jeder Art potenziell geeignet.

WARUM VERSIONSVERWALTUNG?

Viele Nutzer kennen dieses Problem: Kopien von Dateien lassen sich einfach und unkompliziert erstellen. Existieren jedoch verschiedene Versionen z. B. einer Datei, kommt rasch Chaos auf, wie das nachfolgende Beispiel verdeutlicht: Person A arbeitet an Version 1, Person B kreierte aus einem Zwischenstand von Version 1 eine Version 2 und gibt diese an Person C weiter, die basierend auf Version 2 eine Version 3 erzeugt, während Person D gerade eine weitere Version 2 aus einem späteren Bearbeitungsstand von Version 1 erstellt usw. Typische Fehlerquellen bei diesem Vorgehen sind eine nicht eindeutige Kennzeichnung der Versionen oder die Ablage in unterschiedlichen Verzeichnissen oder gar auf verschiedenen lokalen Rechnern. In der Folge werden mitunter die falschen Dateien kopiert, überschrieben oder weitergegeben.

Bei der Softwareentwicklung ist das Arbeiten ohne Versionskontrolle besonders risikoreich. Programmierer entwickelten deshalb bereits vor längerer Zeit die unterschiedlichsten Versionsverwaltungssysteme, um ihre Arbeit sicherer zu machen und Fehlerquellen auszuschließen.

Allen Systemen gemein sind:

- Protokollierung aller Änderungen, wodurch jederzeit nachvollzogen werden kann, wer wann was geändert hat.
- Wiederherstellung alter Bearbeitungsstände, wodurch versehentliche Änderungen jederzeit revidiert werden können.
- Archivierung der einzelnen Bearbeitungsstände, wodurch

Collaborative Source Code Management with GitLab

With GitLab, the GWDG provides a service for tracking changes that have been made to a number of files over time. GitLab is a version control system that also includes additional functionalities. It is primarily intended to manage text files such as source code, LaTeX files or scripts. Git – the versioning software on which the service is based – is therefore preferred by programmers to coordinate changes in software source codes. In this area, it is also used internally by the GWDG. In addition, GitLab is potentially suitable for version management of all types of content.

auf alle existierenden Versionen zurückgegriffen werden kann.

- Koordinierung des Zugriffs durch mehrere Entwickler bzw. Bearbeiter.
- Parallele Arbeit an bzw. Weiterentwicklung von verschiedenen Versionen.

WAS IST GITLAB?

GitLab ist eine Open-Source-Webanwendung zur Versionsverwaltung auf Basis von **Git**, die 2011 von dem Ukrainer Dmitri Saporoschec gemeinsam mit Valery Sizov entwickelt wurde. Insbesondere in den Bereichen (Programmier-)Code- und Softwareentwicklung erlangte GitLab innerhalb kürzester Zeit große Bedeutung. Über 100.000 Organisationen nutzen mittlerweile GitLab, darunter Siemens, Bayer, Sony, die NASA, Intel und das CERN. Das CERN, die Europäische Organisation für Kernforschung, setzt GitLab zunehmend für Softwareprojekte und Code für das Konfigurationsmanagement ein [1].

Die verteilte Versionsverwaltung mit GitLab ist sehr anwenderfreundlich und deshalb auch für Nicht-Programmierer geeignet. Eine übersichtliche, graphische Benutzeroberfläche trägt nicht unerheblich hierzu bei. Über eine zentrale Plattform können Mitglieder eines Projektes beispielsweise Quellcode parallel bearbeiten, speichern, teilen, zusammenführen, veröffentlichen und testen. Darüber hinaus bietet GitLab zusätzliche Funktionen, um den gesamten Lebenszyklus der Softwareentwicklung zu verwalten. Hierzu zählen, neben der bereits angesprochenen Nutzerverwaltung, das „Issue-Tracking“, das beispielsweise zur Erfassung und Dokumentation von Programmfehlern, sog. Bugs, zum Einsatz kommt, sowie Wiki-Funktionalitäten. Vorteilhaft ist, dass Projekte nicht öffentlich sein müssen, wie es bei der kostenlosen Variante von **GitHub** der Fall ist. Die Bearbeitung kann ausschließlich auf eine definierte, geschlossene Gruppe beschränkt werden. Durch die Nutzung der Git-immanenten kryptographischen Hash-Algorithmen bietet GitLab hohe Sicherheit gegen Code-Verfälschung.

Derzeit wird GitLab nur für die Versionierung von Textdateien empfohlen, die lediglich aus den jeweils dargestellten druckbaren Zeichen bestehen, wie Programmiercode, LaTeX-Dateien oder Skripte. Für binäre Datenformate wie Bild-, Office- und Multimedia-Dateien sowie InDesign-Dokumente ist GitLab zwar grundsätzlich ebenfalls geeignet, es ist aber nicht die ideale Lösung. Für das Teilen von größeren Bild-, Office- und Multimedia-Dateien sowie InDesign-Dokumenten bietet die GWGD den Dienst „GWGD own-Cloud“ an.

Als webbasierte Oberfläche ermöglicht GitLab ein bequemes Arbeiten im Browser. Viele Git-typische Arbeitsabläufe sind dabei per graphischer Oberfläche nutzbar. Die Datenübertragung der von GitLab verwalteten Repositorien zum Endgerät des Nutzers und zurück kann über HTTPS oder SSH erfolgen. Letzteres bietet den Vorteil einer asymmetrischen Authentifizierung über sogenannte SSH-Public/Private Keys, ein passwortloses Verfahren für einen effizienten Arbeitsablauf an den Endgeräten. Wir empfehlen diese Methode ausdrücklich.

Die GWGD betreibt den Dienst auf ihren eigenen Servern. Sie überwacht und pflegt GitLab und sorgt für das Backup der Nutzerdaten, so dass die Anwender sich allein auf ihre Projekte konzentrieren können.

Git, GitLab, GitHub? – Eine häufige Quelle der Verwirrung

Git: Ist eine Versionierungs-Software zur Nachverfolgung von Änderungen von Dateien.

GitLab: Ist ein Dienst wie u. a. GitHub, der Dateien, meistens Programmier-Codes von Nutzern, hostet und so den Fernzugriff auf sog. Git-Repositories ermöglicht.

GitHub: Bezeichnen viele als den großen Bruder von GitLab, weil es den Dienst länger als GitLab gibt und er mehr Nutzer hat [3]. Im Unterschied zu GitLab gestattet GitHub allerdings nur dann eine kostenfreie Nutzung, wenn es sich um öffentlich zugängliche Projekte handelt. Soll ein Projekt nur innerhalb eines geschlossenen Projektteams bearbeitet werden, fallen Kosten an.

DIE GITLAB-TERMINOLOGIE

Ein „**Projekt**“ ist die virtuelle Arbeitsumgebung bzw. der Bereich für eine definierte Aufgabe. Ein Nutzer kann mehrere Projekte anlegen. Im Projekt können u. a. die zugehörigen Dateien abgelegt sowie die Arbeit geplant und verteilt werden. Ein Wiki zur Veröffentlichung von Dokumentation steht ebenfalls zur Verfügung. Wird ein Projekt neu erstellt, sind alle GitLab-Funktionen aktiviert (siehe auch Abb. 1 und 2). Funktionen, die nicht benötigt werden, können deaktiviert werden.

Ein „**Repository**“ ist Teil eines Projektes. Hier werden die Dateien abgelegt. Ein Repository ist ein verwaltetes Verzeichnis zur Speicherung und Beschreibung von digitalen Objekten.

Der „**Issue Tracker**“ ist der Ort, um Dinge hinzuzufügen, die in einem Projekt verbessert oder gelöst werden müssen. Hierbei kann es sich um Bugs, Aufgaben oder Ideen handeln, die diskutiert und kommentiert werden können.

„**Milestones**“ fassen mehrere „Issues“ zu einem Teilabschnitt oder Ziel des Projektes zusammen.

„**Branches**“ sind Git-Verzweigungsstrategien, die es ermöglichen, gemeinsam an einem Code zu arbeiten. Hierbei ist es möglich, Zweige zu schützen, um zu verhindern, dass Code ohne Überprüfung in den Hauptentwicklungszweig einfließt.

„**Mergen**“ bedeutet in Git das Zusammenführen von mehreren „Branches“.

„**Merge Requests**“ betitelt die Anfrage eines Projektteilnehmers, einen bestimmten Branch mit einem anderen Branch zusammenzuführen. Die Änderungen, die an einem Projekt vorgenommen wurden, können so mit anderen Teilnehmern diskutiert und verbessert werden, bis sie den Qualitätsstandards des Projektes genügen und dann von einem Projektadministrator „merged“ werden.

DIE VORTEILE VON GITLAB IM ÜBERBLICK

- Das Benutzerinterface ermöglicht die einfache Verwaltung von Projekten, Nutzern, Gruppen, Änderungen und Statistiken auf dem Desktop.
- Umfangreichere Wiki-basierte Dokumentationen, z. B. zur Fehlerbeschreibung, können erstellt werden.
- Es lassen sich individuell modifizierte

Zugangsberechtigungen für verschiedene Team-Mitglieder generieren. Z. B. kann eine Berechtigung für den Issue Tracker ohne eine Erlaubnis für Änderungen am Quellcode erteilt werden.

- Der GitLab Issue Tracker ist ein Werkzeug zum Mel- den und Nachverfolgen von Fehlern oder Fehlfunktio- nen, zur Diskussion neuer Ideen oder der Einreichung von Feature-Vorschlägen.
- „Code Snippets“ erlauben es, auch kleinste Code-Teile eines Projektes zu teilen, ohne das gesamte Projekt kopie- ren zu müssen.
- Bei der Zusammenarbeit mit mehreren Personen lassen sich Überprüfungen von Änderungen einfach über sogea- nannte „Merge Requests“ durchführen.
- Sicherheitsfeatures schützen Codes vor Änderungen und ermöglichen es Benutzern, nur bestimmte Rechte zu erteilen oder ein Projekt komplett nicht-öffentlich laufen zu lassen.
- Pro Projekt kann „Continuous Integration“ konfiguriert werden. Das Verfahren findet bei der Softwareentwick- lung Anwendung und ist ein Prozess eines fortlaufenden Zusammenfügens von Komponenten. Das Ziel ist die Ver- besserung der Softwarequalität. Dabei wird die gesame- te Anwendung neu gebaut. Automatisierte Tests wer- den durchgeführt und Metriken zur Messung der Soft- warequalität erstellt. Der gesamte Vorgang wird nach ent- sprechender Konfiguration automatisch ausgelöst.
- Ein kleines, aber nützliches Werkzeug ist die Möglichkeit, einen „Merge Request“ mit dem Status „Work in Pro- gress“ zu versehen, um zu signalisieren, dass die Arbeit noch nicht abgeschlossen ist. So kann beispielsweise ver- hindert werden, dass unfertiger Code „merged“ wird.
- Mit dem integrierten Tool „GitLab Pages“ lassen sich unter Verwendung von CI/CD Webseiten automatisiert bauen und hosten. Besonders geeignet sind hierfür sogea- nannte „Static Site Generators“. So können Sie schnell eine einfache Projektseite unter <https://<projektname/>username>.pages.gwdg.de> erstellen.

WIE KÖNNEN SIE GITLAB BEI DER GWGD NUTZEN?

Die Beschreibung des GWGD-Dienstes „GitLab“ ist im Kundenportal der GWGD unter dem URL <https://www.gwdg.de/e-mail-collaboration/gitlab> zu finden. Dort findet sich unter „Schnellzugriff“ der Link, der direkt zur Anmeldeseite <https://gitlab.gwdg.de> führt. Nutzen Sie dort für die Anmeldung nur die Variante „E-Mail-Address“ und geben dort Ihre E-Mail-Adre- se und das dazugehörige Passwort ein. Hierbei sind zwei Nutzer- gruppen zu unterscheiden:

Hätten Sie es gewusst?

Git ist eine Entwicklung von Linus Torvalds, dem Schöp- fer von „Linux“, die 2005 aus der Not heraus entstand, weil die Linux-Kernel-Entwickler infolge einer Lizenzänderung das bis dahin eingesetzte Werkzeug zur Versionsverwaltung nicht mehr nutzen konnten. Um den Namen „Git“ ranken sich mitt- lerweile Legenden. Umgangssprachlich bedeutet „Git“ im bri- tischen Englisch „Schwachkopf“ oder „Blödmann“. Torvalds wird wie folgt zitiert: “I’m an egotistical bastard, and I name all my projects after myself. First ‘Linux’, now ‘Git’.” Ande- re Quellen belegen, dass er das Kurzwort „Git“ als praktika- bel befunden habe, da es im Bereich Softwareentwicklung bis dato weitgehend unbenutzt war. [2]

1. Mitarbeiter und Studierende der Universität Göttingen sowie Mitarbeiter der Max-Planck-Gesellschaft, die an Instituten arbeiten, die mit dem Identity Management der GWGD verbunden sind, können sich direkt bei GitLab anmelden.
2. Mitarbeiter eines Max-Planck-Instituts, das nicht am Iden- tity Management der GWGD angeschlossen ist, und externe Benutzer, die mit einem Benutzer der unter 1. genannten Gruppen zusammenarbeiten möchten, kön- nen nach erfolgreicher Registrierung im Kundenportal der GWGD unter dem URL [https://www.gwdg.de/ registration](https://www.gwdg.de/registration) auf Wunsch für die GitLab-Nutzung freige- schaltet werden. Bitte kontaktieren Sie hierzu unsere Ser- vice-Hotline unter support@gwdg.de. Wie in den letz- ten GWGD-Nachrichten 1-2/2018 im Artikel „Einführung von Single Sign-on für Dienste der GWGD“ angekün- digt, wird momentan an der Integration von GitLab in die Single-Sign-on-Lösung gearbeitet. Nach erfolgreichem Abschluss wird sich dadurch dann die Anmeldung an bzw. Nutzung von GitLab vereinfachen.

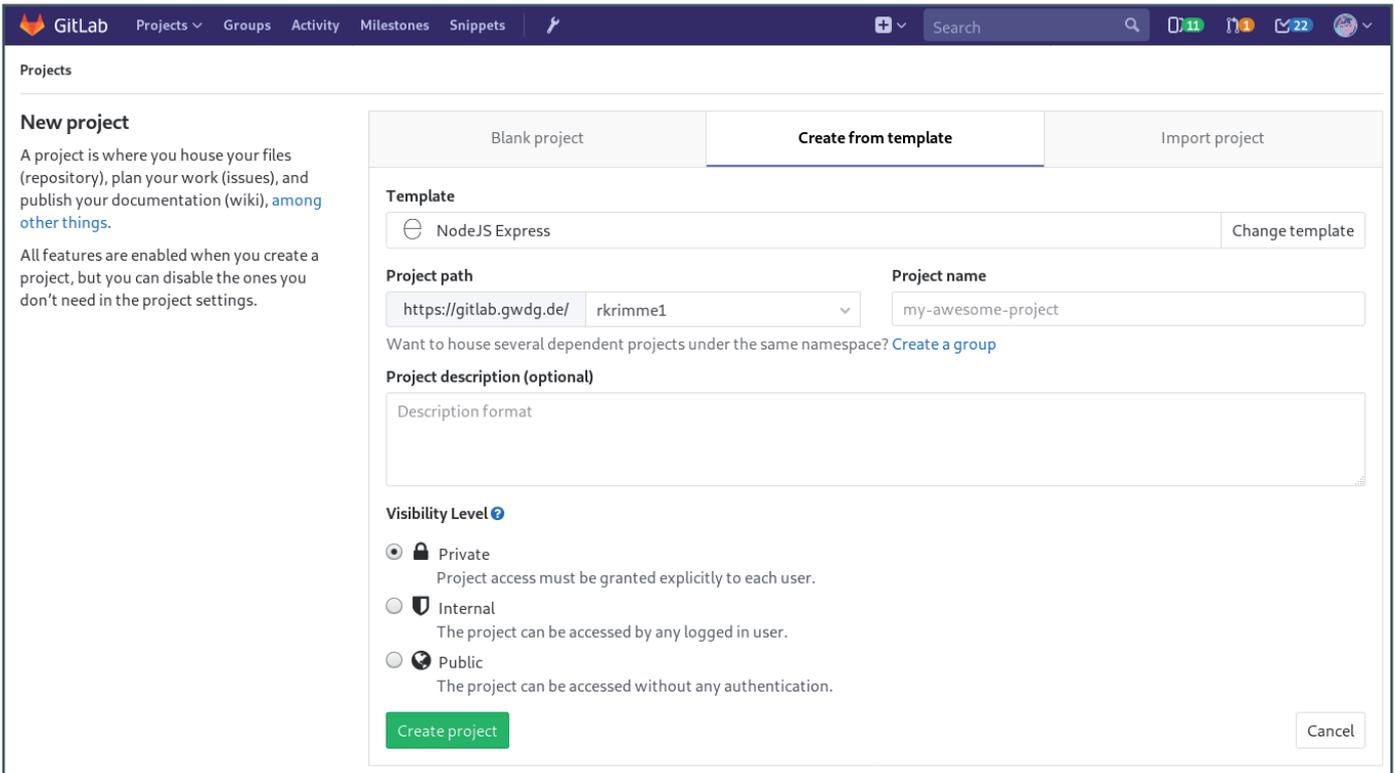
FUSSNOTEN

Sofern nicht gesondert zitiert, basiert der Text auf der Doku- mentation der GWGD zu GitLab (https://info.gwdg.de/docs/doku.php?id=de:services:email_collaboration:gitlab:start), dem Tutorial „Learn Git“ ([https://www.atlassian.com/git/tutorials/what-is-ver- sion-control](https://www.atlassian.com/git/tutorials/what-is-version-control)) und der GitLab Documentation (<https://docs.gitlab.com/ee/README.html>).

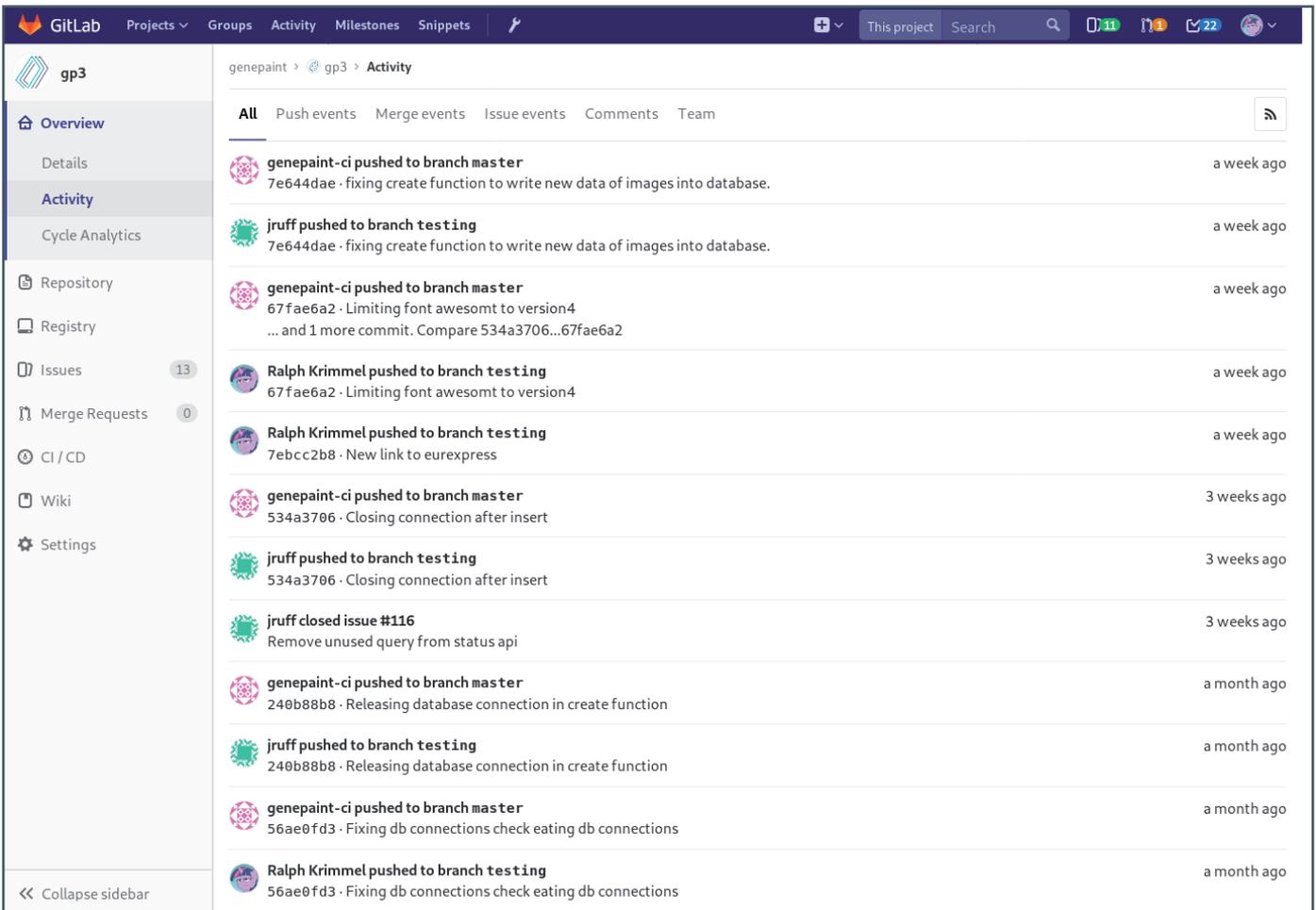
[1] <https://about.gitlab.com/customers>

[2] Alexander Neumann: Vor 10 Jahren: Linus Torvalds baut Git, Heise Developer vom 08.04.2015 (<https://www.heise.de/developer/meldung/Vor-10-Jahren-Linus-Torvalds-baut-Git-2596654.html>)

[3] <https://github.com> ■



1_Einrichtung eines neuen Projektes in GitLab



2_GitLab-Arbeitsumgebung eines Projektes

GWDG ownCloud: Upgrade auf Version X

Text und Kontakt:

Tobias Herbst
tobias.herbst@gwdg.de
0551 201-2135

Ende Dezember 2017 wurde der Sync&Share Dienst „GWDG ownCloud“ auf Version 10.0.4 angehoben. Mit dem Upgrade wurden u. a. drei wesentliche Neuerungen eingeführt: neue Gruppenfunktionen in Form sogenannter „Custom Groups“, flexiblere Sharing-Möglichkeiten sowie File-Integrität durch die verbesserte Kommunikation zwischen Server und Desktop-Client. Zusätzlich wurde das Anmeldeverfahren auf den neuen, auf OAuth2 basierenden Login umgestellt und eine offene Testphase für die web-basierte Office-Suite „Only Office“ gestartet.

GRUPPEN FÜR ALLE

Eine oft gewünschte Neuerung bieten „Custom Groups“. Für das Bereitstellen von Gruppen waren bisher administrative Eingriffe nötig, denn es erforderte das Anlegen der Gruppe und das Hinzufügen von Benutzern im genutzten Verzeichnisdienst (bei der GWDG dem zentralen LDAP-Dienst) und der anschließenden Übertragung in GWDG ownCloud. Mit der neuen Funktion zur Erstellung individueller Gruppen hat nun jeder Anwender die Möglichkeit, weitere Nutzer zu einer Gruppe hinzuzufügen und die

Rechtevergabe in dieser Gruppe entsprechend zu verwalten. Die Funktion finden Nutzer als Unterpunkt „Benutzerdefinierte Gruppen“ in den Einstellungen.

Ein Wermutstropfen: Diese Gruppen sind nicht mit den Gruppen der Kontakte-App verknüpft. Es ist nicht möglich, an Custom Groups Nachrichten zu verschicken, genauso wie es auch nicht möglich war und ist, Shares an die Gruppen der Kontakte-App zu verteilen. Es ist auch nicht klar, ob diese Verknüpfung überhaupt möglich ist, da das modulare Konzept der ownCloud-Apps dafür gedacht ist, dass die Anwendungen unabhängig voneinander funktionieren sollen.

Edit link share: /Presentation

Link Name
Name

Read only
Users can view and download contents.

Read & Write
Users can view, download, edit and upload contents.

Upload only (File Drop)
Receive files from others without revealing the contents of the folder.

Passwort
Wählen Sie ein Passwort für den öffentlichen Link

Ablaufdatum
02-03-2018

Link als E-Mail verschicken Link per E-Mail versch

Speichern Abbrechen

DOPPELT GETEILT HÄLT BESSER

Eine Datei oder ein Ordner konnte bisher nur auf genau einem öffentlichen Share geteilt werden und dadurch konnten auch nur einmal Zugriffsberechtigungen, ein Ablaufdatum oder Passwort vergeben werden. Anwender haben nun die Option, beliebig viele Public Links zu erstellen (siehe Abb. 1). Damit wird eine selektive Datenfreigabe ermöglicht, indem pro Link unterschiedliche Passwörter und Ablaufdaten erstellt werden können. Will ein Anwender also eine Datei oder einen Ordner an verschiedene Personen oder Organisationen mit unterschiedlichen gewünschten Zugriffsrechten freigeben, muss er (etwa bei geänderten Freigabebedingungen) nun nicht mehr nacheinander den einen Link manuell deaktivieren und dafür einen neuen erstellen, sondern definiert einmalig die Zugriffsrechte und Ablaufdaten mit unterschiedlichen Links.

GWDG ownCloud: Upgrade to Version X

The service "GWDG ownCloud" was upgraded to the latest version in late December 2017. The upgrade adds custom groups for easier sharing, an improved user experience for public shares and checks to improve file integrity during transport. Additionally a new login service based on OAuth2 and an open testphase for the web-based office suite "Only Office" was started.

Mit dieser Funktion kann auch jeder beliebige Ordner zu einem „File Drop“ gemacht werden. Dies war bisher eine eigene App und auf einen einzelnen Ordner beschränkt, nun kann für jeden Link die Berechtigung „Upload only“ gesetzt werden.

NEUES ANMELDEVERFAHREN

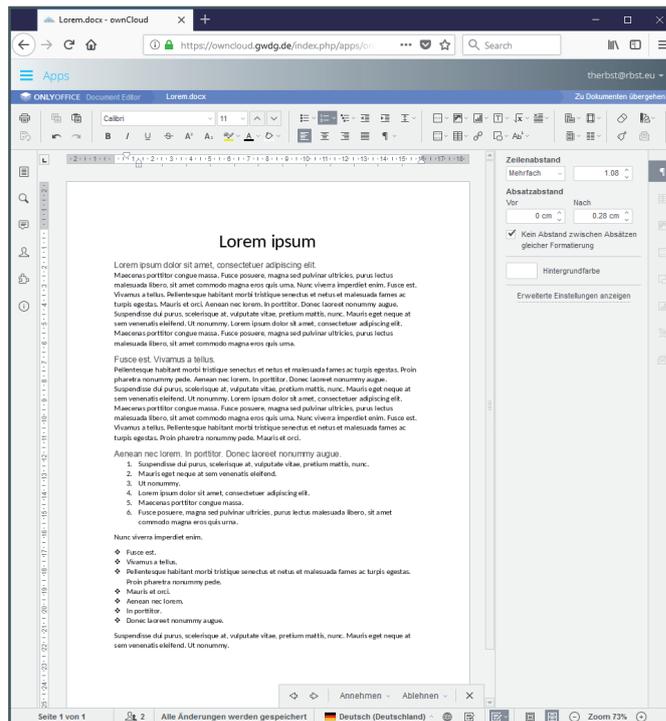
Auch im Zusammenspiel mit den Desktop- und Mobile-Clients hat sich etwas verändert: Die Anmeldung der Browser und Clients an GWDG ownCloud funktioniert nun über den neuen Single-Sign-on-Server der GWDG, der in den GWDG-Nachrichten 1-2/2018 vorgestellt worden ist. Nach erfolgreicher Anmeldung benutzen die Clients das Verfahren „OAuth2“. Dadurch müssen Nutzer ihr Passwort nicht mehr auf dem lokalen Computer speichern und es ist in einfacher Weise möglich, z. B. beim Verlust des Smartphones, die Zugriffsrechte des Gerätes zu widerrufen.

ÜBERPRÜFUNG DER FILE-INTEGRITÄT

Die Überprüfung der Integrität von Dateien kann durch die Berechnung von Prüfsummen bei der Kommunikation zwischen Server und Desktop-Client etwaige Übertragungsfehler (z. B. Bit-Fehler) ermitteln, was die Zuverlässigkeit des Sync-Prozesses künftig signifikant steigern wird.

ONLYOFFICE IN GWDG OWNCLOUD

Mit OnlyOffice steht die Nutzern bereits aus GWDG Cloud Share bekannte Suite aus Textverarbeitung, Tabellenkalkulation und Präsentationssoftware nun auch in GWDG ownCloud



2_Textverarbeitung mit Only Office

zur Verfügung (siehe Abb. 2). OnlyOffice erzeugt und verarbeitet Dateien in den Office-Open-XML-Formaten für Word, Excel oder PowerPoint. Es erlaubt mehreren Nutzern gleichzeitiges Arbeiten an einer Datei, bietet ausgereifte Funktionen und einen Chat zum jeweiligen Dokument.

GWDG Cloud Share: Stufenweise Einstellung ab März 2018

Der Datei-Synchronisationsdienst **GWDG Cloud Share** wird ab März 2018 stufenweise eingestellt und Ende 2018 abgeschaltet.

Warum wird der Dienst abgeschaltet, was ist die Alternative?

Die GWDG bietet im Rahmen von **GWDG ownCloud** basierend auf ownCloud einen Dienst für Dateisynchronisierung für ihre Nutzer an. GWDG ownCloud erfüllt dieselben Anforderungen wie GWDG Cloud Share, so dass für einen effizienteren Betrieb und bessere Benutzerbetreuung das Angebot in 2018 unter GWDG ownCloud zusammengefasst wird.

Bis wann kann GWDG Cloud Share noch genutzt werden?

- GWDG Cloud Share kann noch bis einschließlich April 2018 wie gewohnt genutzt werden. Quota-Erhöhungen sind nicht mehr möglich.
- Ab Mai 2018 können neue und eingeladene Benutzer sich am Dienst noch registrieren, erhalten aber keine Quota mehr. Das bedeutet, alle neuen Benutzer können ab Mai 2018 keine eigenen Folder mehr anlegen, können aber in den Foldern anderer Benutzer weiter arbeiten. Für bereits registrierte Benutzer ändert sich noch nichts.

- Ab August 2018 wird allen Benutzern des Dienstes die Quota entzogen. Damit sind keine Änderungen an Daten oder Hochladen neuer Daten mehr möglich, vorhandene Daten können aber noch über die Webseite oder den Client heruntergeladen werden.
- Am Ende des Jahres 2018 werden die Server des Dienstes abgeschaltet.

Was passiert mit Ihren Daten in GWDG Cloud Share?

Der Zugriff auf alle Daten bleibt bis zum Ende des Jahres 2018 möglich. Arbeiten mit Ihren vorhandenen Daten und das Einladen weiterer Benutzer zu Ihren Foldern sind noch bis zum August 2018 möglich. Mit Abschaltung des Dienstes Ende 2018 werden alle Daten gelöscht, ein Zugriff ist dann nicht mehr möglich.

Können Folder und Daten nach GWDG ownCloud übertragen werden?

Bis zum Mai 2018 wird eine Webseite eingerichtet, über die Benutzer selbstständig den Transfer ihrer Folder und Daten nach GWDG ownCloud starten können.

Herbst

Mehr Rechenleistung für Norddeutschland

Text:
Presseinformation Nr. 55/2018 der
Georg-August-Universität Göttingen
vom 07.03.2018

Norddeutsche Spitzenforscher können bald noch präzisere Modellrechnungen durchführen. Möglich macht dies ein 30 Millionen Euro teurer Supercomputer, der HLRN-IV, den der HLRN-Verbund beschafft. Die entsprechenden Kaufverträge wurden am 7. März 2018 in Göttingen und Berlin, den beiden Standorten des Hochleistungsrechners, unterzeichnet. Das neue System wird mit einer Gesamtleistung von 16 PetaFlop/s etwa sechsmal schneller sein als das bisherige und in zwei Phasen 2018 und 2019 an den beiden Betreiberstandorten installiert.

Zur Unterstützung der Spitzenforschung beschafft der Norddeutsche Verbund für Hoch- und Höchstleistungsrechnen (HLRN-Verbund) einen neuen Supercomputer, der mit knapp einer Viertelmillion Rechenkernen etwa sechsmal schneller als das bisherige System ist. Damit werden künftig noch präzisere Modellrechnungen möglich, beispielsweise in der Umweltforschung, den Lebens-, Material- und Ingenieurwissenschaften sowie in der Grundlagenforschung in Physik, Chemie und Mathematik. Die Firma Atos in Deutschland wird den neuen Computer für den HLRN-Verbund in Berlin und Göttingen installieren. Betreiber sind wie bisher das Zuse-Institut Berlin (ZIB) und erstmals die Universität Göttingen. An den Investitionskosten von insgesamt etwa 30 Millionen Euro an beiden Standorten beteiligt sich das Land Niedersachsen mit fünf Millionen Euro.

„Hochleistungsrechner haben sich längst zu einer Schlüsseltechnologie für unsere Wissenschaftslandschaft entwickelt – exzellente Forschungsleistungen sind in den meisten Disziplinen ohne sie inzwischen undenkbar“, sagte Dr. Sabine Johansen, Staatssekretärin im Niedersächsischen Ministerium für Wissenschaft und Kultur, bei der Vertragsunterzeichnung in Göttingen. „Der Leistungszuwachs des neuen Systems ist beeindruckend. Die Möglichkeiten, die sich daraus für die Weiterentwicklung der norddeutschen Spitzenforschung ergeben, sind vielversprechend.“

„Der neue Rechner stellt eine deutliche Leistungssteigerung für den Standort Göttingen und den gesamten HLRN-Verbund

dar“, so Prof. Dr. Norbert Lossau, Vizepräsident für Infrastrukturen der Universität Göttingen. „Die Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen betreibt seit vielen Jahren Ressourcen für wissenschaftliches Rechnen. Wir freuen uns, dass diese Kompetenz nun für den Betrieb des HLRN-Rechners genutzt wird.“

Der neue Supercomputer HLRN-IV hat eine Leistung von etwa 16 PetaFlop/s. Das System basiert größtenteils auf Intel Xeon-Prozessoren der Cascade Lake-Generation. Insgesamt umfasst der Rechner etwa 244.000 Prozessorkerne. Die erste Phase der Installation beginnt im Herbst 2018, im Jahr 2019 folgt der Vollausbau. Die Betriebskosten für beide Standorte über fünf Jahre liegen bei insgesamt etwa 20 Millionen Euro, die von den Ländern aufgebracht werden.

Seit der Installation des ersten Supercomputers HLRN-I im Jahr 2002 verfolgt der HLRN das Konzept mit zwei Betreiberstandorten. Neben dem bisherigen Betreiber ZIB kommt die Universität Göttingen als Betreiber nun erstmals hinzu. Mit der Gesellschaft

New Supercomputer HLRN-IV

To support cutting-edge research, the North German Supercomputing Alliance (Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen – HLRN) purchases a new supercomputer which, with nearly a quarter of a million cores, is about six times faster than the previous system.

The company Atos in Germany will install the new computer for the HLRN alliance in Berlin and Göttingen. As in the past, the Zuse Institute Berlin (ZIB) and, for the first time, the University of Göttingen will manage the system. The state of Lower Saxony contributes five million euros to the investment costs of a total of around 30 million euros at both locations.

The new supercomputer HLRN-IV has a performance of about 16 PetaFlop/s. The system is based mainly on Intel Xeon processors of the Cascade Lake generation. The computer contains a total of about 244,000 processor cores. The first phase of the installation will begin in the autumn of 2018, followed by full expansion in 2019. The operating costs for both sites over a period of five years will be in total around 20 million euros, which will be financed by the federal states.



Vertragsunterzeichnung am 7. März 2018 in Göttingen



Teilnehmer an der Vertragsunterzeichnung (von links): Thomas Theissen, Prokurist Atos, Dr. Philipp Wieder, stellvertretender Leiter der GWDG, Dr. Sabine Johannsen, Staatssekretärin im MWK, Prof. Dr. Norbert Lossau, Vizepräsident für Infrastrukturen der Universität Göttingen, Prof. Dr. Ulrike Beisiegel, Präsidentin der Universität Göttingen, und Peter Waue, MWK.

für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) besitzt die Universität ein gemeinsames Rechen- und IT-Kompetenzzentrum mit der Max-Planck-Gesellschaft. Beide Betreiberzentren verfügen über langjährige technische und organisatorische Erfahrung im Betrieb und in der Weiterentwicklung von Hoch- und Höchstleistungsrechnern. Eine dedizierte, redundant ausgelegte Glasfaserleitung wird zwischen Berlin und Göttingen die gemeinsame systemtechnische Betreuung von beiden Seiten sicherstellen.

Im HLRN-Verbund schlossen sich 2001 die Länder Berlin,

Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen und Schleswig-Holstein zusammen, um den wachsenden Bedarf der Wissenschaft an Rechen- und Speicherkapazität zu adressieren. Im Jahr 2012 stieß das Land Brandenburg hinzu. In der deutschen Landschaft für Hochleistungsrechnen bildet der Verbund eine Ausnahme, indem er den wissenschaftlichen Bedarf von sieben Bundesländern mit insgesamt 120 Hochschulen und 171 Forschungseinrichtungen bündelt. Der HLRN verfolgt das Ziel, das Hochleistungsrechnen im Interesse der Wissenschaft nachhaltig weiterzuentwickeln und durch spezialisierte Dienstleistungen für die Spitzenforschung nutzbar zu machen.

Atos ist ein weltweit führender Anbieter für die digitale Transformation mit etwa 100.000 Beschäftigten in 72 Ländern und einem Jahresumsatz von rund 13 Milliarden Euro. Als europäischer Marktführer für Big Data, Cyber Security, High Performance Computing und Digital Workplace unterstützt Atos Unternehmen mit Cloud Services, Infrastruktur- und Datenmanagement sowie Business- und Plattformlösungen. Hinzu kommen Services der Tochtergesellschaft Worldline, dem europäischen Marktführer für Zahlungsverkehrs- und Transaktionsdienste. Atos begleitet die digitale Transformation von Kunden aus so unterschiedlichen Marktsegmenten wie dem Bankensektor, der Bildung, Energie und Versorgung, Gesundheit, Produktion, Telekommunikation, Transport und Logistik sowie Versicherungen. Der Konzern ist der weltweite IT-Partner der Olympischen und Paralympischen Spiele. Atos SE ist an der Pariser Börse als eine der 40 führenden französischen Aktiengesellschaften notiert.

Weitere Informationen sind unter www.hlrn.de, www.gwdg.de und www.atos.net zu finden. ■



FTP-Server

Eine ergiebige Fundgrube!

Ihre Anforderung

Sie möchten auf das weltweite OpenSource-Softwareangebot zentral und schnell zugreifen. Sie benötigen Handbücher oder Programmbeschreibungen oder Listings aus Computerzeitschriften. Sie wollen Updates Ihrer Linux- oder FreeBSD-Installation schnell durchführen.

Unser Angebot

Die GWDG betreibt seit 1992 einen der weltweit bekanntesten FTP-Server, seit sieben Jahren mit leistungsfähigen Ressourcen für schnellen Service.

Ihre Vorteile

- > Großer Datenbestand (50 TByte), weltweit verfügbar
- > Besonders gute Anbindung im GÖNET



- > Aktuelle Software inkl. Updates der gebräuchlichsten Linux-Distributionen
- > Unter pub befindet sich eine aktuell gehaltene locatedb für schnelles Durchsuchen des Bestandes.
- > Alle gängigen Protokolle (http, https, ftp und rsync) werden unterstützt.

Interessiert?

Wenn Sie unseren FTP-Server nutzen möchten, werfen Sie bitte einen Blick auf die u. g. Webseite. Jeder Nutzer kann den FTP-Dienst nutzen. Die Nutzer im GÖNET erreichen in der Regel durch die lokale Anbindung besseren Durchsatz als externe Nutzer.

>> www.gwdg.de/ftp-server

Prozessor-Schwachstellen „Spectre“ und „Meltdown“

Text und Kontakt:

Dr. Tim Ehlers
tim.ehlers@gwdg.de
0551 201-1520

Dr. Vanessa End
vanessa.end@gwdg.de
0551 201-2176

Spectre und Meltdown bedrohen unsere moderne IT-Infrastruktur. Die Lücken befinden sich praktisch in allen modernen Geräten auf Hardware-Ebene und untergraben die Abschottung von Informationen zwischen Nutzern, Programmen und dem Gerät. Insbesondere die Variante 2 der Spectre-Lücke ist nur schwer per Software zu beheben. Dieser Artikel beschreibt in kurzen Worten den aktuellen Wissensstand über die Lücken und die möglichen Maßnahmen, um eine Ausnutzung zu verhindern.

ÜBERBLICK

Seit Anfang des Jahres sind die Themen „Spectre“ und „Meltdown“ fester Bestandteil der IT-Tagesnews. Dabei werden Fakten vom Vortag relativiert, korrigiert oder detaillierter nachgeliefert. Sicherheits-Updates werden wieder zurückgezogen, da sie offenbar zu voreilig herausgegeben wurden. Selbst für Leser, die sich intensiv mit dem Thema beschäftigen, ist es schwer, dabei den Überblick zu behalten. Dieser Artikel soll den aktuellen Wissensstand über die Lücken beleuchten. Des Weiteren werden die Auswirkungen auf die HPC-Systeme der GWDG und die damit einhergehenden Maßnahmen beschrieben.

Die Sicherheitslücken **CVE-2017-5753** [1] (Spectre 1, Bounds Check Bypass), **CVE-2017-5715** [2] (Spectre 2, Branch Target Injection) und **CVE-2017-5754** [3] (Spectre 3 oder Meltdown, Rogue Data Cache Load) wurden Anfang 2018 öffentlich bekannt. Während die Spectre-Lücke „Meltdown“ nur Intel- (und einige ARM-) Prozessoren betrifft, betreffen die Spectre-Lücken Variante 1 und 2 praktisch alle modernen Prozessoren sämtlicher Hersteller.

Was ist passiert? Eine Forschergruppe fand heraus, dass Daten aus spekulativer Ausführung der CPU abgegriffen werden können. Moderne CPUs führen Instruktionen bereits spekulativ aus, bevor diese angefordert werden, um dann im Anforderungsfall ohne Wartezeit bereits mit dem Ergebnis aufwarten zu können. War die spekulative Ausführung falsch, wird der Pfad von der CPU wieder gelöscht. Falls diese spekulative Ausführung einen Speicherzugriff enthält, wird der entsprechende Inhalt auch bereits aus dem langsamen Hauptspeicher in den schnellen Zwischenspeicher des Prozessors (Cache) geladen. Gerade hierdurch ergeben sich die großen Geschwindigkeitsgewinne dieses Verfahrens. Durch geschicktes Ausnutzen bestimmter Eigenschaften dieses spekulativen Ladens und der Tatsache, dass der Zugriff auf Daten im Cache um Größenordnungen schneller ist als auf Daten im Hauptspeicher, ermöglichen die drei Angriffe in unterschiedlicher Weise Lesezugang zu Speicherbereichen anderer Prozesse oder des Kernels. Da der Abgriff der Daten nicht auf direktem Wege erfolgt, spricht man auch von einem Seitenkanalangriff.

Bei einem erfolgreichen Angriff sind alle Geheimnisse

(Passwörter, Credentials, allg. Daten) in Gefahr, abgegriffen zu werden. Auf Einzelplatzsystemen besteht im Wesentlichen durch Angriffe via Javascript Gefahr. Hierzu können und sollten natürlich die bereits zur Verfügung gestellten Browser-Updates eingespielt werden. Insbesondere auf Systemen mit mehreren Benutzern (Login-Systeme, der High-Performance-Computing-Bereich [HPC], Cloud Server) besteht aber Handlungsbedarf.

SPECTRE VARIANTE 1 UND MELTDOWN

Spectre Variante 1 ist mit einem Kernel-Update ohne nennenswerte Performance-Probleme einfach zu schließen. Spectre Variante 3 (Meltdown) ist mit Performance-Einbußen durch ein Kernel-Update mit „Kernel Page Table Isolation“-Unterstützung (KPTI) zu schließen. Die KPTI-Unterstützung ist schon länger als Patch für den Linux-Kernel unter dem Namen „Kernel Address Isolation to have Side-channels Efficiently Removed“ (KAISER) verfügbar, deren Implementierung ursprünglich gar nichts mit Meltdown zu tun hatte. 2014 wurde für Linux eine sog. „Kernel Address Space Layout Randomization“ (KASLR) eingeführt, die das Ausnutzen anderer Schwachstellen verhindern sollte. Da sich aber bei einigen modernen Prozessoren durch andere Formen von Seitenkanallattaken die Adressen ausspähen lassen, kann KASLR umgangen werden. Im Juni 2017 wurden deshalb die ersten Patches unter dem Namen KAISER vorgeschlagen, die den Speicher des Kernels viel weitreichender vor Anwenderprogrammen verstecken, als dies

Spectre and Meltdown

Spectre and Meltdown are threatening our modern IT infrastructure. These vulnerabilities can be found on the hardware level in almost any modern device and undermine the separation of information between user, program and device. Variant 2 of Spectre is especially difficult to patch via software. This article summarizes in short the current state of knowledge about the vulnerabilities and several possibilities to mitigate them.

bisher der Fall war. Dies verhindert nebenbei auch die Meltdown-Attacke. Alle großen Betriebssysteme erhielten daher in den letzten Wochen Patches, die dem KAISER-Ansatz folgen.

Es ist zu empfehlen, die Patches gegen Meltdown (und Spectre Variante 1) in jedem Fall einzuspielen, da Meltdown vergleichsweise leicht ausgenutzt werden kann und mit wenig Aufwand der vollständige Speicher eines Rechners (inkl. aller virtuellen Maschinen im Falle von Cloud Servern) ausgelesen werden kann. Im Falle eines Cloud Servers besteht sogar die Gefahr, dass eine virtuelle Maschine den Speicher anderer Maschinen auf demselben Host ausliest.

SPECTRE VARIANTE 2

Im Folgenden geht es um die Probleme und Ansätze bezüglich Spectre Variante 2. Diese Lücke ist nicht so einfach zu beheben und kann deutliche Leistungsprobleme nach sich ziehen.

Ansatz 1: Neue CPU-Funktionen (IBRS/IBPB) müssen durch ein Microcode-Update nachgerüstet werden. Die „Indirect Branch Restricted Speculation“ (IBRS) muss gesetzt werden, wenn die CPU in den privilegierten Modus schaltet, und sorgt dafür, dass Zweige aus dem weniger privilegierten Modus vergessen werden. An dieser Stelle bedeutet das praktisch, dass die spekulative Ausführung deaktiviert ist. Diese Methode reicht aber nicht vollständig aus, weil die CPU nicht zwischen verschiedenen virtuellen Instanzen unterscheiden kann. Daher braucht man eine weitere Barriere, die „Indirect Branch Prediction Barriers“ (IBPB). Sie sorgt dafür, dass auch beim Kontextwechsel bekannte Zweige vergessen werden. Gleichzeitig müssen diese Funktionen vom Kernel und Hypervisor einer Virtualisierungsschicht verwendet bzw. gesteuert werden.

Ansatz 2: Eine neue Compiler-Funktion „Retpoline“ (return trampoline) sorgt auf Compiler-Ebene dafür, dass der erzeugte Code keine angreifbare spekulative Ausführung mehr beinhaltet. Bei diesem Ansatz muss der Kernel mit Retpoline kompiliert sein, und *jede Software*, die zu schützende Geheimnisse enthält, muss ebenso mit Retpoline neu übersetzt werden.

Hier muss man Vor- und Nachteile abwägen. Insbesondere Ansatz 1 führt nach unseren Messungen im HPC-Bereich zu erheblichen, inakzeptablen Leistungseinbußen und funktioniert nur auf CPUs, deren Microcode-Update freigegeben wurde. Ansatz 2 beinhaltet das Neuübersetzen zumindest aller „systemkritischen“ Programme, führt aber zu weniger oder keinem [4] Leistungseinbruch. Die verschiedenen Distributoren setzen zur Zeit auf unterschiedliche Ansätze zur Behebung.

Unserer Einschätzung nach sollte man bei dieser Lücke nicht zu voreilig verfahren. Jedes System muss auf die Art der Verwendung hin geprüft werden und ob dabei tatsächlich eine wahrscheinliche Gefahr des Speicherauslesens besteht. Bis auf die durch Browser-Updates bereits verhinderten Angriffe via Javascript gibt es bislang nur einfache, stark beschränkte Proof-of-Concept-Angriffe auf Spectre Variante 1 und 2, sodass bislang noch kein echter Schädling für diesen Angriff bekannt ist. Für Systeme mit mehreren Nutzern ist das Absichern langfristig sicherlich nötig. Bis dahin wird sich aber ein favorisierter Weg aller Distributoren durchsetzen, der vermutlich durch Retpoline kritische Software wie Kernel, PAM, LDAP, Browser oder ähnliches speziell schützt. Zumindest die Entwicklung des offiziellen Linux-Kernels deutet auf diesen Weg hin. Bei Microsoft scheint sich Ansatz 1 durchzusetzen.

PERFORMANCE-MESSUNGEN AUF DEM HPC-SYSTEM DER GWDG

Zur Einschätzung der zu erwartenden Leistungseinbußen wurden die Programme „Berlin Quantum ChromoDynamics“ (BQCD) in der Benchmarkversion [5] und „Interleaved-or-Random“ [6] (IOR) genutzt. BQCD ist ein Hybrid-Monte-Carlo-Programm, das in mehreren Iterationen Berechnungen durchführt und Zwischenschritte im Filesystem speichert. IOR ist ein weit verbreitetes I/O-Benchmarkprogramm, das zum Testen der Filesystemleistung (I/O-Leistung) genutzt wird. Beide Programme liefen auf einem dedizierten Knoten in drei Versionen: 1. ungepatcht und somit ungeschützt gegen Meltdown und Spectre in allen Versionen, 2. mit dem für RHEL verfügbaren Kernel-Update zur Behebung von Meltdown (KPTI) und Spectre Variante 1, und 3. mit dem Kernelupdate und einem Microcode-Update, wie in Ansatz 1 beschrieben. Die Abbildungen 1 und 2 zeigen die Ergebnisse der IOR-Benchmarks auf verschiedenen Speicherebenen.

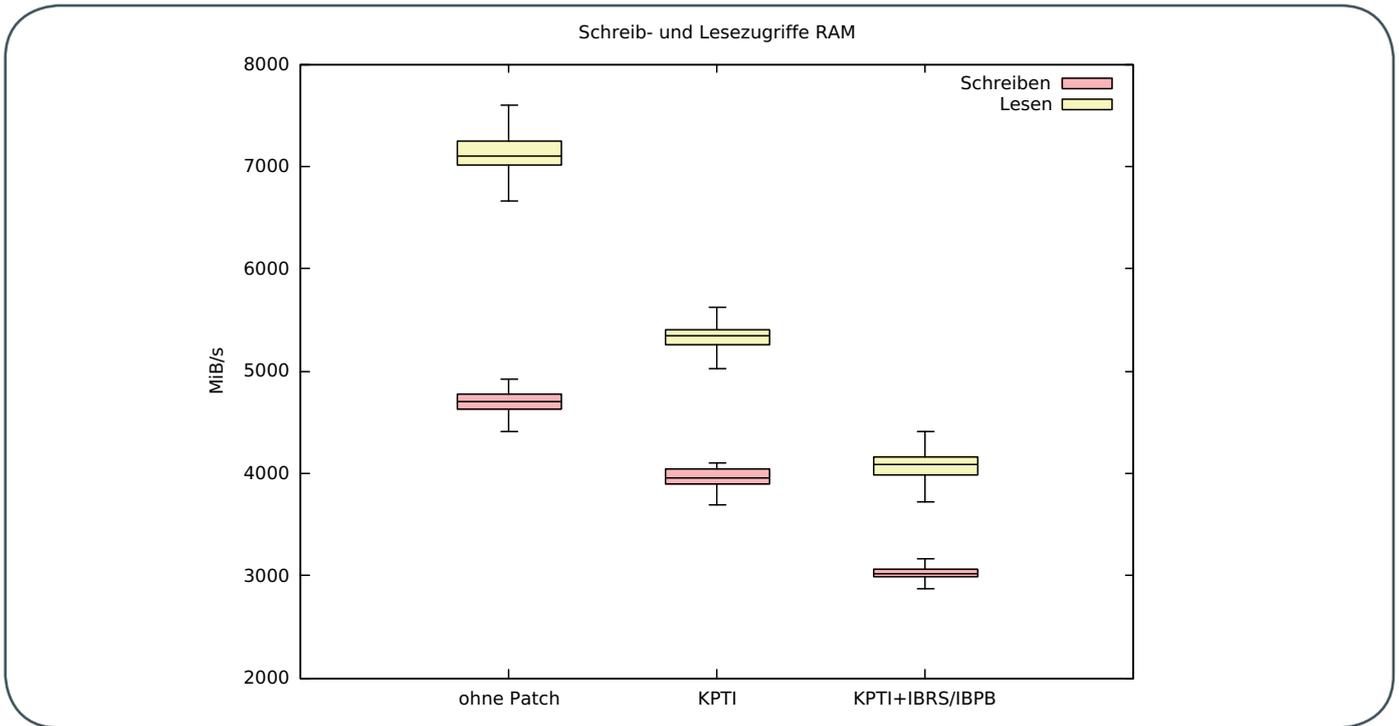
In der RAM-Disk sind die Performance-Einbußen sowohl beim Schreiben, als auch beim Lesen deutlich zu erkennen (siehe Abb. 1). Bereits das Kernel-Update mit aktivem KPTI reduziert den Schreib- und Lesedurchsatz erheblich. Durch das Hinzuschalten von IBRS und IBPB wird er sogar noch weiter reduziert. Anders sieht dies allerdings beim Schreiben auf eine lokale SSD (hier: */local*) oder auf ein paralleles, über Ethernet angebundenes Filesystem (hier: *\$HOME*) aus. Wie in Abb. 2 zu sehen ist, sind die Performance-Unterschiede sowohl auf dem lokalen Speicher als auch auf dem parallelen Filesystem bei Schreibzugriffen vernachlässigbar. Bei Lesezugriffen zeigte sich zwar ein ähnliches Bild wie im RAM: Der Durchsatz sinkt auf der lokalen SSD ebenso wie beim Lesen aus dem Home-Verzeichnis. An der Lesegeschwindigkeit (zwischen 8.000 MiB/s bis 4.000 MiB/s in den Spitzenmessungen) wurde allerdings schnell klar, dass es einen Cache-Effekt geben musste, der dazu führt, dass große Teile aus dem Cache gelesen werden und sich daher der Flaschenhals zurück zum Kernel und dessen I/O-Performance verschiebt. Daher beschränken wir uns hier bei der Darstellung auf die Schreib-Performance.

Die Messungen der Laufzeiten von BQCD – einem Benchmark, der eine typische HPC-Anwendung simuliert, – zeigen, dass sich die Performance-Einbrüche stark in Grenzen halten (siehe Tab. 1). Der Flaschenhals einer solchen Anwendung liegt eher nicht im Cache-I/O:

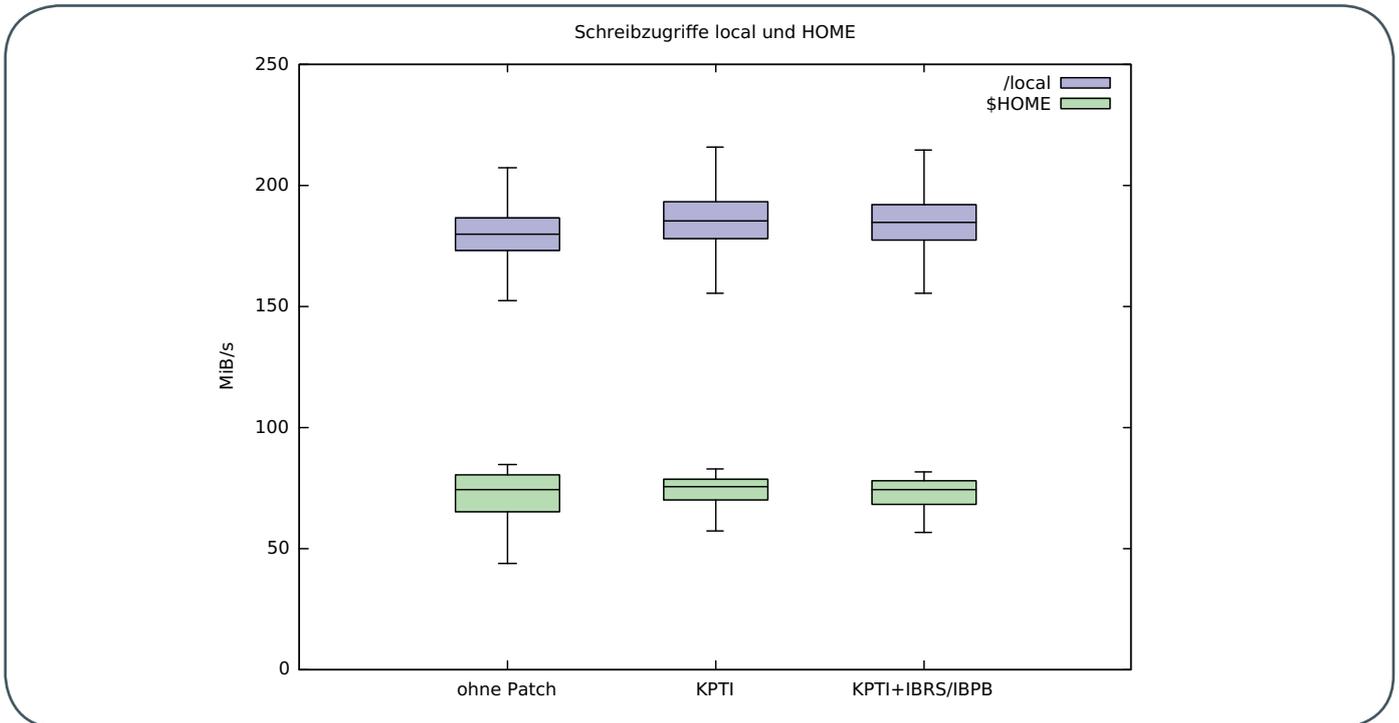
BQCD			
Kernel	o. Patch	KPTI	KPTI+IBRS/IBPB
Zeiten (s)	7,1885	7,1962	7,2872

Tabelle 1: Laufzeiten von BQCD

BQCD zeigt bei Läufen auf der lokalen SSD und dem Kernel-Update mit Patches gegen Spectre 1 und Meltdown eine Laufzeiteinbuße von durchschnittlich unter 1 % der Laufzeit. Bei zusätzlich aktiviertem IBRS/IBPB ist die Einbuße noch immer weit unter 2 %. Wir erwarten also, dass ein Kernel-Update mit aktivem KPTI weder bei I/O-gebundenen Anwendungen, noch bei anderen Anwendungen zu erheblichen Laufzeiteinbußen führt. Gleiches gilt im Prinzip auch für IBRS/IBPB, da der Flaschenhals hier vermutlich noch immer eher woanders liegt.



1_Von IOR gemessene Bandbreiten bei Speicherzugriffen auf `/dev/shm` (ein temporäres Filesystem im RAM) ohne Patch, mit dem Patch gegen Meltdown (KPTI) und Spectre 1 und mit zusätzlich aktiviertem Schutz gegen Spectre 2 (KPTI+IBRS/IBPB). Verteilung von jeweils 3.000 Läufen und kleinen Blockgrößen. Schreibzugriffe sind in rot, Lesezugriffe in gelb dargestellt.



2_Von IOR gemessene Bandbreiten bei Speicherzugriffen auf `/local` (lokale SSD, in blau) und auf `$HOME` (in grün) ohne Patch, mit dem Patch gegen Meltdown (KPTI) und Spectre 1 und mit zusätzlich aktiviertem Schutz gegen Spectre 2 (KPTI+IBRS/IBPB). Verteilung von 3.000 Läufen auf `/local` und 300 Läufen auf `$HOME` mit jeweils kleinen Blockgrößen.

FAZIT

Dieser Artikel gibt einen Überblick über die aktuelle Informationslage zu Spectre und Meltdown. Während Spectre Variante 1 und Meltdown relativ leicht durch ein Kernel-Update zu beheben sind, ist bei Spectre Variante 2 noch Vorsicht bei den vorhandenen Lösungen geboten. Entweder rüstet man CPU-Funktionen mit

einem Microcode-Update nach, was bisher nur bei einer sehr ausgewählten Menge an CPUs möglich ist, und lebt mit den I/O-Einbußen, oder man kompiliert die gesamte Software neu. Zur Zeit gehen die Linux-Distributoren unterschiedliche Wege und man hat bei gesetzter Distribution keine andere Wahl, als den jeweiligen Ansatz mitzugehen. Bei RHEL sind das zur Zeit IBRS und IBPB.

Trotz der Performance-Einbußen bei den Lese- und

Schreibgeschwindigkeiten ist ein Update zur Absicherung gegen Meltdown wegen der relativ einfachen Ausnutzbarkeit unumgänglich. Zur Zeit kann man sich bei den Spectre-Lücken noch Zeit lassen und die Updates der Distribution mit IBRS und IBPB deaktivieren, da es keine lauffähigen Exploits gibt. Das macht die Performance-Probleme nach dem Update etwas erträglicher. Auf lange Sicht wird man aber zumindest auf Mehrbenutzersystemen Updates einspielen müssen, bei denen dann hoffentlich das Gros der Distributoren denselben konzeptuellen Pfad eingeschlagen hat.

LINKS

- [1] <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5753>
- [2] <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5715>
- [3] <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5754>
- [4] <https://www.theinquirer.net/inquirer/news/3024392/google-claims-its-spectre-patch-results-in-no-degradation-to-system-performance>
- [5] <https://www.rrz.uni-hamburg.de/services/hpc/bqcd.html>
- [6] <https://github.com/LLNL/ior>



Mailinglisten

Mailversand leicht gemacht!

Ihre Anforderung

Sie möchten per E-Mail zu oder mit einer Gruppe ausgewählter Empfänger kommunizieren, auch außerhalb Ihres Instituts. Sie möchten selbstständig eine Mailingliste verwalten, z. B. Empfänger hinzufügen oder entfernen. Bei Bedarf sollen sich auch einzelne Personen in diese Mailingliste einschreiben dürfen.

Unser Angebot

Wir bieten Ihnen einen Listserver, der zuverlässig dafür sorgt, dass Ihre E-Mails an alle in die Mailingliste eingetragenen Mitglieder versendet werden. Die E-Mails werden automatisch archiviert. Das Archiv kann von allen Mitgliedern der Liste nach Schlagwörtern durchsucht werden. Die Anzahl Ihrer Mailinglisten ist unbegrenzt.

Ihre Vorteile

- > Leistungsfähiges ausfallsicheres System zum Versenden von vielen E-Mails
- > Sie senden Ihre E-Mail lediglich an eine Mailinglisten-Adresse, die Verteilung an die Mitglieder der Mailingliste übernimmt der Listserver.

- > Listenmitglieder können an diese E-Mail-Adresse antworten. Eine Moderationsfunktionalität ist verfügbar, mit der Sie die Verteilung einer E-Mail genehmigen können.
- > Voller administrativer Zugriff auf die Einstellungen der Mailingliste und der Listenmitglieder
- > Obsolete E-Mail-Adressen werden vom System erkannt und automatisch entfernt.
- > Wenn Ihre E-Mail-Domäne bei uns gehostet wird, können Sie auch die Adresse der Mailingliste über diese Domäne einrichten lassen.

Interessiert?

Für die Einrichtung einer Mailingliste gibt es zwei Möglichkeiten: Zum einen als registrierter Benutzer der GWDG im Selfservice über das Kundenportal der GWDG und zum anderen, indem Sie bitte eine entsprechende E-Mail an support@gwdg.de senden, die die Wunsch-E-Mail-Adresse der Liste sowie die E-Mail-Adresse der Person, die die Liste bei Ihnen administrieren soll, enthalten sollte. Die administrativen Aufgaben sind leicht zu erlernen.

>> www.gwdg.de/maillinglisten

Steigende Nutzung der Plagiatspräventionsdienste der GWDG

Text und Kontakt:

Roland Groh
roland.groh@gwdg.de
0551 201-1838

Seit 2014 bietet die GWDG den Mitarbeitern der Universität Göttingen die Möglichkeit an, kostenfrei Dienste zur Plagiatsprävention zu nutzen. Auch 2017 konnte erneut ein starkes Wachstum bei der Nutzung dieser Dienste verzeichnet werden. In diesem Artikel werden einige Informationen zur Nutzungsentwicklung bei den beiden zur Verfügung stehenden Produkten Turnitin und iThenticate gegeben.

NUTZUNGSENTWICKLUNG

Um sicherzustellen, dass bei der Erstellung von studentischen Arbeiten wie Haus-, Bachelor- oder Masterarbeiten wie auch bei Doktorarbeiten fremde „Schöpfungen“ nicht als eigenes Werk deklariert wurden, kann zur Überprüfung des korrekten Zitierens Plagiatserkennungssoftware eingesetzt werden. Die GWDG bietet den Mitarbeitern der Universität Göttingen seit 2014 die Produkte **Turnitin** und **iThenticate** an, um zu helfen, Plagiatsfälle zu erkennen.

Im Jahr 2013 konnten mit einem Workshop und einer anschließenden Testphase bei der GWDG erste Erfahrungen mit der Plagiatserkennungssoftware gewonnen werden. Zu diesem Zeitpunkt konnte niemand ahnen, welcher durchschlagende Erfolg und welche große Nachfrage sich daraus später entwickeln würde (siehe Abb. 1). Damals nahmen nur 35 Personen an der Testphase teil. Doch schon 2014, im ersten Jahr des Regelbetriebs, konnten bis zum Jahresende mehr als 100 Nutzer gezählt werden. Während 2014 die Zahl der Nutzer 129 erreichte, stieg sie in 2015 auf 359, um dann Ende 2016 bereits 604 zu erreichen. Im Jahr 2017 konnten dann 903 Nutzer verzeichnet werden. Am 31.01.2018 wurde dann der 1.000ste Nutzer erreicht. Aktuell (Stand: 22.02.2018) haben wir 1.027 Nutzer und rechnen damit, bis Jahresende ca.

1.250 Nutzer zu erreichen.

Bei der Untergliederung der Nutzer nach ihren Fakultäten steht nach wie vor die Medizinische Fakultät an erster Stelle. Im Jahr 2017 kamen 214 Nutzer von der Medizinischen Fakultät. Als zweites folgte die Wirtschaftswissenschaftliche Fakultät mit 128 Nutzern. Die Fakultät für Agrarwissenschaften verzeichnete 95 Nutzer und die Fakultät für Biologie und Psychologie erreichte 67.

Plagiarism Prevention Services

In order to prevent plagiarism in students' papers and scientific publications, the GWDG offers a plagiarism prevention service. After an exploration period with 35 participants in 2013, the service started in 2014. Currently (22.02.2018), 1,027 registered users are registered for our service.

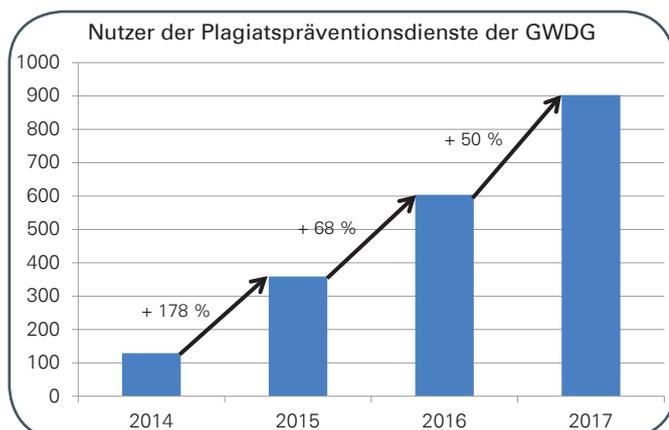
Counting by faculty, in 2017 most users (214) came from the Medicine Faculty, followed by the Faculty of Economic Sciences (128 users). 95 users belonged to the Faculty of Agricultural Sciences and 67 to the Faculty of Biology and Psychology.

As to the papers checked, there is also a continuous rise. In the first year 2014 their number was 1,107, and it was more than doubled in 2015 (2,353 documents). In 2016, a number of 4,142 papers was reached. In 2017 the count surpassed the 6,000 with 6,556 papers.

If you have any questions regarding plagiarism feel free to contact the plagiarism advisor Roland Groh or any other member of the plagiarism team (e-mail: gwdg-plagi@gwdg.de; Roland Groh, phone: 0551 201-1838; Regina Bost, phone: 0551 201-1831; Anke Bruns, phone: 0551 201-1519; Simon Heider, phone: 0551 201-1840).

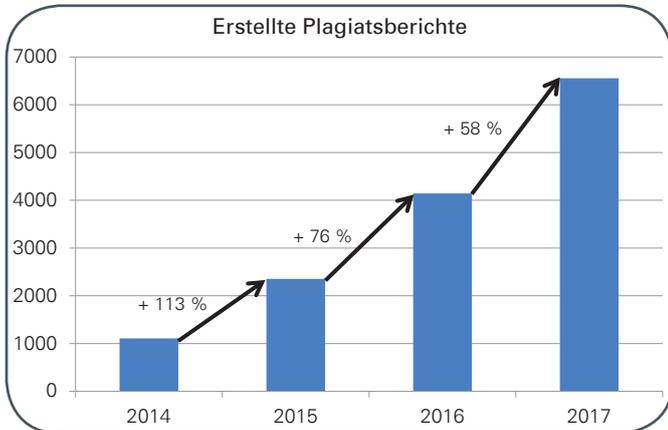
For further information, please look at https://info.gwdg.de/docs/doku.php?id=en:services:application_services:plagiarism_prevention:start.

To use our service you can fill in the form at <https://plagiat.gwdg.de>.



1_Nutzerentwicklung 2014 – 2017

Betrachtet man die Zahl der Dokumente, welche mit unseren Plagiatspräventionsdiensten geprüft wurden, so ist auch hier ein deutliches Wachstum zu verzeichnen (siehe Abb. 2). Wurde 2014 mit 1.107 Dokumenten bereits eine große Zahl an Dokumenten geprüft, so verdoppelte sich diese Zahl in 2015 auf 2.353 Dokumente. 2016 gab es ebenfalls eine große Steigerung: Mit 4.142 Dokumenten wurden erneut mehrere Tausend Dokumente



2_Berichtsentwicklung 2014 – 2017

geprüft. In 2017 wurden dann 6.556 Plagiatsberichte generiert, was einem Wachstum von 58 % gegenüber dem Vorjahr entspricht. 3.585 Berichte wurden mit Turnitin und 2.971 Berichte mit iThenticate erstellt.

ANSPRECHPARTNER

Mit dem Start der Plagiatspräventionsdienste in 2014 wurde von der Universität Göttingen eine bei der GWDG angesiedelte „Kompetenzstelle Plagiatsvermeidung“ geschaffen, die von Herrn Roland Groh besetzt wird. Zu allen Fragen rund um das Thema Plagiatsvermeidung stehen Ihnen bei der GWDG Herr Groh und das Plagiatspräventions-Team (E-Mail: gwdg-plagi@gwdg.de) gern zur Verfügung. **Zum Team gehören:**

- Roland Groh (Tel.: 0551 201-1838)
- Regina Bost (Tel.: 0551 201-1831)
- Anke Bruns (Tel.: 0551 201-1519)
- Simon Heider (Tel.: 0551 201-1840)

Weitere Informationen finden Sie unter https://info.gwdg.de/docs/doku.php?id=de:services:application_services:plagiarism_prevention:start. Ein Formular zur Beantragung der Dienste ist unter <https://plagiat.gwdg.de> zu finden. ■

Kurz & knapp

Öffnungszeiten des Rechenzentrums um Ostern

Das Rechenzentrum der GWDG ist vom 30.03.2018, Karfreitag, bis zum 02.04.2018, Ostermontag, geschlossen.

Falls Sie sich zu der Zeit, in der das Rechenzentrum geschlossen ist, in dringenden Fällen an die GWDG wenden wollen, schicken Sie bitte eine E-Mail an support@gwdg.de. Das dahinter befindliche Ticket-System wird auch während dieser Zeit von Mitarbeiterinnen und Mitarbeitern der GWDG regelmäßig überprüft.

Wir bitten alle Benutzerinnen und Benutzer, sich darauf einzustellen.

Grieger

Internationales Symposium „The critical role of university Research Data Management infrastructure in transforming data to knowledge“

Datenmanager und Bibliothekare tauschen sich an der Niedersächsischen Staats- und Universitätsbibliothek Göttingen über die Entwicklung von Forschungsdaten-Services aus

Um von den Erfahrungen anderer Institutionen bei der Etablierung von Forschungsdaten-Management-Infrastrukturen zu lernen und eigene Kenntnisse in diesem Bereich weiterzugeben, veranstaltet die Göttingen eResearch Alliance, eine Kooperation der GWDG und der Niedersächsischen Staats- und Universitätsbibliothek Göttingen (SUB), vom 18. bis 20. März 2018

ein international renommiert besetztes Symposium zum Thema „The critical role of university Research Data Management infrastructure in transforming data to knowledge“. Das Symposium wird gemeinsam mit dem Committee on Data for Science and Technology (CODATA) veranstaltet, einer Organisation des Internationalen Wissenschaftsrates.

Wie auch andere Universitäten und Bibliotheken weltweit, bietet die SUB Forschern umfangreiche Services für die Arbeit mit Forschungsdaten. Von der Unterstützung beim Schreiben von Anträgen und Datenmanagement-Plänen, über die Bereitstellung von Repositorien zur Publikation von Daten und deren Speicherung, bis hin zur Vergabe von dauerhaften Identifikatoren für Autoren und Werke wird ein breites Spektrum des digitalen Forschungsprozesses bereits durch Services abgedeckt.

Obgleich weltweit alle Universitäten vor ähnlichen Herausforderungen im Forschungsdaten-Management stehen, gibt es kaum Plattformen für einen organisierten Austausch über diese Themen. Diese Lücke wird nun durch das geplante Symposium geschlossen. Vom 18. bis 20. März 2018 werden Fachleute aus allen Teilen der Welt in über 50 Vorträgen und Keynotes Erfahrungen austauschen und existierende Lösungen präsentieren und so gegenseitig voneinander lernen, unter anderem werden aus den USA Vertreter aus Stanford, MIT und Harvard anwesend sein. Die kostenlose Veranstaltung ist als Vorkonferenz für eines der wichtigsten internationalen Treffen zum Thema Forschungsdaten, dem 11. Research Data Alliance Plenary Meeting, das in diesem Jahr in Berlin stattfindet, geplant.

Programm und Registrierung: <http://www.eresearch.uni-goettingen.de/content/pre-rda-symposium>

Brase (SUB)

Thread Scheduling on Multi-core Systems

Text and Contact:

Dr. Konrad Heuer
konrad.heuer@gwdg.de
0551 201-1540

Probably many computer users and system administrators feel more relaxed when the computer hardware they buy contains as many processor cores as possible. Regardless of the operating system used they know that the number of cores, in addition to other factors, has a positive effect on the performance of the hardware. But how do operating systems master the challenge of using many processor cores? This is covered by the following article, with a slight emphasis on FreeBSD, but also with some eyes beyond the box towards Linux, macOS and Windows. A lot of unfortunately inevitable technical terms including the term "scheduling" will be explained in the running text.

FIRST TECHNICAL TERMS

At the beginning of this article it may be useful to look at the challenges that operating systems need to master on server systems, but also on workstations in a weakened form. In this context some explanations on basic technical terms will be presented, too. Heart of every operating system is the **kernel** that gets loaded into memory on system boot. The kernel stays resident in memory and is responsible for managing computer resources and data flow between all running programs and all devices including the network.

Each modern operating system provides tools to view the current state of the system. Figure 1 shows a snapshot of the screen output of the useful tool *sysstat* which is available on FreeBSD. The content is probably not easy to understand on the first look, but in the next paragraphs some values will be explained which are particularly interesting in the current context and thus marked in Figure 1.

The right side of Figure 1 shows a column titled as **interrupts**. **Interrupt requests** are sent to the operating system kernel by devices which need attention. In the example output, the first and main network card called *bce0* sends **6274** interrupt requests per second to the kernel because it receives data from the network which must (preferably) immediately be processed by the kernel to avoid package losses in network communication. In total over all devices, the kernel has to deal with **13698** interrupt requests per second. Of course, these numbers fluctuate from second to second, but will be named exactly here for better traceability.

Starting from the left, and approximately vertically centered there is a row summing up CPU usage. Since **79.3%** of the 100% CPU capacity are idle and only **0.3%** are used for interrupt processing, the system is obviously not overloaded. **11.3%** of CPU capacity is used for user mode processing, and **9.1%** for system mode processing. Programs run in **user mode** as long as they do calculations as number crunching or string processing. But to do data input or output (I/O) operations of any kind like e.g. reading data from disk, keyboard or network connections or writing data to

disk or to network connections, programs need to enter the kernel by so-called **system calls**, and processing will continue in **system mode** until the I/O operations are completed and the programs exit from the kernel. A **system call** (abbreviated as **Sys** in Figure 1) is a specially reserved interrupt request, and as can be seen, more than 1 million system calls (**1M**) per second happen currently on the example system. **5** processes are runnable, and **488** are sleeping.

THREAD MODEL

A **process** is a program in execution with all its environment, particularly all the dynamic data. A process can have multiple user space **threads** of execution, but has at least one user space thread of execution. For example, a web browser could create a user space thread for each tab that gets opened. All user space threads share the virtual address space of the process they belong to, and each user space thread has a corresponding **kernel thread**. This last statement is true for nearly all relevant operating systems,

Thread-Scheduling auf Multicore-Systemen

Viele Computerbenutzer und Systemadministratoren sind vermutlich gelassener, wenn die neu beschaffte Computerhardware so viele Prozessorkerne wie möglich enthält. Unabhängig vom verwendeten Betriebssystem ist bekannt, dass sich die Anzahl der Kerne neben anderen Faktoren positiv auf die Leistungsfähigkeit der Hardware auswirkt. Aber wie meistens Betriebssysteme die Herausforderung, viele Prozessorkerne zu verwenden? Dies wird im folgenden Artikel behandelt, mit einem kleinen Schwerpunkt auf FreeBSD, aber auch mit einigen Blicken in Richtung Linux, macOS und Windows. Die leider unvermeidlichen Fachbegriffe einschließlich des Begriffs „Scheduling“ werden im laufenden Text erklärt.

```

4 users      Load  5.55  4.91  4.23                Jan 18 14:05
Mem:KB      REAL          VIRTUAL          VN PAGER    SWAP PAGER
           Tot  Share  Tot  Share  Free          in  out      in  out
Act 2720636 57704 10206656 105220 27184k count  2   4
All 2757492 87828 11230612 611184  pages  15  6
Proc:
r  p  d  s  w  Csw  Trp  Sys  Int  Sof  Flt  2  ioflt  13698  total
5                               24k 8093 1M 6623 40 4005 1937 cow      atkbd0 1
                               488                               1600 zfod      ohci0 ohci
                                                             ozfod      ehci0 17
                               %ozfod      ohci2 ohci
                               daefr      1 ehci1 19
                               1902 prcfr      ahci0 22
Namei      Name-cache  Dir-cache  621821 desvn  4485 totfr  52  cpu0:timer
      Calls  hits  %  hits  %  233690 numvn  2  react  6274  bce0 259
178998 171771 96  1  0  155456 frevn  382 pdpqs  69  bce2 261
                               2  intrn  22  bce3 262
Disks mfid0 mfid1  da0  da1  da2  da3  cd0  2377824 wire  10  mfi0 263
KB/t  14.11 16.38 15.17  0.00 13.72  0.00  0.00 897432 act  5  isp0 264
tps    6    4    5    0    5    0    0  2178668  inact 5  isp1 265
MB/s   0.09 0.07 0.07  0.00 0.06  0.00  0.00 380908 cache 42  cpu3:timer
%busy  4    0    1    0    1    0    0  26803712 free 296  cpu2:timer

```

Figure 1: Snapshot of the output of `sysstat -vmstat` on a medium-loaded GWDG FreeBSD server

but definitely for (in alphabetical order) FreeBSD, Linux, macOS and Windows. They all use the so-called 1:1 thread model with a 1:1 correspondence of user space and kernel threads. In this approach, threads are individually scheduled by the kernel. Thus to be honest, there are 5 runnable threads and 488 sleeping threads on the example system of Figure 1. Task of the scheduler is to constantly assign CPU time in small portions to all the threads on the system which are ready for execution to give each of them the illusion of running exclusively on the hardware. A context switch happens when a thread gives up CPU usage either voluntary or involuntary. The scheduler saves the run-time environment of such a thread in the necessary extent and gives CPU capacity to another thread by loading its environment and starting it. In Figure 1, there are more than 24.000 (24k) context switches per second.

As told above, all threads of a single process share a virtual address space. This virtual address space gives each process an illusion of an exclusive access to main memory and therefore protects the true physical memory locations against access from other processes. As a consequence, processes are isolated from each other. This is of course implicitly true for threads of different processes, too. The virtual address space is divided in so-called pages which have commonly a size of 4 kByte. Pages are loaded by the kernel from disk into real memory on demand and may later get written to the so-called swap space on disk when main memory is getting scarce due to high system load.

In addition to kernel threads belonging to user space threads there are some special kernel threads fulfilling operating system tasks. They never exit from system mode and are exclusively executed with kernel privileges. Examples for these threads are the threads servicing interrupt requests.

BACK AT THE GOOD OLD DAYS

Before getting started with today's complex multi-core scheduling a quick look back at the "good old days" of computers with

only single CPUs makes some sense. In those times, there were no multi-threaded applications, and each process had only a single thread of execution. The term thread was not used at that time. State of the art was a process scheduling algorithm called "multilevel feedback queues" as implemented for example by the 4.3BSD operating system which was back then the basis of many commercial UNIX-like operating systems like DEC ULTRIX, SunOS or OSF/1.

A typical szenario at that time was that department employees had terminals at their workplaces connected to a central computer via serial ports. Time-sharing operating systems like UNIX or VMS made small parts of the CPU time and other computer resources available for each individual employee. Ideally, users had the illusion of using their own computers. At least one process (the command-line interface) was running for each user, but usually the number of processes was slightly higher. As already briefly mentioned above, the process scheduler was responsible for distributing the CPU time fairly across all runnable processes and for ensuring short response times of highly-interactive programs like e.g. text editors.

Looking at the 4.3BSD scheduler, a process priority in the range from 0 (highest priority) to 127 (lowest priority) was assigned to each process. By dividing the priority by 4, each runnable process could be placed on one of 32 run queues, and whenever the CPU became available, the first process from the first non-empty queue was run.

A running process was interrupted at the latest after a maximal time slice of 100 ms had expired. Compute-intensive processes which took considerable advantage of their time slices got their priority quickly lowered, whereas processes which got blocked after a very short run time due to some I/O operation (for example text editors waiting for further keyboard input) were allowed to keep their high initial priority. A digital decay filter was periodically applied to all processes so that temporarily compute-intensive processes were not punished forever.

For reasons of space and because it is only historically interesting no detailed formulas are presented here. The important message is that this 4.3BSD scheduling algorithm guaranteed exemplary behavior which is also sought after with the many modern schedulers.

Particularly, it is important to keep in mind that a process (or today, a thread) which gets blocked because it has to wait for an I/O operation, an event or a lock sets the CPU (or today, a CPU core) free allowing a voluntary context switch whereas a compute-bound process gets involuntarily suspended by the scheduler after expiration of its time slice.

Furthermore, it is very important to be aware of the fact that anytime on a single CPU system only one process could run. As a consequence, not more than one process could enter the kernel by a system call. There was no need for a reentrant kernel design. It just had to be taken care of possible interferences between kernel code for servicing interrupt requests and kernel code for servicing system calls, since interrupt request handling could not be delayed by processing system calls.

THE BIG GIANT LOCK

When the first small but powerful computer systems like PCs or workstations with more than one CPU became available, kernel design had to change on a painful path. FreeBSD and Linux, the most popular UNIX-like operating systems today with a big lead for Linux, helped first with the so-called big giant lock. Processes could be spread over 2 or 4 CPUs, but whenever one process entered the kernel by a system call and locked the kernel, processes on other CPUs had to wait in case of I/O requests until the first process left the kernel.

The big giant lock was gradually replaced by finer-grained locking and eventually removed from the Linux kernel 2.6.39 in 2011. About the same time, it became finally irrelevant with FreeBSD 9.0, whereas Windows NT and its successors were designed differently from the beginning with reentrant-ready kernels.

GOALS OF THE ULE SCHEDULER

With introduction of kernel version 2.6 the previous Linux $O(n)$ scheduler was replaced by an $O(1)$ scheduler which means that time needed for scheduling itself was no longer dependent on the total number n of threads in the system, a very important step on the way to systems running many hundred or even thousands of threads. In kernel version 2.6.23 this scheduler was replaced again by the Completely Fair Scheduler (CFS) which will be described briefly later.

Basic ideas of the $O(1)$ Linux scheduler were picked up by some FreeBSD developers which had set themselves the task to write a new scheduler replacing the traditional 4.3BSD $O(n)$ scheduler mentioned some paragraphs before. An $O(n)$ scheduler is undesirable today since time needed for scheduling itself is wasted time and thus reduces system throughput. Furthermore, traditional schedulers had no idea about multi-core systems and their special requirements.

The new FreeBSD scheduler is called ULE scheduler. ULE is not an acronym but just part of the word schedULER. The ULE scheduler was introduced with FreeBSD 5 in 2003 and continuously further developed for about 10 years until FreeBSD 9. It is

described in the context of the complete kernel in [1]. Design goals of the ULE scheduler are to

- provide interactivity and timesharing performance similar to the traditional 4.3BSD scheduler.
- improve the performance of the scheduling system itself to enhance scalability.
- provide better support for multi-core CPUs.
- guarantee a fair load distribution over several CPUs.
- achieve the best possible CPU affinity on multiprocessor systems.
- handle properly all possible dependencies between threads.
- take care of thrashing.
- introduce back-propagation of priority values from child to parent processes.

The need for the first four points is obvious, at least based on the explanations given above. CPU affinity is important because of cache memory hierarchy. Since computer main memory is slow compared to CPU speed, data fetched by threads in the course of processing get loaded into fast cache memory located on the processor chips themselves. Moving threads unnecessarily between CPU cores and particularly between processor chips by the scheduler would cause partial or complete losses of cache contents and lead to serious performance losses.

Dependencies between threads will be explained below by an example. Thrashing can occur when a system is low on memory. After nearly every context switch virtual memory pages may have to be written to swap space on disk to free real memory for the new threads to run, and probably, for these threads some or many of their virtual memory pages have to be read back from disk to main memory. Thrashing will quickly lead to an unusable system. To prevent from thrashing, FreeBSD will completely swap out the most memory-consuming process for 20 seconds in such a situation, and then continue with swapping out the second most memory-consuming process and so on. At last, back-propagation of priority is useful for a fair scheduling of tasks consisting of chains of threads like compiling and linking large software projects.

EXAMPLE HOW A MODERN KERNEL WORKS

Before going into details of the ULE scheduler, an example will be given to illustrate how a modern kernel works in principle. Nearly every useful program reads data from disk, and traditionally this is done by using the system call `read`. In assembly language, a `read` system call looks approximately like this:

```
read:
    push    dword file_descriptor
    push    dword pointer_to_buffer
    push    dword number_of_bytes_to_read
    mov     eax, 3
    push    eax
    int     80h
```

Arguments of the system call are first pushed on the stack (which is a specific region in virtual memory) by the user mode thread (called thread A for a simple name in the further text), then the number of the system call (3 for `read`) is moved to the `eax` processor register, and at last interrupt number `80h` (hexadecimal

notation, equals 128 in decimal notation) is generated to enter the kernel. Now the corresponding kernel thread (called thread B) takes over control and will try to add the I/O request to the work queue of the disk controller. For this, it has to acquire a mutex (mutex is an abbreviation for mutual exclusion) lock to make sure that no two threads try to modify the work queue simultaneously which would corrupt it.

If the lock cannot be acquired because another thread (issued for example by a different user) already holds the lock, thread B will give up the CPU immediately and will be moved to the so-called turnstile queue. Otherwise, if thread B gets the lock, it will add the read request to the work queue, put aside a so-called wait channel, give up the lock and the CPU and begin to sleep. If the lock was not available initially, thread B will be scheduled for run as soon as the lock becomes available and then work as described.

Typically, some ten milliseconds later data can be read from disk. The disk controller will send an interrupt request, and the kernel thread (called thread C) responsible for handling interrupt requests of this device will be run by the scheduler as soon as possible. Thread C will copy the data to some internal kernel buffer space, identify thread B by its wait channel and wake up thread B which now gets runnable and will be put on a run queue. Then thread C will try to acquire the mutex lock to remove the request from the work queue and to start to process the next request. As soon as thread B gets run by the scheduler, it can copy the data from the internal kernel buffer space to the buffer space whose address was handed over by the system call and which is located within the virtual address space of the user process. Finally, thread A can continue with processing the data as desired by the user.

The complete example is additionally shown in Figure 2 because it is really important to be aware that a modern kernel is able to do a couple of things in parallel. Fine-grained synchronization of threads is often done by mutex locks and sometimes by

spin locks. Threads waiting for a spin lock which cannot be acquired momentarily do not give up the CPU but simply run in a loop until the lock gets available. Spin locks are useful in case of very short times of waiting.

Threads competing for a lock will probably have different priority values. In such a situation priority propagation may happen to speed up the whole process by temporarily handing over to the thread currently holding the lock the highest priority in the pool of the waiting threads. Finally, it should be pointed out that it is important for the scheduler to protect threads against deadlock situations.

THREAD PRIORITIES AND LOW-LEVEL SCHEDULER

It should be comprehensible now that implementing a scheduler like the ULE scheduler is a big challenge. Unlike many years before, a great number of threads instead some ten of user processes compete for CPU resources, and a lot of them are kernel threads handling urgent I/O and interrupt requests.

Different scheduling classes are reserved for threads depending on their function as can be read in Table 1. The FreeBSD kernel is logically divided up into a bottom half for interrupt handling

SCHEDULING CLASS	PRIORITY RANGE	RUN QUEUE INDEX
Bottom-half kernel (interrupt) threads	0 – 47	0 – 11
Real-time threads	48 – 79	12 – 19
Top-half kernel threads	80 – 119	20 – 29
Time-sharing threads	120 – 223	30 – 55
Idle threads	224 – 255	56 – 63

Table 1: Thread priority classes of the ULE scheduler

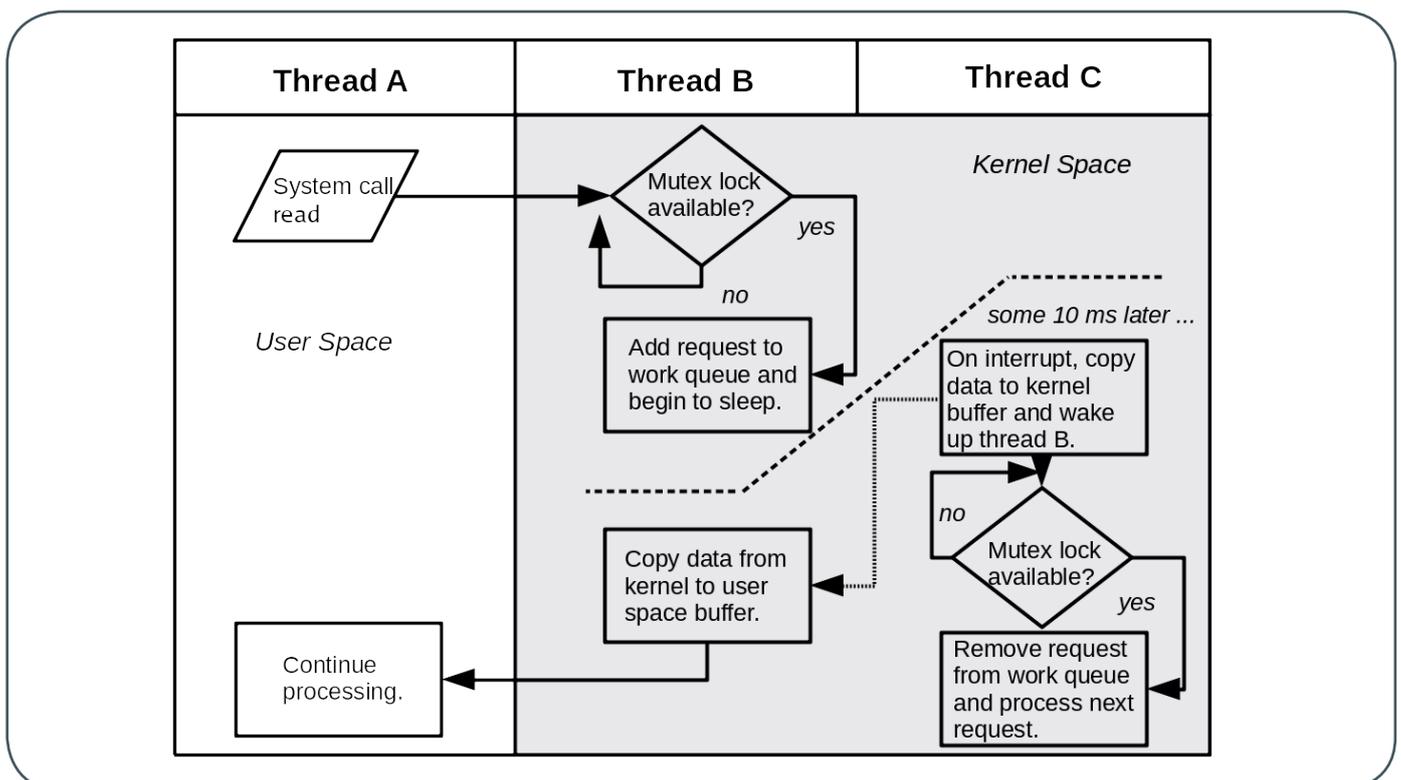


Figure 2: Example of processing a read system call issued by a user process; explanations can be found in the text.

and a top half for servicing system calls. In the example outlined in Figure 2 thread B runs in the top half and thread C in the bottom half of the kernel.

64 run queues are supported, and by dividing the priority value by 4 the run queue index can be computed. The ULE scheduler consists of two parts, a low-level and a high-level scheduler. The low-level scheduler is run very, very frequently and must be absolutely efficient. For each CPU core separate arrays containing altogether 64 run queue addresses are held up, and the low level scheduler will run the first thread found in the first non-empty run queue starting at index 0 and stopping at index 37. As soon as a running thread goes to sleep, is moved to the turnstile queue or has consumed its time slice, the scheduler will look for the next thread to run. If more than one thread is placed on a queue, these threads will be run round-robin.

The run queues 38 to 55 are only taken into account if the first 38 run queues are empty. These run queues are virtually meant for non-interactive threads. The queues are not processed down from index 38 to index 55 because this might lead to starvation of the least-priority threads; instead a pointer is cyclically moved over the run queues to make sure that each run queue gets served from time to time. The remaining run queues 56 to 63 are reserved for idle threads.

THE HIGH-LEVEL SCHEDULER

The high-level part of the ULE scheduler is responsible for deciding on which CPU a thread should be run. Furthermore, the high-level scheduler recalculates about 30 times per second priority values of time-sharing threads. For all other scheduling classes, priority values never change. It is useful, for example, to assign fixed but different priorities to threads handling interrupt requests sent by different devices. Another task of the high-level scheduler is to evaluate the length of the maximum time slice a thread can consume depending on system load. The time slice is hence no longer fixed to 100 ms but can vary between 1 ms and 140 ms.

Recalculation of the priorities of the time-sharing threads is very important to guarantee short response times of interactive threads. Key value in this context is the so-called interactivity score based on the ratio of voluntary (!) sleep time to run time for each thread. In detail, following equations are used:

$$t_s > t_r: \quad s = a \frac{S/2}{t_s/t_r} \quad (1)$$

$$t_r > t_s: \quad s = a \left(\frac{S/2}{t_r/t_s} + S/2 \right) \quad (2)$$

In these equations, s is the interactivity score, S the maximal interactivity score, t_r the run time and t_s the sleep time. As easily can be seen, if the value of a would be 1, in case of equation (1) s will be in the range from 0 to $S/2$, whereas in case of equation (2) s will be in the range from $S/2$ to S . In practice, a is used to scale s to a range from 0 to 100. If s is smaller than the so-called interactivity threshold which is fixed to a value of 30, the thread is assumed to be interactive.

The ULE scheduler sums up voluntary sleep and run times for each thread for a sliding period of about 10 seconds which allows short bursts of heavy CPU usage for an interactive thread while

retaining this preferable status.

For an interactive thread, the priority value will directly be derived from the interactivity score by scaling it to the priority range from 120 to 151 after counting in the nice value (which defaults to 0 on FreeBSD and Linux and is the base priority of a thread). As a result, an interactive thread will be placed in one of the CPU run queues 30 to 37.

The priority of a non-interactive thread is calculated using its run time and nice value. The thread will be placed in one of the batch run queues 38 to 55. Since these run queues are cyclically processed by a moving pointer there is no fixed relation between priority value and run queue index. The better the priority is, the smaller the distance between the current pointer position and the appropriate run queue on which the thread will be placed will be. Figure 3 illustrates the interaction between high-level and low-level scheduler.

MULTIPROCESSOR SCHEDULING

The high-level scheduler has to select a suitable CPU whenever a thread becomes runnable and needs to be placed on a run queue. Doing this, care must be taken to avoid unnecessary shifts of threads between CPUs to prevent from avoidable cache data losses. The whole system is searched for the least-loaded CPU running a lower-priority thread, and after comparing with the situation on the current CPU (the CPU on which the thread has run before), a preferable decision will be calculated. CPU hierarchy will be taken into account, for example, on a system with 2 physical CPUs each of them containing 8 cores it might be reasonable to move to a core inside the same physical CPU, but not to an external one. Furthermore, interrupt threads servicing I/O requests are scheduled on the current CPU if their priority is high enough to run immediately.

On a longer timescale, an idle CPU can take a thread from another CPU if its load exceeds a threshold, and an active CPU can similarly migrate a thread to an idle CPU. The long-term load balancer periodically pushes threads to keep the system balanced.

EXAMPLE

On a lightly loaded test system some aspects of the way of working of the ULE scheduler can be observed. Figure 4 shows the output of the *top* utility when a matrix multiplication is run in parallel by using 4 threads on a system with many cores. The 4 threads are never moved between cores after the first decision has been done on which core they will be started. Priority values of the threads have already decreased to 103.

The loss of priority during run time of these compute-bound threads can be measured by periodically calling the *ps* utility and is plotted in Figure 5. It becomes obvious that in the beginning the loss of priority happens fast, and after almost 40 seconds the worst priority value of 203 is reached. It is worth noting that *ps*, *top* and similar tools subtract the base time-sharing priority value of 120 from their output, thus a base value of 120 has been added to generate Figure 5.

SCHEDULING IN WINDOWS AND MACOS

The macOS operating system is based on the Mach kernel

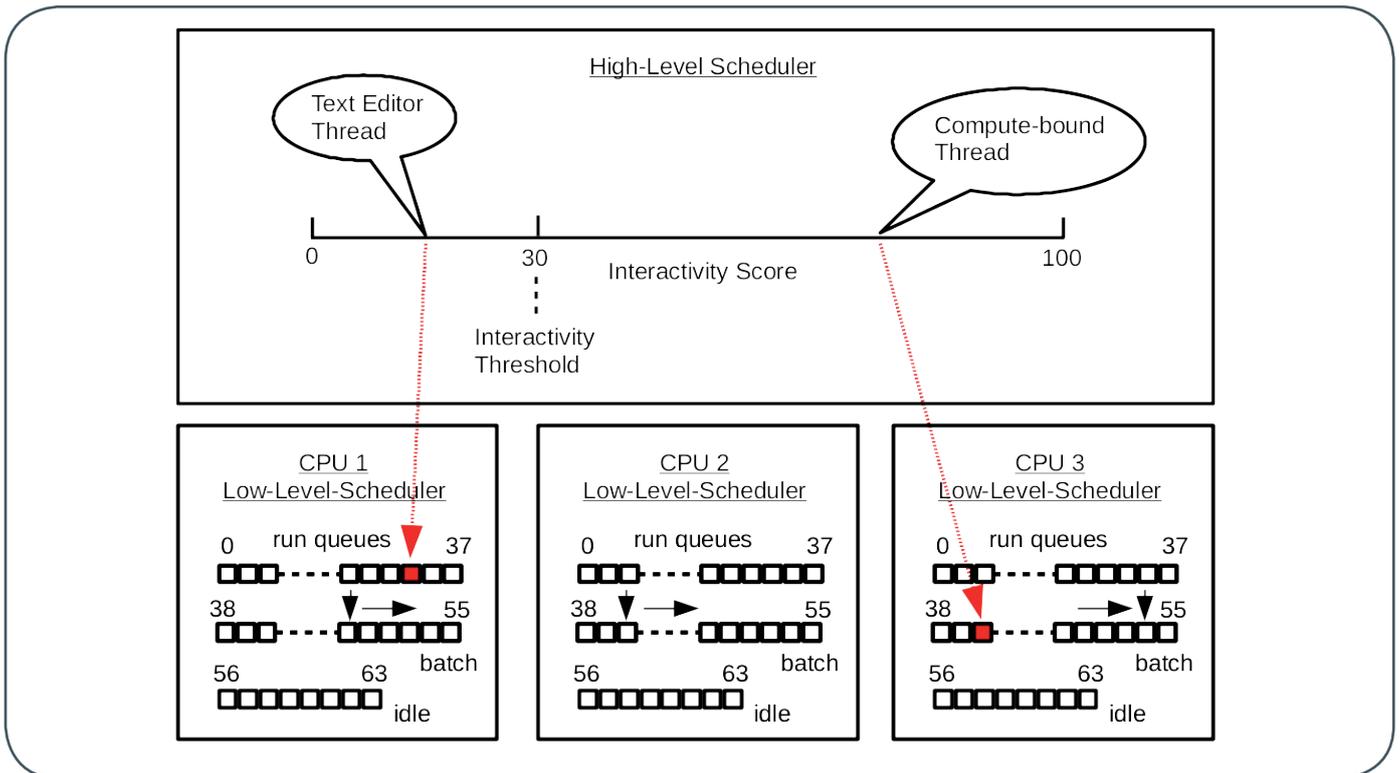


Figure 3: Mode of operation of the ULE scheduler on e.g. 3 CPUs: After calculation of the interactivity score and comparing it to the interactivity threshold threads will be placed on different run queues on different CPUs. Threads from the batch run queues are selected by cyclically moving pointers, and thread priority determines placement on the queues by counting in the distance to the current pointer position.

```
last pid: 50387; load averages:  3.54,  1.92,  0.87  up 11+02:56:33 17:02:58
28 processes:  5 running, 23 sleeping
CPU: 12.5% user,  0.0% nice,  0.0% system,  0.0% interrupt, 87.5% idle
Mem: 1440M Active, 32M Inact, 1581M Wired, 1015M Buf, 28G Free
Swap: 2689M Total, 2689M Free

  PID USERNAME  PRI NICE   SIZE   RES STATE  C  TIME    WCPU COMMAND
50357 kheuer    103    0 1547M 1440M CPU19 19  1:37   99.41% matrix-mul-pt
50357 kheuer    103    0 1547M 1440M CPU27 27  1:37   99.41% matrix-mul-pt
50357 kheuer    103    0 1547M 1440M CPU3   3  1:37   99.41% matrix-mul-pt
50357 kheuer    103    0 1547M 1440M CPU2   2  1:37   99.41% matrix-mul-pt
50385 kheuer     20    0 22208K 4652K CPU15 15  0:00    0.01% top
49900 kheuer     20    0 85228K 7924K select 13  0:00    0.00% sshd
  713 root        20    0 20636K 6252K select 11  0:02    0.00% sendmail
  578 root        20    0 10500K 2448K select 13  0:01    0.00% syslogd
  710 root        20    0 57812K 7052K select 11  0:03    0.00% sshd
  720 root        20    0 12592K 2436K nanslp 14  0:01    0.00% cron
50357 kheuer     39    0 1547M 1440M uwait  22  0:00    0.00% matrix-mul-pt
  716 smmsp      20    0 20636K 5940K pause  26  0:00    0.00% sendmail
  432 root        20    0 9560K  5056K select  8  0:00    0.00% devd
49901 kheuer     20    0 19660K 3680K pause  23  0:00    0.00% tcsh
49909 kheuer     20    0 19660K 3692K pause  26  0:00    0.00% tcsh
49894 root        20    0 85228K 7852K select  6  0:00    0.00% sshd
```

Figure 4: The `top` utility shows a multi-threaded application (matrix multiplication) running with 4 threads. CPU affinity is perfect, the threads are never moved between cores (as can be seen from the column titled `C`).

and on FreeBSD, and documentation can be found in [2]. Windows [3] and macOS use similar scheduling classes to those shown in Table 1. Details of the implementation are different compared to FreeBSD, of course, for example in macOS and Windows a higher priority value implies a higher priority (a little detail). Handling of time-sharing threads is similar, and since all operating systems face the same challenges, comparable strategies concerning for example efficiency, thread dependencies or CPU affinity exist or are to be expected.

LINUX SCHEDULERS

If Linux would have been stick with the O(1) scheduler, the scheduling analogies between FreeBSD and Linux would be significant. But as mentioned earlier, Linux moved to an O(log n) Complete Fair Scheduler (CFS) [4, 5] with kernel version 2.6.23. Characteristic feature of the CFS is not to use run queues or heuristic algorithms to differentiate between interactive and non-interactive threads. Considering that the FreeBSD developers needed about

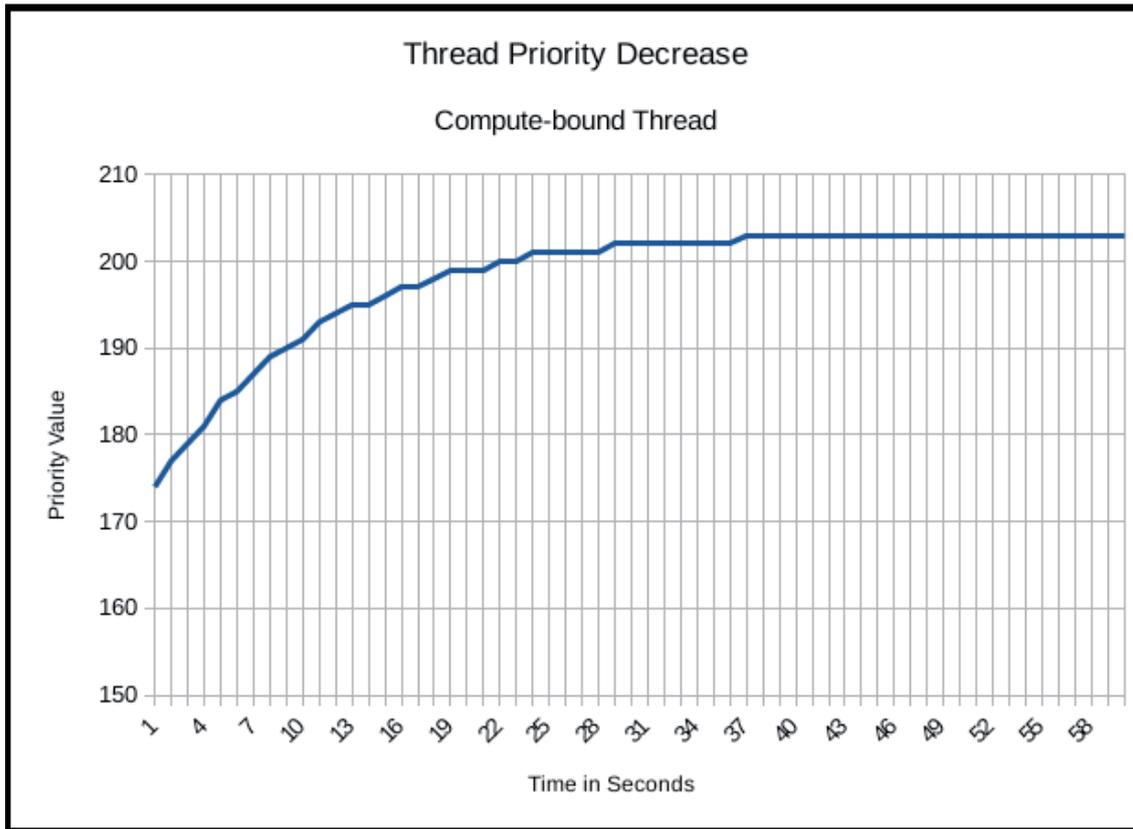


Figure 5: Decrease of priority value of compute-bound threads over time.

10 years to perfect the ULE scheduler, the turn to the CFS in Linux might seem to be understandable. The CFS uses a binary tree (a red-black tree) to manage threads. Key value for each thread is its virtual run time (the real run time altered for example by the nice value), which is compared to the virtual run time of other threads to find the place in the tree. The thread with lowest virtual run-time will get scheduled next. Thread priority only affects the maximal time slice a thread can use. Furthermore, the CFS implements group scheduling. If two users are active on a system and the first of them runs 20 threads and the second one only 10 threads, then most schedulers would favour the first user with in total 2/3 of the CPU time. The CFS is able to assign 50% of the CPU time to each of them. Finally, it shall be mentioned that the Linux CFS uses different scheduling classes, too.

It is not a secret that the world of Linux is colorful. To enhance interactivity on less powerful hardware the Brain Fuck Scheduler (BFS) was added as an alternative to Android, is used in some minor spread Linux distributions and may depending on opinion be superior to the CFS [6].

CONCLUSION

The question whether it is better to spend 10 years to perfect a scheduler like the FreeBSD kernel developers did or whether is it better to replace the scheduler from time to time like the Linux kernel developers did cannot be answered here. But both strategies show that implementing a powerful scheduler which distributes CPU time fairly and which guarantees high interactivity is

anything else but easy.

The not surprising result of this article is that the number of CPU cores has a positive impact on system performance. The effects of cache losses due to harming CPU affinity should not be underestimated. Thus, the higher the number of cores is, the better the CPU affinity of threads can be kept, and the more threads can be run in parallel which is intensively used by modern operating system kernels.

REFERENCES

- [1] "The Design and Implementation of the FreeBSD Operating System", Marshall Kirk McKusick, George V. Neville-Neil, Robert N. M. Watson, 2nd Edition, Addison-Wesley (2015)
- [2] "Mac OS X Internals: A Systems Approach", Amit Singh, Addison-Wesley (2006)
- [3] "Processes, Threads, and Jobs in the Windows Operating System", Mark E. Russinovich and David A. Solomon, <https://www.microsoftpressstore.com/articles/article.aspx?p=2233328&seqNum=7> (2009)
- [4] "Inside the Linux 2.6 Completely Fair Scheduler", M. Tim Jones, <https://www.ibm.com/developerworks/linux/library/l-completely-fair-scheduler/> (2009)
- [5] "Completely Fair Scheduler", https://en.wikipedia.org/wiki/Completely_Fair_Scheduler
- [6] "Brain Fuck Scheduler", https://en.wikipedia.org/wiki/Brain_Fuck_Scheduler



Software und Lizenzverwaltung

DER EINFACHE WEG ZUR SOFTWARE!

Ihre Anforderung

Sie benötigen eine Software, für die es keine von Ihnen nutzbare Rahmenvereinbarung gibt. Die Anzahl der erforderlichen Lizenzen ist nicht genau festgelegt.

Unser Angebot

Wir verfügen über eine Reihe von Rahmen- und Campusvereinbarungen mit namhaften Softwareherstellern und -lieferanten, über die Software auch in geringerer Stückzahl bezogen werden kann. Wir wickeln für Sie die Beschaffung der erforderlichen Lizenzen ab. Wir können uns bei Vertragsverhandlungen und Bedarfsanalysen engagieren. Zugriffslizenzen können auch über Lizenzserver verwaltet werden.

Ihre Vorteile

- > Sie können die benötigte Software in vielen Fällen sofort nutzen.

- > Sie brauchen kein eigenes Ausschreibungs- und Beschaffungsverfahren durchzuführen.
- > Sie ersparen sich die zeitraubenden Verhandlungen mit den Softwareherstellern und -lieferanten.
- > Die Anzahl der benötigten Lizenzen wird Ihnen flexibel zur Verfügung gestellt.
- > Wir können die Nachfrage von verschiedenen Nutzern für neue Lizenzvereinbarungen bündeln.

Interessiert?

Informationen zu bestehenden Lizenzvereinbarungen sind auf der u. g. GWDG-Webseite zu finden. Falls Sie nach spezieller Software suchen, die noch nicht auf unserer Webseite erwähnt ist, kommen Sie bitte auf uns zu. Wir werden prüfen, ob wir eine Vereinbarung abschließen können und bündeln die Nachfrage mit anderen Nutzern.

Stellenangebot

Die GWDG sucht ab sofort zur Unterstützung der Arbeitsgruppe „Basisdienste und Organisation“ (AG O) eine

Studentische Hilfskraft (m/w)

mit einer Beschäftigungszeit von 68 Stunden pro Monat. Die Vergütung erfolgt entsprechend den Regelungen für Studentische/Wissenschaftliche Hilfskräfte.

Aufgabenbereiche

- Mitarbeit bei Softwareentwicklungsprojekten im Umfeld von Web-Anwendungen und Identity- und Access-Management sowie deren Unterstützung im 2nd/3rd-Level-Support
- Das Themengebiet umfasst die Bereiche C#, ASP.NET MVC und .NET CORE oder alternativ die Programmierung im Bereich Liferay Portal mit Java-Web-Technologien (JEE, JSP/JSF).

Anforderungen

- Erfahrungen im Umgang mit Datenbanken wie z. B. MySQL
- Fundierte Kenntnisse in einer Programmiersprache wie C# oder Java
- Erfahrungen mit gängigen Web-Technologien (XHTML, CSS, JavaScript)
- Teamfähigkeit und Spaß am Umgang mit Kunden

Angebot

- Flexible, mit dem Studium vereinbare Arbeitszeiten
- Spannende Themen aus den Bereichen Softwareentwicklung und Identity-Management
- Hilfestellung bei der Einarbeitung in aktuelle Technologien und Konzepte
- Große inhaltliche Nähe zur Wissenschaft und Forschung
- Mitarbeit in einem jungen, kompetenten und engagierten Team

Die GWDG will den Anteil von Frauen in den Bereichen erhöhen, in denen sie unterrepräsentiert sind. Frauen werden deshalb ausdrücklich aufgefordert, sich zu bewerben. Die GWDG ist bemüht, mehr schwerbehinderte Menschen zu beschäftigen. Bewerbungen Schwerbehinderter sind ausdrücklich erwünscht.

Bitte reichen Sie Ihre Bewerbung mit allen wichtigen Unterlagen **bis zum 31. März 2018** über das Online-Formular unter <https://s-lotus.gwdg.de/gwdgdb/ago/20180227.nsf/bewerbung> ein.

Fragen zur ausgeschriebenen Stelle beantworten Ihnen:

Herr Stephan Hilker

Tel.: 0551 201-2167

E-Mail: stephan.hilker@gwdg.de oder

Herr Sascha Klöppner

Tel.: 0551 201-2162

E-Mail: sascha.kloppner@gwdg.de

Stellenangebot

Die GWDG sucht ab sofort für die Mitarbeit im von der Göttingen eResearch Alliance initiierten BMBF-geförderten Projekt „GRAcE – Göttingen Research Data Exploratory – management parameters derived from the Göttingen eResearch Alliance“ zur Unterstützung bei Interviews, einer Online-Befragung und weiteren Tätigkeiten eine

Wissenschaftliche Hilfskraft (m/w)

mit einer Beschäftigungszeit von bis zu 83 Stunden pro Monat auf bis zu 14 Monate befristet. Die Vergütung erfolgt entsprechend den Regelungen für Studentische/Wissenschaftliche Hilfskräfte.

Die „Göttingen eResearch Alliance“ (eRA) unterstützt seit 2014 Wissenschaftler am Göttingen Campus bei der Verwendung innovativer digitaler Informations- und Kommunikationstechnologien. Hierzu zählen sowohl das Forschungsdatenmanagement, die Visualisierung von Forschungsdaten, der Einsatz von Kollaborationsplattformen und virtuellen Forschungsumgebungen, als auch Publikationsstrategien und weitere Aspekte in den einzelnen Phasen von Forschungsvorhaben. Die Göttingen eResearch Alliance wird gemeinsam von der Niedersächsischen Staats- und Universitätsbibliothek Göttingen und der Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen betrieben.

Ziel des Projektes GRAcE ist es, in Zusammenarbeit mit der Universitätsmedizin Göttingen verallgemeinerbare Faktoren des Forschungsdatenmanagements im universitären Umfeld zu identifizieren, um daraus eine Ressourcenabschätzung entwickeln zu können, die in ein langfristig nutzbares und auf andere Standorte übertragbares Planungsinstrument überführt werden kann. Zudem wird untersucht, inwiefern Services zum Umgang mit Forschungsdaten, die für spezielle Projektkontexte entwickelt wurden, für eine Nutzung in anderen Umgebungen geeignet sind und welche Anpassungsmaßnahmen hierfür ggfs. notwendig sind. In diesem Zusammenhang werden unter anderem Interviews, eine Online-Befragung und Workshops durchgeführt.

Aufgabenbereiche

- Transkription von Interviews
- Unterstützung bei der Vorbereitung, Durchführung und Auswertung von Interviews und einer Online-Befragung
- Unterstützung bei der Organisation regionaler und überregionaler Workshops

- Redaktionelle Unterstützung bei der Erstellung von Schulungsmaterialien und Berichten
- Unterstützung von Projektmanagement und Öffentlichkeitsarbeit

Anforderungen

- Abgeschlossenes Hochschulstudium (Bachelor) in einem sozialwissenschaftlichen oder einem für das Tätigkeitsprofil relevanten Fach und laufendes weiterführendes Hochschulstudium
- Gute Kenntnisse der Methoden empirischer Sozialforschung
- Gute kommunikative und organisatorische Fähigkeiten
- Sehr gute Kenntnisse in gängigen Office-Anwendungen
- Sehr gute Deutschkenntnisse

Wünschenswert

- Erfahrung in der Transkription von Interviews
- Erfahrung in der Planung, Durchführung und Auswertung von Interviews und/oder Umfragen
- Erfahrung bei der Organisation und Durchführung von Workshops und ähnlichen Veranstaltungen
- Kenntnisse im Bereich Forschungsdatenmanagement und Forschungsdateninfrastrukturen

Die GWDG will den Anteil von Frauen in den Bereichen erhöhen, in denen sie unterrepräsentiert sind. Frauen werden deshalb ausdrücklich aufgefordert, sich zu bewerben. Die GWDG ist bemüht, mehr schwerbehinderte Menschen zu beschäftigen. Bewerbungen Schwerbehinderter sind ausdrücklich erwünscht.

Bitte reichen Sie Ihre Bewerbung mit allen wichtigen Unterlagen **bis zum 6. April 2018** über das Online-Formular unter <https://s-lotus.gwdg.de/gwdgdb/age/20180308.nsf/bewerbung> ein. Die Bewerbungsgespräche sind für den 20. April 2018 geplant.

Bewerbungs- und Reisekosten werden nicht übernommen oder erstattet.

Fragen zur ausgeschriebenen Stelle beantworten Ihnen:

Herr Dr. Sven Bingert

Tel.: 0551 201-2164

E-Mail: sven.bingert@gwdg.de oder

Frau Claudia Engelhardt

Tel.: 0551 39-20284

E-Mail: claudia.engelhardt@sub.uni-goettingen.de

NEUER AUSZUBILDENDER STEPHAN BÖMEKE

Am 1. Februar 2018 hat Herr Stephan Bömeke die Ausbildung zum Fachinformatiker in der Fachrichtung Anwendungsentwicklung in der Arbeitsgruppe „Basisdienste und Organisation“ (AG O) begonnen. Herr Bömeke kann sich dabei auf Kenntnisse in der Java-Programmierung stützen, die er während eines zwischenzeitlichen Studiums der Wirtschaftswissenschaften erlernt hat. Nach einem Umweg über den Fachinformatiker – Fachrichtung Systemintegration stehen nun Programmiersprachen und -umgebungen im Enterprise-Umfeld im Fokus. Herr Bömeke ist per E-Mail unter *stephan.boemeke@gwdg.de* und telefonisch unter 0551 201-2190 zu erreichen.



Pohl



PROMOTION ERFOLGREICH ABGESCHLOSSEN DR. FARAZ FATEMI MOGHADDAM

Herr Dr. Faraz Fatemi Moghaddam, wissenschaftlicher Mitarbeiter der Arbeitsgruppe „eScience“ (AG E), hat am 25. Januar 2018 seine Promotion zum Dr. rer. nat. der Georg-August-Universität Göttingen im Promotionsprogramm PCS der Georg-August University School of Science (GAUSS) erfolgreich abgeschlossen. Das Thema seiner Dissertation lautete „Multi-Layered Policy Generation and Management in Clouds“. Wir gratulieren hierzu ganz herzlich.

Wieder

INFORMATIONEN:
support@gwdg.de
0551 201-1523

März bis
Dezember 2018

Kurse



KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	07.03.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	28.02.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	08.03.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	01.03.2018	4
INDESIGN – GRUNDLAGEN	Töpfer	13.03. – 14.03.2018 9:30 – 16:00 Uhr	06.03.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	04.04.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	28.03.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	05.04.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	29.03.2018	4
USING THE GWGD SCIENTIFIC COMPUTE CLUSTER – AN INTRODUCTION	Dr. Boehme, Dr. Ehlers	09.04.2018 9:30 – 16:00 Uhr	02.04.2018	4
PARALLELRECHNER-PROGRAMMIERUNG MIT MPI	Prof. Haan	10.04. – 11.04.2018 9:15 – 17:00 Uhr	03.04.2018	8
EINFÜHRUNG IN DIE STATISTISCHE DATEN-ANALYSE MIT SPSS	Cordes	17.04. – 18.04.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	10.04.2018	8
PROGRAMMING WITH CUDA – AN INTRODUCTION	Prof. Haan	17.04.2018 9:15 – 17:00 Uhr	10.04.2018	4
PHOTOSHOP FÜR FORTGESCHRITTENE	Töpfer	24.04. – 25.04.2018 9:30 – 16:00 Uhr	17.04.2018	8

KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
QUICKSTARTING R: EINE ANWENDUNGSORIENTIERTE EINFÜHRUNG IN DAS STATISTIKPAKET R	Cordes	15.05. – 16.05.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	08.05.2018	8
ADMINISTRATION VON PCS IM ACTIVE DIRECTORY DER GWDC	Quentin	24.05.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	17.05.2018	4
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	30.05.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	23.05.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	31.05.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	24.05.2018	4
INDESIGN – AUFBAUKURS	Töpfer	05.06. – 06.06.2018 9:30 – 16:00 Uhr	29.05.2018	8
OUTLOOK – E-MAIL UND GROUPWARE	Helmvoigt	14.06.2018 9:15 – 12:00 und 13:00 – 16:00 Uhr	07.06.2018	4
ANGEWANDTE STATISTIK MIT SPSS FÜR NUTZER MIT VORKENNTNISSEN	Cordes	20.06. – 21.06.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	13.06.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	27.06.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	20.06.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	28.06.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	21.06.2018	4
STATISTIK MIT R FÜR TEILNEHMER MIT VOR-KENNTNISSEN – VON DER ANALYSE ZUM BERICHT	Cordes	03.07. – 04.07.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	26.06.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	15.08.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	08.08.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	16.08.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	09.08.2018	4
GRUNDLAGEN DER BILDBEARBEITUNG MIT PHOTOSHOP	Töpfer	21.08. – 22.08.2018 9:30 – 16:00 Uhr	14.08.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	12.09.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	05.09.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	13.09.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	06.09.2018	4
INDESIGN – GRUNDLAGEN	Töpfer	18.09. – 19.09.2018 9:30 – 16:00 Uhr	11.09.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	17.10.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	10.10.2018	4

KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	18.10.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	11.10.2018	4
PHOTOSHOP FÜR FORTGESCHRITTENE	Töpfer	23.10. – 24.10.2018 9:30 – 16:00 Uhr	16.10.2018	8
EINFÜHRUNG IN DIE STATISTISCHE DATEN-ANALYSE MIT SPSS	Cordes	13.11. – 14.11.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	06.11.2018	8
ADMINISTRATION VON PCS IM ACTIVE DIRECTORY DER GWDG	Quentin	15.11.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	08.11.2018	4
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	21.11.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	14.11.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	22.11.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	15.11.2018	4
INDESIGN – AUFBAUKURS	Töpfer	27.11. – 28.11.2018 9:30 – 16:00 Uhr	20.11.2018	8
OUTLOOK – E-MAIL UND GROUPWARE	Helmvoigt	06.12.2018 9:15 – 12:00 und 13:00 – 16:00 Uhr	29.11.2018	4
ANGEWANDTE STATISTIK MIT SPSS FÜR NUTZER MIT VORKENNTNISSEN	Cordes	11.12. – 12.12.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	04.12.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	19.12.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	12.12.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	20.12.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	13.12.2018	4

Teilnehmerkreis

Das Kursangebot der GWDG richtet sich an alle Mitarbeiterinnen und Mitarbeiter aus den Instituten der Universität Göttingen und der Max-Planck-Gesellschaft sowie aus einigen anderen wissenschaftlichen Einrichtungen.

Anmeldung

Anmeldungen können schriftlich per Brief oder per Fax unter der Nummer 0551 201-2150 an die GWDG, Postfach 2841, 37018 Göttingen oder per E-Mail an die Adresse support@gwdg.de erfolgen. Für die schriftliche Anmeldung steht unter <https://www.gwdg.de/antragsformulare> ein Formular zur Verfügung. Telefonische Anmeldungen können leider nicht angenommen werden.

Kosten bzw. Gebühren

Unsere Kurse werden wie die meisten anderen Leistungen der GWDG in Arbeitseinheiten (AE) vom jeweiligen Institutskontin-

gent abgerechnet. Für die Institute der Universität Göttingen und der Max-Planck-Gesellschaft erfolgt keine Abrechnung in EUR.

Absage

Sie können bis zu acht Tagen vor Kursbeginn per E-Mail an support@gwdg.de oder telefonisch unter 0551 201-1523 absagen. Bei späteren Absagen werden allerdings die für die Kurse berechneten AE vom jeweiligen Institutskontingent abgebucht.

Kursorte

Alle Kurse finden im Kursraum oder Vortragsraum der GWDG statt. Die Wegbeschreibung zur GWDG sowie der Lageplan sind unter <https://www.gwdg.de/lageplan> zu finden.

Kurstermine

Die genauen Kurstermine und -zeiten sowie aktuelle kurzfristige Informationen zu den Kursen, insbesondere zu freien Plätzen, sind unter <https://www.gwdg.de/kursprogramm> zu finden.



Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen