

# GWDG NACHRICHTEN 04|18

Rocket.Chat

IHK-Abschlussprojekt

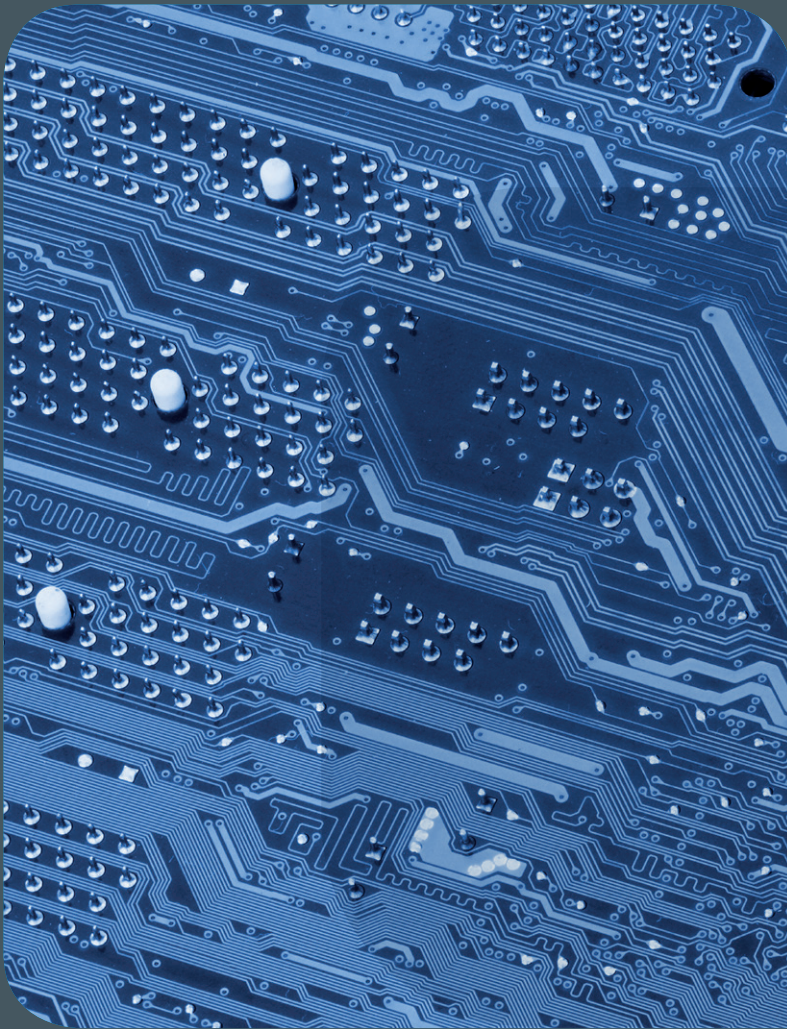
Printserver und Linux

HLRN-IV

Entwicklung von  
GPU-Anwendungen

ZEITSCHRIFT FÜR DIE KUNDEN DER GWDG





## GWGD NACHRICHTEN

# 04|18 Inhalt

.....

**4 Messaging-Dienst „Rocket.Chat“ geht in den Regelbetrieb** **6 Die Integration des DFN-Webkonferenzdienstes in den Virtuellen Forschungsraum MWW – ein IHK-Abschlussprojekt** **9 Tipps & Tricks** **10 HLRN-IV in Berlin und Göttingen** **12 Kernkraftwerke bei der Arbeit** **16 Kurz & knapp** **17 Kurse**

### Impressum

.....

Zeitschrift für die Kunden der GWGD

ISSN 0940-4686  
41. Jahrgang  
Ausgabe 4/2018

**Erscheinungsweise:**  
monatlich

[www.gwdg.de/gwdg-nr](http://www.gwdg.de/gwdg-nr)

**Auflage:**  
550

**Fotos:**  
© Cybrain - Fotolia.com (1)  
© pineapple - Fotolia.com (4)  
© MPLbpc-Medienservice (3)  
© Zuse-Institut Berlin (10)  
© Universität Göttingen / Jan Vetter (10)  
© GWGD (2, 17)

**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](mailto:thomas.otto@gwdg.de)

**Herstellung:**  
Franziska Schimek  
E-Mail: [franziska.schimek@gwdg.de](mailto:franziska.schimek@gwdg.de)

**Druck:**  
Kreationszeit GmbH, Rosdorf



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

## *Liebe Kunden und Freunde der GWDDG,*

*über die Jahre stellt man immer wieder fest: Eine der größten Herausforderungen in unserer Welt bleibt Kommunikation. Dies betrifft auch die Bereitstellung und den Betrieb von Informationsdiensten. Dabei handelt es sich weniger um ein technisches, sondern vielmehr um ein organisatorisches Thema: Wie erreicht man mit relevanten Informationen die richtige Zielgruppe? Wir haben heutzutage ein Überangebot an Kommunikationskanälen, die sich gleichzeitig in Technik und Art hochdynamisch verändern. Welche die richtige ist, hängt zunehmend vom Thema ab. Sicherlich spielt hier auch eine Rolle, was gerade en vogue ist, und zentrale Vorgaben bringen hier wenig.*

*In Gesprächen mit Nutzern der GWDDG wird immer wieder Interesse an mehr Informationen gewünscht. Dabei stellt sich aber im Detail häufig heraus, dass die Informationen bereits vorliegen: entweder im Web, in einem Wiki, den GWDDG-Nachrichten, auf Mailinglisten, Twitter usw. Eine Reduktion der Plattformen zur Vereinfachung würde das Gegenteil erreichen. Letztlich definiert die Zielgruppe, welche Plattformen sie goutiert. Dieses Phänomen ist jedoch nicht GWDDG-spezifisch, sondern scheint immanent in allen größeren Organisationen vorzuliegen, in denen man darum kämpft, Informationen geeignet weiterzureichen.*

*In dieser Ausgabe der GWDDG-Nachrichten finden Sie dazu einen Artikel über Rocket.Chat als eine weitere Messaging-Plattform, die sich zunehmender Beliebtheit erfreut. Sie wird sicherlich andere nicht ablösen und auch nicht die letzte bleiben.*

*Ich wünsche Ihnen viel Spaß mit diesem und den weiteren Beiträgen der GWDDG-Nachrichten.*

**Ramin Yahyapour**

GWDDG – IT in der Wissenschaft



# Messaging-Dienst „Rocket.Chat“ geht in den Regelbetrieb

## Text und Kontakt:

Barbara Altmann  
barbara.altmann@gwdg.de  
0551 201-2190

Björn Braunschweig  
bjoern.braunschweig@gwdg.de  
0551 201-2133

Jannis Thomae  
jannis.thomae@gwdg.de  
0551 201-2178

Benedikt Wegmann  
benedikt.wegmann@gwdg.de  
0551 201-1870

Im Mai letzten Jahres startete die GWDG mit dem webbasierten Messenger „Rocket.Chat“ einen Dienst mit Fokus auf Gruppenkommunikation zunächst im offenen Testbetrieb. Diese einjährige Testphase endet nun im Mai. Aufgrund der überaus positiven Resonanz und der gesammelten Erfahrungen beim Betrieb geht der Dienst dann in den Regelbetrieb über und erweitert somit das reguläre Dienst-Portfolio der GWDG.

## WOZU DIENTE DIE TESTPHASE?

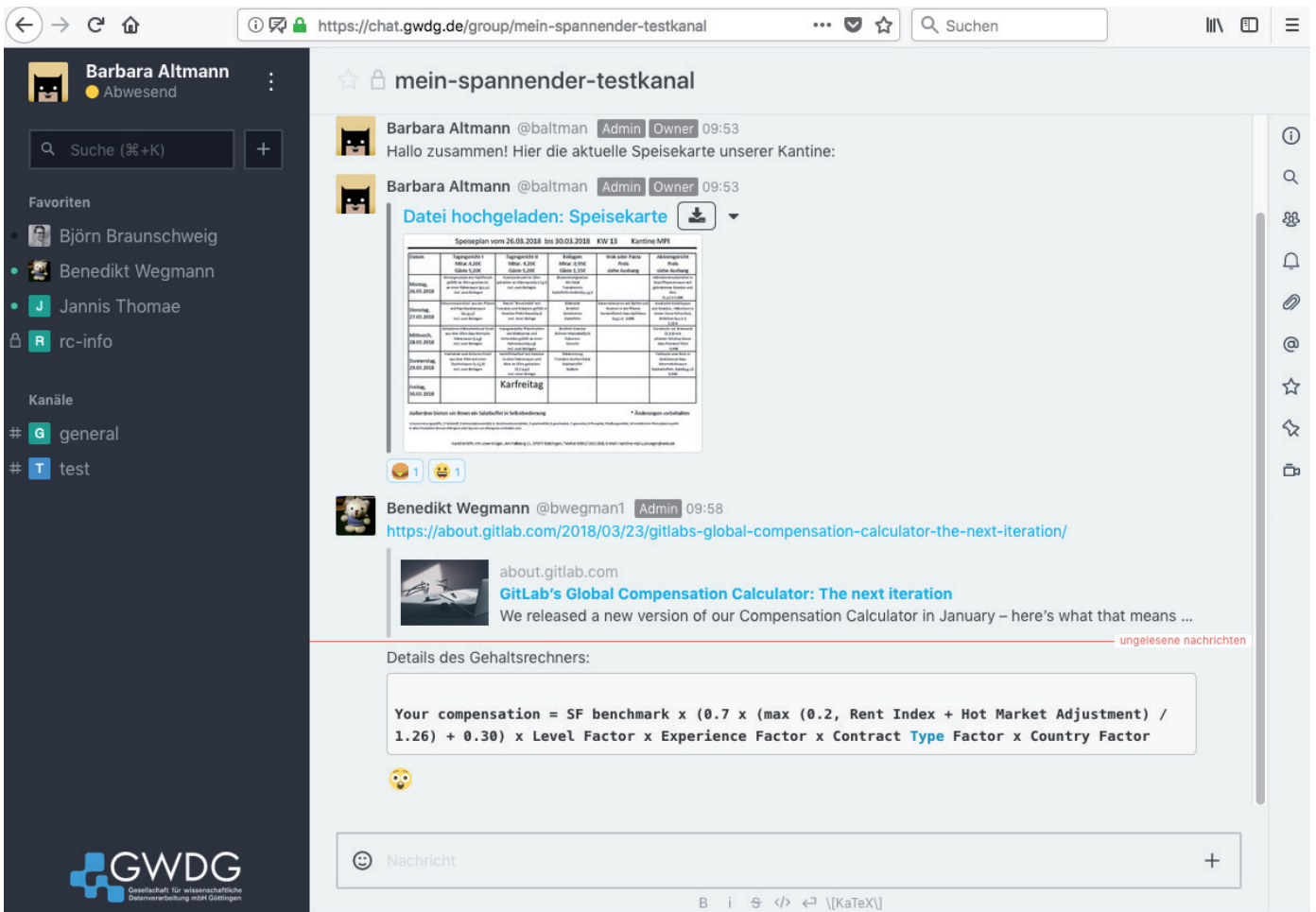
Bevor ein Dienst bei der GWDG angeboten werden kann, müssen eine Reihe von Fragen geklärt werden. Dazu gehören, ob und wie gut der Dienst angenommen wird, wie sich der Betrieb und der Support gestalten, ob ein wirtschaftlicher Betrieb möglich ist und wie dieser aussehen kann. Diese Fragen konnten in der seit Mai 2017 laufenden Testphase von Rocket.Chat beantwortet werden. Der Dienst stellte sich sowohl im Betrieb als auch im Support als vergleichbar „pflegeleicht“ heraus. Außerdem wurde in der Testphase beschlossen, dass für Rocket.Chat bis auf Weiteres keine Arbeitseinheiten abgerechnet werden.

## WELCHE ERGEBNISSE HAT DIE EINJÄHRIGE TESTPHASE GELIEFERT?

Die Statistiken zeigen, dass der Dienst inzwischen fast 1.000 Nutzer hat, wobei sich zu Stoßzeiten rund 200 Nutzer gleichzeitig am Dienst anmelden. Inzwischen werden über 2.000 Unterhaltungen geführt – die meisten davon sind Direktunterhaltungen zwischen je zwei Nutzern. Der Rest sind rund 300 öffentliche und fast 50 private Kanäle. Zusammen haben alle Rocket.Chat-Nutzer im vergangenen Jahr fast 230.000 Nachrichten verfasst.

## Messaging Service “Rocket.Chat” goes live

Last year, the GWDG launched the messaging service “Rocket.Chat” in an annual test phase. The test phase expires in May and from May on the service goes live as scheduled. The reasons for this decision are based on the approval and the time expenditure, which includes the operation and the support of the service, which are both satisfying. Rocket.Chat has a continuously rising number of users and a low expenditure of time as well. For the users of Rocket.Chat, nothing changes after the end of the test phase. The service can be used as usual. A valid GWDG account is required to use Rocket.Chat. After the first login, users will automatically be added to the #general-channel which can be left if required. After logging in, you can join public channels or start one-to-one conversations with other members. Private groups are not joinable. You can be invited to private groups or you can create your own channel which can be set private. To start, just sign in at <https://chat.gwdg.de> with your common GWDG credentials.



1\_Chatfenster bei Rocket.Chat

Rocket.Chat wurde ausgesprochen positiv aufgenommen und ist inzwischen auch innerhalb der GWDG für die interne Kommunikation unverzichtbar. Um gerade auch in „heiklen“ Situationen verfügbar zu sein, wurde der Dienst schon in der Anfangsphase in das Gold-Cluster der GWDG verschoben, damit eine hohe Verfügbarkeit garantiert werden kann. So bewährte sich der Dienst auch als internes und externes „Sprachrohr“ beispielsweise während der Stromabschaltung Mitte letzten Jahres.

### WELCHE VERÄNDERUNGEN MÜSSEN DIE BENUTZER DES DIENSTES NACH ABSCHLUSS DER TESTPHASE ERWARTEN?

Für die Benutzer von Rocket.Chat wird der Übergang nicht bemerkbar sein. Der Dienst kann also wie bisher verwendet werden.

### ICH HABE BISHER SLACK BENUTZT. KANN ICH AUF ROCKET.CHAT UMSTEIGEN?

Rocket.Chat bietet eine Schnittstelle für den Import von Daten aus Slack an. Darüber können Kanäle, Nachrichten und Benutzer übertragen werden. Hierbei sind Anpassungen notwendig, wie z. B. die Konvertierung von Slack-Benutzernamen in GWDG-Benutzernamen. Da die Anpassungen manuell erfolgen, muss je nach Größe des Exports individuell entschieden werden, ob ein Import von der GWDG geleistet werden kann.

### WIE UND WO KANN ICH DEN DIENST NUTZEN ODER AUSPROBIEREN?

Jeder, der einen Account bei der GWDG hat, kann sich mit seinen gewohnten Zugangsdaten unter <https://chat.gwdg.de> anmelden und sofort loslegen. Auch ein über das Kundenportal <https://www.gwdg.de> selbstregistrierter Account der GWDG kann hier genutzt werden. Nach dem Einloggen befindet sich der Benutzer im Kanal #general, dem zunächst jeder neue Benutzer hinzugefügt wird. Weitere Kanäle können über das Suchfeld am linken oberen Fensterrand gefunden werden (siehe auch Abb. 1). Zusätzliche Informationen zur Nutzung von Rocket.Chat finden Sie in den GWDG-Nachrichten 5/2017 ([https://www.gwdg.de/documents/20182/27257/GN\\_5-2017\\_www.pdf](https://www.gwdg.de/documents/20182/27257/GN_5-2017_www.pdf)) oder in unserem Wiki-Bereich unter [https://info.gwdg.de/docs/doku.php?id=de:services:email\\_collaboration:rocketchat](https://info.gwdg.de/docs/doku.php?id=de:services:email_collaboration:rocketchat).

### SAMMLUNG WICHTIGER LINKS

Rocket.Chat der GWDG	<a href="https://chat.gwdg.de">https://chat.gwdg.de</a>
Offizielle Rocket.Chat-Seite	<a href="https://rocket.chat">https://rocket.chat</a>
Hilfe für Benutzer (Englisch)	<a href="https://rocket.chat/docs/user-guides">https://rocket.chat/docs/user-guides</a>
Download von Clients für Desktop- und Mobilgeräte	<a href="https://rocket.chat/download">https://rocket.chat/download</a>
Erstellung eines neuen GWDG-Accounts	<a href="https://www.gwdg.de/registration">https://www.gwdg.de/registration</a>
Informationen zum GWDG-Dienst „Rocket-Chat“	<a href="https://www.gwdg.de/services/e-mail-collaboration/rocket.chat">https://www.gwdg.de/services/e-mail-collaboration/rocket.chat</a>

# Die Integration des DFN-Webkonferenzdienstes in den Virtuellen Forschungsraum MWW – ein IHK-Abschlussprojekt

## Text und Kontakt:

Barbara Altmann  
barbara.altmann@gwdg.de  
0551 201-2190

Im Januar beendete ich erfolgreich meine Ausbildung zur Fachinformatikerin für Anwendungsentwicklung bei der GWDG. Teil der Abschlussprüfung war dabei die Durchführung eines 70-stündigen Programmierprojektes. Hierfür setzte ich ein Verwaltungstool für den Webkonferenzdienst des DFN um, der als Weboberfläche in einen von der GWDG implementierten Virtuellen Forschungsraum eingebunden wird.

## FACHINFORMATIKER-AUSBILDUNG

Die GWDG bildet bereits seit mehreren Jahren Fachinformatiker in der Fachrichtung Anwendungsentwicklung aus. Hier erlernen angehende Programmierer nicht nur Programmiersprachen (schwerpunktmäßig Java), sondern auch den Umgang mit Bibliotheken und Frameworks, die Auswahl der richtigen Werkzeuge, das Verständnis für Programmierparadigmen und Softwarearchitekturen und die Kommunikation von Rechnern über Netzwerke. Praktisch können sich die Auszubildenden in die Programmierung des GWDG-Kundenportals <https://www.gwdg.de> und des IdM-Portals <https://idm.gwdg.de> einbringen und das Erlernte anwenden.

Begleitend besuchen die Auszubildenden die Berufsschule, wo neben Fach- und Kernkompetenzen (z. B. Objektorientierung, Hardware, Netzwerke und kaufmännisches Wissen) auch Englisch, Deutsch und Politik unterrichtet werden. Nach drei Jahren wird die Ausbildung mit der Abschlussprüfung der IHK abgeschlossen.

Seit Sommer 2017 bietet die GWDG außerdem die Ausbildung zum Fachinformatiker in der Fachrichtung Systemintegration an. Der schulische Teil ist hier deckungsgleich mit dem des Anwendungsentwicklers, doch die praktischen Tätigkeiten konzentrieren sich auf die Administration von Systemen, Netzwerken und Infrastruktur.

## VERLAUF DER AUSBILDUNG

Im Oktober 2015 begann ich meine Ausbildung zur Fachinformatikerin für Anwendungsentwicklung bei der GWDG. Zuvor hatte ich mein Mathematikstudium abgebrochen – ein Punkt im Lebenslauf, der sich entgegen der geläufigen Erwartungen nicht negativ bei der Suche eines Ausbildungsplatzes auswirkt, sondern im Gegenteil oftmals positiv angerechnet wird. Die dort erworbenen Fähigkeiten zum analytischen Denken und selbstständigen Lernen

sowie erste Erfahrungen mit der Programmierung erwiesen sich in der Ausbildung als hervorragende Voraussetzungen.

Gerade in der Programmierung müssen sich die Auszubildenden einer steilen Lernkurve stellen, denn von der ersten Variablendeklaration bis zum versierten Webentwickler ist es ein weiter Weg. Die ersten Wochen und Monate der Ausbildung haben für mich somit den bekannten „Sprung ins kalte Wasser“ bedeutet – die eine oder andere kurze Panik inbegriffen. Schnell zeigte sich, wie wichtig es war, Kollegen mit Fragen zu löchern und selbstständiges Engagement zu zeigen.

Bei der GWDG beteiligen sich viele Mitarbeiter an der Betreuung während der Ausbildung, was einen weit gefächerten Einblick in die Welt der IT im Allgemeinen und die Anwendungsentwicklung im Speziellen bedeutet. Angesiedelt werden die zukünftigen Programmierer zunächst im Team, welches das Kundenportal der

## Graduation project for an apprenticeship at the GWDG

This January I successfully finished my apprenticeship as a "Computer Science Expert, Subject Area: Software Development". Part of the final exam was a 70 hour coding project. For the topic I chose the integration of an existing web conference service, hosted by the DFN (Deutsches Forschungsnetz), into a virtual research environment that is currently being developed by the GWDG. This project required me to work with multiple frameworks and interfaces, amongst others Liferay Portal and the Liferay Service Builder, a code generator tool provided by the former, as well as the DFN webservice. Development for the backend happened primarily in Java. For the frontend I used Angular with TypeScript as the programming language of choice. HTML and CSS were used for the views.

GWDG entwickelt – was hauptsächlich die Arbeit mit Java nach sich zieht. Dienstags verbrachte ich den Vormittag (und manchmal auch den Nachmittag) damit, beim IdM-Team die .NET-Entwicklung mit C# zu erkunden. Hier dürfen sich die Auszubildenden auch das erste Mal produktiv einbringen: Zahlreiche kleine Funktionen und Bugfixes auf <https://idm.gwdg.de> wurden in den letzten Jahren von den Azubis umgesetzt. Und einmal alle zwei Wochen finden die Programmierübungen statt, wo die Auszubildenden mal über den Tellerrand blicken und Aspekte der Programmierung kennenlernen dürfen, die unterschiedlicher nicht sein könnten: Programmierung von Treibern und Plugins, automatische Erstellung von Entwicklerdokumentationen, Einblicke in die Arbeit von Compilern und sogar Assembler-Code stehen auf dem Programm.

Darüber hinaus ermöglichen hilfsbereite Kollegen auch Exkurse in andere Arbeitsfelder der GWDG: Ich durfte beispielsweise dabei sein, als WLAN-Ausleuchtungen vorgenommen und Switches konfiguriert wurden.

## DAS ABSCHLUSSPROJEKT

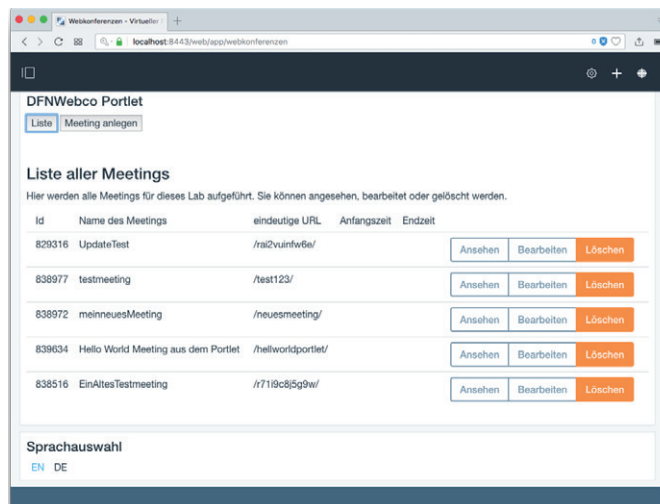
Am 22. Januar 2018 beendete ich erfolgreich meine Ausbildung. Im Rahmen der IHK-Abschlussprüfung ist in der Fachrichtung Anwendungsentwicklung ein 70-stündiges Projekt vorgesehen, in dem der Prüfling seine in der Ausbildung erworbenen Fähigkeiten unter Beweis stellen soll. Dieses Projekt soll möglichst im beruflichen Alltag des Betriebs verankert und somit „lebensecht“ sein; die Themenwahl ist jedoch frei. Für mich stand die Durchführung dieses Abschlussprojektes bereits im Oktober 2017 an.

Die GWDG entwickelt im Rahmen eines drittmittelfinanzierten Forschungsprojektes einen Virtuellen Forschungsraum für den Forschungsverbund Marbach Weimar Wolfenbüttel (kurz MWW). Dabei handelt es sich um ein Webportal, das Wissenschaftlern Werkzeuge für ihre Arbeit zur Verfügung stellt und zahlreiche Kollaborationsmaßnahmen ermöglicht. Ein Teilauftrag ist die Integration des bestehenden Webkonferenzdienstes, der vom DFN-Verein (DFN = Deutsches Forschungsnetz) angeboten wird, in die Oberfläche des Virtuellen Forschungsraums. Unter anderem bedeutet dies, dass im Virtuellen Forschungsraum Webkonferenzen angelegt, eingesehen, bearbeitet und gelöscht werden können sollen. Die Umsetzung dieser Basisfunktionen schien für ein Abschlussprojekt gut geeignet. Abb. 1 und 2 zeigen zwei der als Ergebnis dieser Arbeit entstandenen Webansichten.

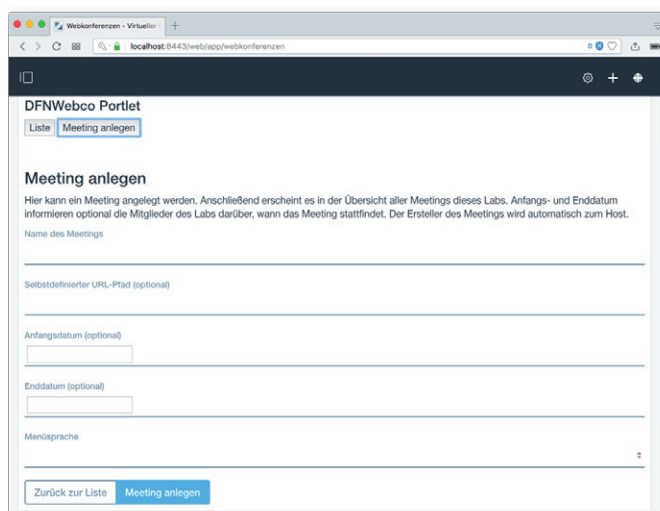
## DIE UMSETZUNG

Der DFN bietet einen Webservice an, welcher die oben genannten Basisfunktionen für den angebotenen Webkonferenzdienst bereitstellt. Darüber hinaus sind einige weitere Funktionen vorhanden. Ziel des Projektes war es, diese Funktionen in der Weboberfläche des Virtuellen Forschungsraums zur Verfügung zu stellen und somit eine Verwaltung der Webkonferenzen aus dem Virtuellen Forschungsraum heraus zu ermöglichen. Dazu waren folgende Teilschritte notwendig:

1. Implementierung eines **Clients**, der die vom Webservice angebotenen Funktionen aufruft und die zurückgelieferten XML-Daten für die weitere Verwendung aufbereitet: Dieser Programmteil übernimmt also die Kommunikation mit dem Webservice des DFN und ist damit das einzige Modul, das die „Sprache“ des DFN-Webservice sprechen



1\_Liste aller Meetings



2\_Anlegen eines Meetings

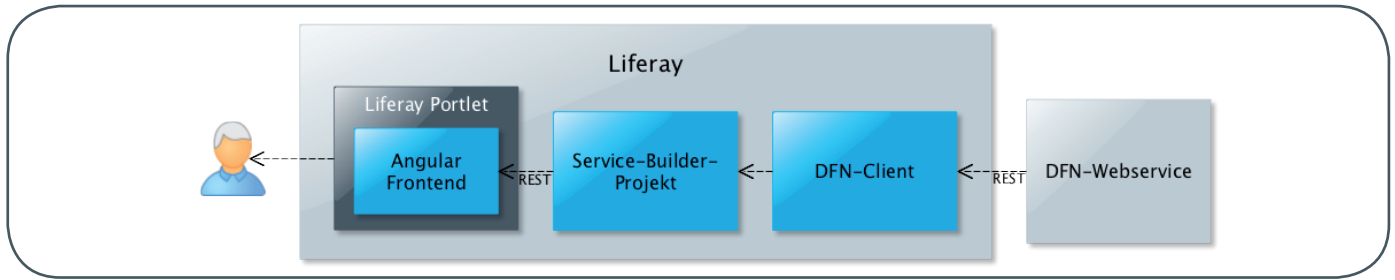
muss. Der Rest der Anwendung muss sich darum nicht mehr kümmern.

2. Implementierung der **Geschäftslogikfunktionen (auch „Backend“)**, welche die empfangenen Daten verarbeiten, verändern und sonstige Operationen darauf vornehmen: Diese Schicht der Anwendung setzt die eigentlichen Funktionalitäten um – hier geschieht die „Magie“.
3. Implementierung der eigentlichen **Weboberfläche (auch „Frontend“)**, welche die Bedienung im Virtuellen Forschungsraum ermöglicht: Hier kann der Benutzer die in 1. und 2. implementierten Funktionen aufrufen und Webkonferenzen verwalten. Dazu bedarf es in dieser Schicht der Anwendung nur noch der Betätigung einer Schaltfläche. Der Benutzer wird also der komplizierten Kommunikation mit dem Webservice enthoben und genießt den gewohnten Komfort einer intuitiven Weboberfläche.

Abb. 3 zeigt den groben Aufbau dieser drei Schichten sowie den dadurch erfolgenden Informationsfluss.

## TECHNISCHE KOMPONENTEN

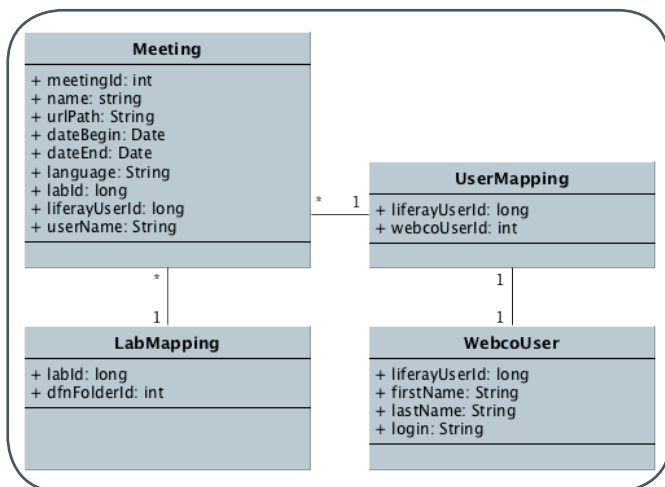
Der DFN-Client sowie die Geschäftslogik wurden mit Java umgesetzt, während bei der Weboberfläche Angular unter Verwendung der Programmiersprache TypeScript zum Einsatz kam. Für die eigentlichen Ansichten wurden die Webstandards HTML und CSS benutzt.



3\_Architektur

Darüber hinaus gibt die Programmierung im Virtuellen Forschungsraum die Verwendung von zahlreichen Komponenten vor, unter anderem beispielsweise des Portal-Frameworks Liferay, das auch im Kundenportal der GWDG zum Einsatz kommt. Deshalb bestand die Herausforderung dieses Projektes hauptsächlich darin, diese Komponenten zu vereinen und die fehlerfreie Kommunikation zwischen ihnen sicherzustellen. Insbesondere beinhaltete dies die Kommunikation mit dem Webkonferenzdienst des DFN über dessen Webservice (per XML), das Konvertieren der empfangenen Daten in Java-Objekte, das Abbilden dieser Objekte auf Strukturen des Virtuellen Forschungsraums und die Weitergabe der Daten an die Weboberfläche (per JSON), wo diese vom Benutzer eingesehen und angepasst werden können.

Ein zentrales bei der Umsetzung der Geschäftslogik verwendetes Werkzeug war der Liferay Service Builder, ein von Liferay zur Verfügung gestelltes Code-Generierungswerkzeug. Dieses ermöglichte es, einen Großteil des benötigten Programmcodes mithilfe einer Konfiguration automatisch zu erzeugen. Für die Erstellung dieser Konfiguration war zunächst der Entwurf eines Datenmodells erforderlich (siehe Abb. 4), in dem detailliert dargestellt wurde, welche Form die von der Anwendung zu verarbeitenden Daten haben werden. Der Liferay Service Builder generierte hieraus anschließend Java-Klassen, in denen unterschiedliche Funktionen vordefiniert wurden – unter anderem das Ablegen der Daten in der zu Liferay gehörenden Datenbank. Auf diese Weise konnten viele komplexe Funktionalitäten mit einem vergleichsweise geringen Zeitumfang umgesetzt werden.



4\_Datenmodell

## HERAUSFORDERUNGEN

Die Abschlussprojekte der Auszubildenden sind so unterschiedlich wie die Ausbildungsbetriebe und die dort geleistete Arbeit. Entsprechend haben unterschiedliche Azubis mit gänzlich

unterschiedlichen Herausforderungen zu kämpfen.

In meinem Fall bestand die größte Herausforderung darin, die unterschiedlichen im Kontext des Virtuellen Forschungsraums verwendeten Bibliotheken und Frameworks unter einen Hut zu bekommen. Diese gestalten sich ein wenig wie Puzzleteile, die nicht zusammenpassen: Um die Zusammenarbeit aller Module untereinander zu gewährleisten, müssen erst die Zwischenstücke zwischen den Einzelteilen umgesetzt werden. Dies beinhaltet beispielsweise, die Fehlerausgaben eines Moduls aufzufangen und zu verarbeiten, statt sie ungefiltert an ein anderes Modul weiterzugeben.

Gerade diese Fehlerbehandlung und die Suche nach möglichen Fehlerquellen nahm einen signifikanten Teil der Projektzeit in Anspruch. Jede Schnittstelle musste zunächst einzeln und dann im Verbund mit weiteren Programmschichten getestet werden. Daraus ergab sich dann auch eine leichte Verzögerung der Projektphase gegenüber der ursprünglichen Planung, die in anderen Projektphasen wieder eingeholt werden musste.

Zum Projekt gehört auch die Dokumentation der geleisteten Arbeit. Dies kann eine Entwickler- und/oder Benutzerdokumentation umfassen, bedeutet aber insbesondere auch die Anfertigung einer 15-seitigen Projektdokumentation, welche als Bewertungsgrundlage für die IHK dient. Sie gestaltet sich ähnlich wie wissenschaftliche Arbeiten, jedoch fallen Umfang und Komplexität entsprechend geringer aus. Die Projektdokumentation definiert das Projekt und beschreibt die einzelnen Projektphasen. In Abbildungen und Diagrammen können die verwendeten Strukturen besonders anschaulich dargestellt werden.

## FAZIT

Das Projekt hat Spaß gemacht – ich bin froh, dieses Thema gewählt zu haben. Die hohe Komplexität in Bezug auf Programmarchitektur und Schnittstellen war eine große Herausforderung und erforderte die Einarbeitung in Themengebiete, mit denen ich mich bisher nicht beschäftigt hatte. Doch durch die sorgfältige Vorbereitung während der Ausbildung und Kollegen, die mir mit Rat und Tat zur Seite standen, ließ sich jede Schwierigkeit bewältigen.

Unter den IHK-Abschlussprojekten sticht eine solche Arbeit mit Sicherheit hervor. Denn nicht jeder Azubi hat die Möglichkeit, in einer so komplexen Umgebung zu programmieren und ein Projekt auszuwählen, welches eine solch hohe Kundennähe hat. Insofern „reicht“ für die Abschlussprüfung auch ein weniger spezialisiertes Projekt. Doch der Abschluss dieses Projektes bescherte mir nicht nur einen gewissen Stolz und Selbstvertrauen in meine erlernten Fähigkeiten, sondern auch einen Punkt im Lebenslauf, der sich sicherlich sehen lassen kann. Was die berufliche Zukunft angeht, können die Azubis der GWDG mit einer solchen oder vergleichbaren Arbeit jedenfalls beruhigt ins Berufsleben starten. ■



# Tipps & Tricks

## Nutzung des Printservers gwd-winprint.top.gwdg.de unter Linux

Die GWDG stellt ihren Benutzerinstitutionen einen Printserver [gwd-winprint.top.gwdg.de](http://gwd-winprint.top.gwdg.de) zur Verfügung, über den Institutsdrucker angebunden werden können. In erster Linie ist dieser Printserver für Windows-Systeme ausgelegt, aber er kann auch unter Linux genutzt werden.

Im Folgenden soll am Beispiel von Ubuntu 16.04 anhand des Druckers HP Officejet Pro X576 (Freigabename: `smb://gwd-winprint.top.gwdg.de/GWDG-P154`) beschrieben werden, wie dieser unter Linux eingerichtet werden kann. Öffnen Sie dazu als erstes die „System Settings“ (das Symbol mit dem Schraubenschlüssel und dem Zahnrad am linken Rand) und klicken auf das Symbol „Printers“. Anschließend wird die Liste der schon eingerichteten Drucker angezeigt („Printers – localhost“). Um einen Drucker hinzuzufügen, klicken Sie auf das grüne „+“-Symbol. Sie gelangen jetzt zum „New Printer“-Dialog. Wählen Sie dort in der Spalte „Devices“ -> „Network Printer“ -> „Windows Printer via SAMBA“. In dem Textfeld hinter `smb://` geben Sie bitte den Freigabennamen Ihres Druckers (in unserem Beispiel `GWDG/gwd-winprint.top.gwdg.de/GWDG-P154`) und Ihre Anmeldedaten ein und klicken dann auf „Forward“ (siehe für sämtliche beschriebenen Schritte Abb. 1).

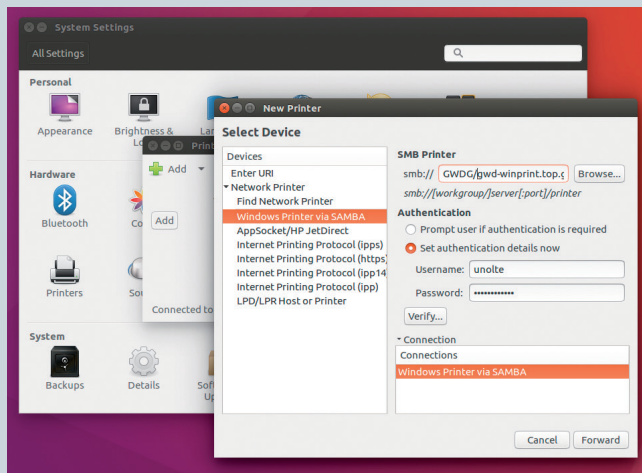


Abb. 1

Im nächsten Schritt müssen Sie einen Druckertreiber auswählen. Für unseren HP Officejet Pro X576 bringt Linux bereits einen passenden Postscript-Treiber mit (siehe Abb. 2). Wenn Sie mit dem Auswählen fertig sind, klicken Sie bitte auf „Forward“. Als letztes können Sie noch eine kurze Beschreibung Ihres Druckers eingeben (siehe Abb. 3). Jetzt sollte Ihr Drucker unter „System Settings“ -> „Printers – localhost“ angezeigt werden. Wenn Sie mit der rechten Maustaste auf den Drucker klicken und „Properties“ auswählen, bekommen Sie die Möglichkeit, eine Testseite auszudrucken. Das Ergebnis sollte wie in Abb. 4 dargestellt aussehen.

Nohte

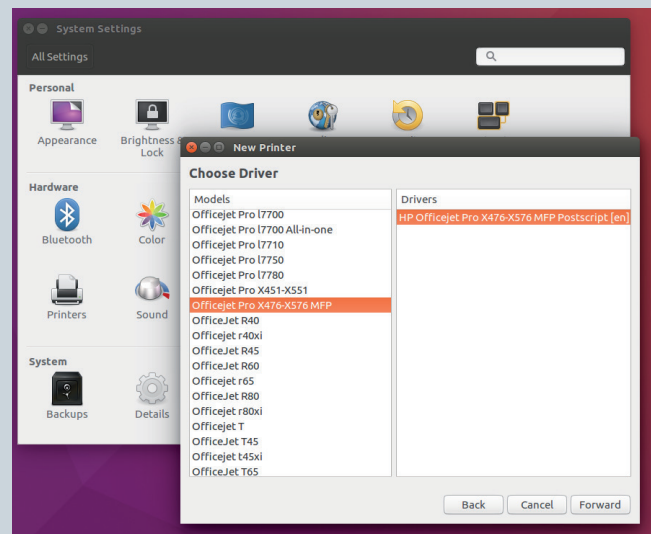


Abb. 2

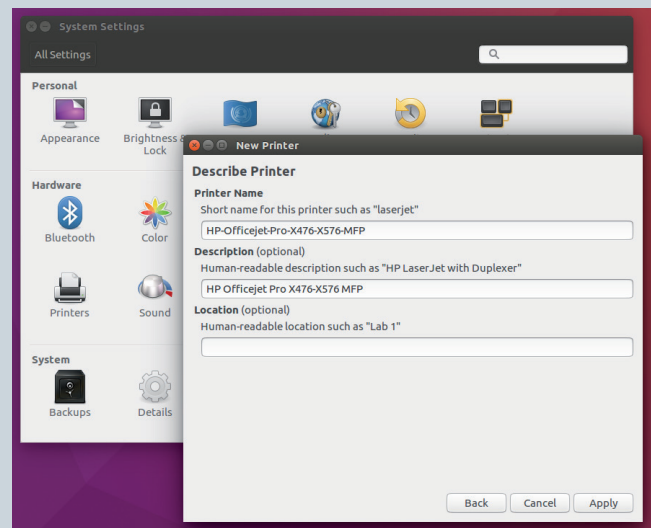


Abb. 3

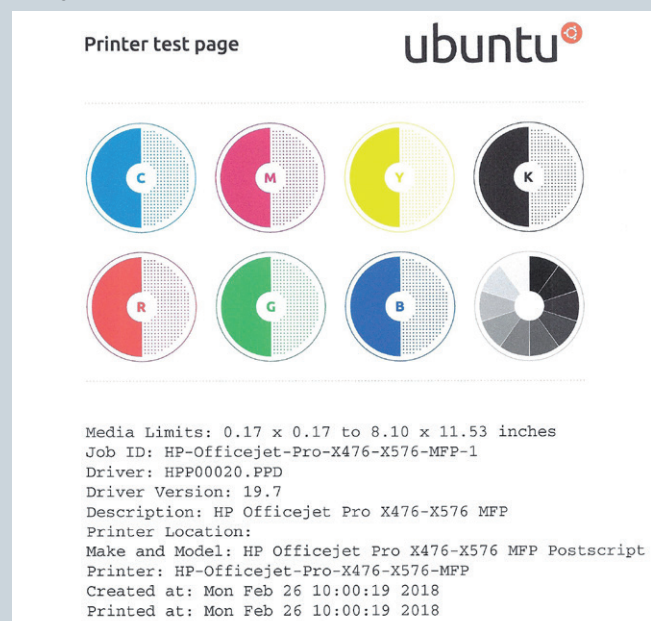


Abb. 4

# HLRN-IV in Berlin und Göttingen

## Text und Kontakt:

Dr. Christian Köhler  
christian.koehler@gwdg.de  
0551 201-2193

Der Norddeutsche Verbund für Hoch- und Höchstleistungsrechnen (HLRN-Verbund) beschafft einen neuen Supercomputer, den HLRN-IV. Dieser wird in zwei Phasen installiert und gemeinsam vom Zuse-Institut Berlin (ZIB) und der Universität Göttingen betrieben. Das System wird über eine Gesamtleistung von ca. 15 PetaFlop/s verfügen. Für das Interconnect wird die Intel Omni-Path Architecture (OPA) verwendet.

## BESCHAFFUNG UND INSTALLATION DES HLRN-IV

Wie bereits in den GWDG-Nachrichten 3/2018 und einer GWDG-News vom 07.03.2018 [1] berichtet, beschafft der Norddeutsche Verbund für Hoch- und Höchstleistungsrechnen (HLRN-Verbund) zur Unterstützung der Spitzenforschung ein neues verteiltes Supercomputer-System, den HLRN-IV. Dieser soll weiterhin

den wissenschaftlichen Bedarf der 120 Hochschulen und 171 Forschungseinrichtungen in den seit 2001 im Verbund zusammenarbeitenden Mitgliedsländern Berlin, Brandenburg (seit 2012), Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen und Schleswig-Holstein decken [2]. Betreiber sind wie bisher das Zuse-Institut Berlin (ZIB) sowie erstmals die Georg-August-Universität Göttingen. Über beide Zentren ist der HLRN-Verbund Mitglied der Gauß-Allianz. Technisch sind beide Standorte über den HLRN-Link, eine dedizierte, redundant ausgelegte Glasfaserverbindung mit einer Bandbreite von 10 Gbit/s miteinander verbunden, die der gemeinsamen technischen Betreuung des Systems dient.

Insgesamt betragen die Investitionskosten für den HLRN-IV etwa 30 Millionen Euro und die Betriebskosten liegen über einen Zeitraum von fünf Jahren bei etwa 20 Millionen Euro. Der entsprechende Vertrag wurde am 7. März 2018 unterzeichnet (siehe Abb. 1 und 2). Die Installation des neuen Rechners erfolgt durch Atos SE (Societas Europaea), einen weltweit agierenden französischen IT-Dienstleister [3]. Dazu wird im Herbst des Jahres 2018 in Phase 1 in Göttingen mit einem Übergangssystem begonnen. Im dritten Quartal 2019 erfolgt schließlich mit Phase 2 der Vollausbau an beiden Standorten, auf dessen Rechenkapazität im folgenden Abschnitt eingegangen werden soll.

## EINORDNUNG IN DIE TOP500-LISTE

Mit insgesamt 244.000 Prozessorkernen, die an den beiden Betreiberstandorten Berlin und Göttingen installiert werden, wird der HLRN-IV über eine Peak-Rechenleistung von ca. 15 PetaFlop/s verfügen, wobei Intel Xeon-CPU's der kommenden Generation Cascade Lake zum Einsatz kommen.



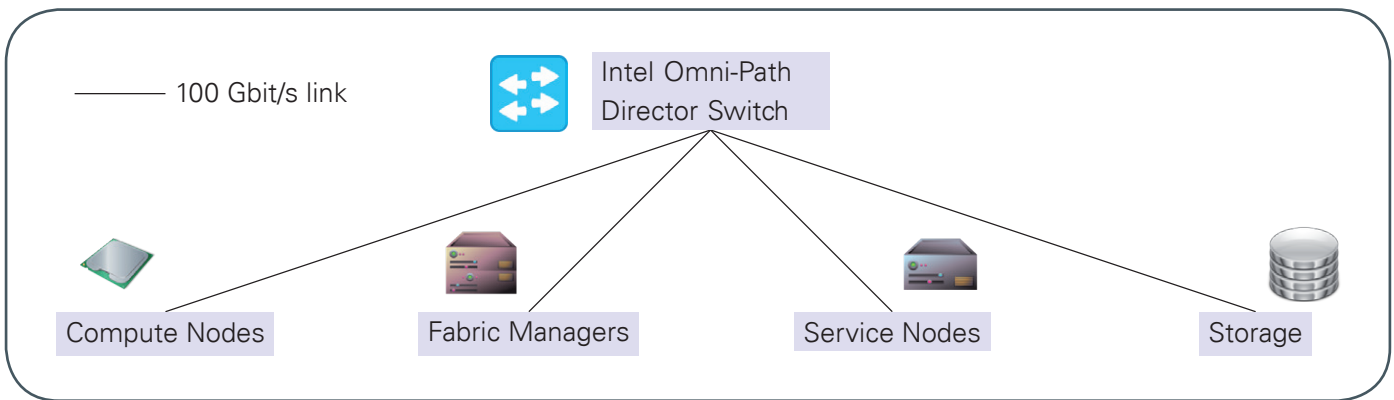
1\_Vertragsunterzeichnung am 7. März 2018 in Berlin (von links): Dr. Martin Matzke (Senior VP Atos Deutschland), Prof. Dr. Christof Schütte (Präsident des ZIB), Steffen Krach (Staatssekretär für Wissenschaft und Forschung, Senatskanzlei Berlin), Prof. Dr. Alexander Reinefeld (Leiter des Bereichs Parallele und Verteilte Systeme im ZIB)



2\_Vertragsunterzeichnung am 7. März 2018 in Göttingen (von links): Thomas Theissen (Prokurist Atos), Dr. Sabine Johannsen (Staatssekretärin im Niedersächsischen Ministerium für Wissenschaft und Kultur), Prof. Dr. Ulrike Beisiegel (Präsidentin der Universität Göttingen)

## HLRN-IV in Berlin and Göttingen

The North-German Supercomputing Alliance (Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen – HLRN) procures a new supercomputer, the HLRN-IV. The system will be installed in two phases and operated by the Zuse Institute Berlin (ZIB) and the University of Göttingen. The system will feature a combined performance of approx. 15 PetaFlop/s. For the interconnect Intel Omni-Path Architecture (OPA) is used.



3\_Anbindung der Rechenknoten, des Filesystems und weiterer Infrastrukturkomponenten an das gemeinsame Intel Omni-Path Architecture (OPA) Interconnect

Zur Einordnung dieses Wertes hilft ein Blick auf die aktuelle Top500-Liste der weltweit schnellsten Computersysteme [4], in der man einen Rechner mit dieser Peakleistung derzeit auf Platz 10 erwarten würde. Dabei ist jedoch zu beachten, dass die Sortierung hier grundsätzlich nach der im LINPACK-Benchmark (Lösung linearer Gleichungssysteme) erzielten Leistung erfolgt, für deren Kommunikationsbedarf eine Aufteilung auf die zwei Standorte nicht optimal wäre. Weiterhin kann die Messung erst nach Beginn der zweiten Aufbauphase erfolgen und mit der dann aktuellen Top500-Liste verglichen werden.

## NEUES INTERCONNECT: INTEL OMNI-PATH

Als Cluster Interconnect, das lokale Kommunikationsnetzwerk für Rechenjobs, die über mehrere Knoten verteilt sind, kommt an beiden Standorten die Intel Omni-Path Architecture (OPA) zur Anwendung [5]. Auch Serviceknoten und das Filesystem sind über Intel OPA 100 Gbit/s Host Fabric Interfaces an das Netzwerk angebunden (siehe Abb. 3). Durch die verwendete Fat-Tree-Topologie wird eine Gesamtbandbreite von ca. 25 GByte/s zwischen je zwei Knoten erzielt.

Ein besonderes Merkmal dieser Netzwerkarchitektur ist die Übertragung der zwischen den Knoten zu versendenden Fabric Packets (FP), die bei einem Vergleich mit dem klassischen OSI-Modell am ehesten Layer 2-Paketen entsprechen [6]. Statt diese individuell mit Fehlerkorrekturinformationen zu versehen, werden zuvor jeweils 64 Bits als Flow Control Digits (FLIT) von 16 (möglicherweise verschiedenen) FPs zu einem Link Transfer Packet (LTP) kombiniert. Die zugehörige Netzwerkschicht wird bei Intel dementsprechend als Layer 1,5 bezeichnet. Ein Fehler im 14 Bit großen CRC-Prüfwert eines solchen Pakets würde zwar zu einer Neuübertragung der FLITs aus allen beteiligten FPs führen, dies kann jedoch auf jedem Teilstück der Route, bspw. zwischen Knoten und Switch oder zwischen zwei Switches, erfolgen. Hierdurch kann im Vergleich zur Vorwärtsfehlerkorrektur mit eventueller Ende-zu-Ende-Neuübertragung die für verteilte Berechnungen besonders

kritische Latenz gesenkt werden.

Aufgrund der direkten Einbindung des Filesystems in die OPA-Fabric muss die mit der Latenz konkurrierende Anforderung nach hoher I/O-Bandbreite ebenfalls berücksichtigt werden. Die von Intel so bezeichnete Traffic Flow Optimization (TFO) bewerkstelligt dies, indem FPs hoher Priorität bevorzugt im nächsten LTP berücksichtigt werden können. Dazu lassen sich sog. Service Level definieren, deren Abbildung auf die im Paket enthaltenen Steuerungsbits vom Fabric Manager übernommen werden. Im Falle einer Überlastung des Netzwerks (Congestion) schaltet dieser den verwendeten Routing-Algorithmus um vom standardmäßig aktiven Dispersive Routing, welches zu Lastverteilungszwecken die Übertragung zwischen zwei Knoten über mehrere redundante Verbindungen realisiert, hin zum Adaptive Routing, das versucht, jederzeit die am wenigsten ausgelastete Verbindung zu wählen. Dies geschieht aus Sicht der höheren Netzwerkschichten transparent, insbesondere muss nicht manuell durch den Nutzer eingegriffen werden.

## LINKS

- [1] GWDG-News „Mehr Rechenleistung für Norddeutschland“ vom 07.03.2018: <https://info.gwdg.de/news/mehr-rechenleistung-fuer-norddeutschland/>
- [2] HLRN – Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen: <https://www.hlrn.de/>
- [3] Atos SE – Company Profile: <https://atos.net/en/about-us/company-profile>
- [4] Top500 List – November 2017: <https://www.top500.org/list/2017/11/>
- [5] Intel Omni-Path Architecture: <https://www.intel.de/content/www/de/de/high-performance-computing-fabrics/omni-path-driving-exascale-computing.html>
- [6] Exploring Intel's Omni-Path Network Fabric: <https://www.anandtech.com/show/9561/exploring-intels-omnipath-network-fabric>

# Kernkraftwerke bei der Arbeit

## Text und Kontakt:

Thorsten Hindermann  
thorsten.hindermann@gwdg.de  
0551 201-1837

Dieser Artikel beschäftigt sich mit der Fragestellung, ob mit Hilfe des OpenCL-Standards die Möglichkeit besteht, auf Quellzeilen-Ebene plattformübergreifende GPU-Anwendungen zu entwickeln. Exemplarisch wird ein einfaches Programm verwendet, um die Übersichtlichkeit zu bewahren. Der Weg soll von Mac OS X über Windows 10 hin zum HPC-Cluster der GWDG führen. Ob und wie steinig dieser Weg sein wird, klärt sich in den folgenden Zeilen.

## WIE OFFEN...

Im Normalfall sind heutige Anwendungen so programmiert, dass diese die zentrale Prozessoreinheit (engl. Central Processing Unit, kurz **CPU**) nutzen, sowohl für die Ausführung der Anwendung als auch für, teilweise auch komplexe, Berechnungen in der Anwendung.

In jedem heutzutage gekauften Rechner stecken gerade für die Berechnung von komplexen Sachverhalten enorm leistungsfähige Prozessoreinheiten. Die Rede ist hier von der Grafikkarte. Diese besitzt mehr Prozessoreinheiten (engl. Graphics Processing Unit, kurz **GPU**) und darin enthaltene Kerne als die o. g. CPUs. Die Grafikkarten sind sehr stark für Berechnungen optimiert und somit richtige „Kraftprotze“. Ihre Hauptanwendung besteht in der (Vor-) Berechnung von Bildern für heutige sehr grafiklastige Spiele.

Aber die Grafikkarten-Hersteller haben einen Weg gefunden, die Leistungsfähigkeit dieser „Kernkraftwerke“ auch für Anwendungen zur Verfügung zu stellen, die auch komplexe Berechnungen durchführen. Jeder Hersteller hat hierfür seinen eigenen Standard definiert. Die Grafikkartenhersteller haben sich aber auch zusammengeschlossen und einen offenen Standard entworfen und jeder Hersteller unterstützt diesen auch mit entsprechendem Anwendungsentwicklungs-Rahmenwerk (engl. Software Development Kit, kurz **SDK**). Der Standard heißt **OpenCL** und steht für **Open Computing Language**. Ein guter Einstiegspunkt für OpenCL ist sicherlich der URL <https://www.khronos.org/opencl/>.

## IST OPENCL...

Die einfachste, schnellste und unkomplizierteste Art und Weise, in das Thema OpenCL praktisch einzusteigen, ist mit Hilfe von Apples Mac OS X. Apple stellt seine Systeme so zur Verfügung, dass auf jedem neueren System gleich ein lauffähiges OpenCL-Projekt übersetzt und ausgeführt werden kann. Zusätzlich notwendig ist Apples eigene, speziell für Mac OS X abgestimmte Entwicklungsumgebung (engl. Integrated Development Environment, kurz **IDE**) **Xcode**.

Das Beispielprojekt, das den Ausgangspunkt für diesen Artikel bildet, ist unter dem URL <https://developer.apple.com/library/>

[content/samplecode/OpenCL\\_Hello\\_World\\_Example/Introduction/Intro.html](https://developer.apple.com/library/content/samplecode/OpenCL_Hello_World_Example/Introduction/Intro.html) mit dem Titel „OpenCL Hello World Example“ zu erreichen.

Allgemeine Informationen zum Thema OpenCL, speziell für Mac OS X, finden sich unter dem URL <https://developer.apple.com/opencl/>.

Mittels der Schaltervariable in Zeile 86 der Quellzeilen-Auflistung (siehe S. 19) kann entschieden werden, ob die Berechnung normal in der zentralen Prozessoreinheit durchgeführt werden soll oder in den GPUs der Grafikkarte:

```
int gpu = 1; //GPU = 1; CPU = 0
```

## WIRKLICH?

Nach diesem geglückten Start in die OpenCL-Welt unter Mac OS X wenden wir uns dem Umzug dieses einfachen OpenCL-Programms in die OpenCL-Welt unter Windows 10 zu. Hier gehen die ersten Schwierigkeiten schon los und das Schlagwort „Open“ beginnt zu bröckeln.

Zuerst muss der Benutzer des Rechners ermitteln, welche Grafikkarte in seinem ihm zur Verfügung stehenden Computer-System eingebaut wurde. Hier muss der ungeübte Benutzer sicherlich die Hilfe der IT-Administratoren des eigenen Instituts in Anspruch nehmen. Wenn die Frage des Grafikkarten-Herstellers geklärt ist, besteht der nächste Schritt darin, für die entsprechende Grafikkarte die passende OpenCL-SDK zu installieren.

## Development of GPU Applications with OpenCL

This article deals with the question of whether the OpenCL standard can be used to develop cross-platform GPU applications at the source line level. For example, a simple program is used to maintain clarity. The way should lead from Mac OS X via Windows 10 to the HPC cluster of the GWDG. The following lines clarify whether and how stony this path will be.

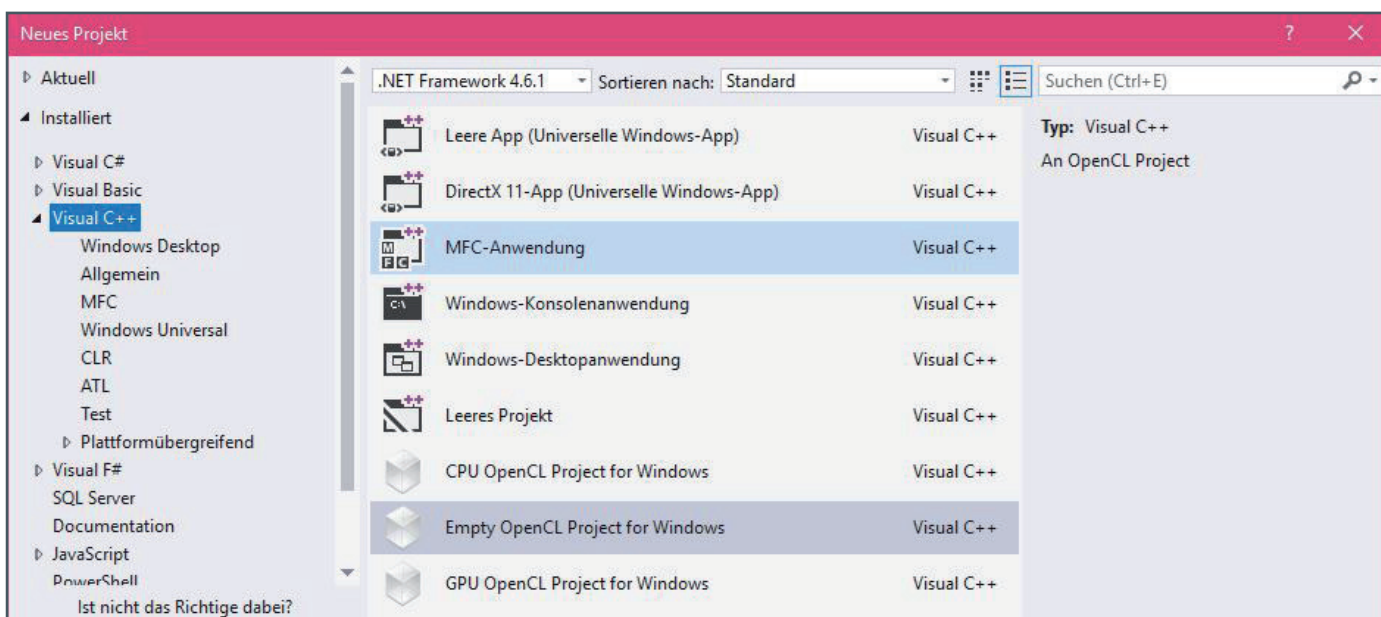


Abb. 1

Im vorliegenden Beispiel ist in dem verwendeten Laptop Dell Latitude E7470 eine Grafikkarte des Herstellers Intel verbaut. Dafür wurde die passende OpenCL-Version vom URL <https://software.intel.com/en-us/intel-opencl/download> heruntergeladen. Namentlich heißt das Paket **Intel® SDK for OpenCL™ applications**. Das Setup beinhaltet auch eine Integration in die IDE **Microsoft Visual Studio 2017**. Nach erfolgter Installation unter Visual Studio ist ein leeres OpenCL-Projekt zu erstellen (siehe Abb. 1).

In dem Projekt befindet sich eine leere Quellzeilen-Datei mit dem Namen *host.cpp*. In diese Datei sind die Quellzeilen aus *hello.c* des Mac OS X OpenCL-Projekts zu kopieren. Ein erstes Übersetzen (engl. to compile) dieses neuen Projektes hat zur Folge, dass Fehler auftreten und sich das Projekt nicht so einfach übersetzen lässt.

Nun muss die Möglichkeit des C/C++-Präprozessors ausgenutzt werden und mit bedingten Definitionen (engl. conditional defines) gearbeitet werden. Dazu wird, wie in Zeile 1 der Quellzeilen-Auflistung zu sehen ist, die folgende Definition gesetzt:

```
#define W10OCL
```

Alle Abweichungen von der Mac OS X OpenCL-Quellzeilen-Datei zur Datei für die Intel OpenCL-Quellzeilen-Datei werden mit folgendem Konstrukt umgeben:

```
#ifdef W10OCL
    ...
#else
    ...
#endif
```

Diese Vorgehensweise wird in den Zeilen 9 – 15, 50 – 53, 61 – 84 und 87 – 91 der Quellzeilen-Auflistung deutlich. Das bedeutet, dass alle Quellzeilen-Teile, die der Definition *W10OCL* gleich WAHR entsprechen, übersetzt werden. Ab diesem Zeitpunkt lässt sich die so veränderte Quellzeilen-Datei unter Windows 10 mit

einer Intel-Grafikkarte übersetzen.

## OPENCL UND HPC

Aus der vorherigen Vorgehensweise ließe sich Folgendes ableiten: Mit Hilfe von OpenCL ist es möglich, bestimmte rechenintensive Berechnungsverfahren erst einmal am eigenen Arbeitsplatzsystem mittels OpenCL zu entwickeln. Ist diese Entwicklung abgeschlossen, ist der nächste Schritt, dieses nun auf ein HPC-System zu überführen und die intensive Berechnung größerer bis großer Datenmengen (Stichwort Big Data) dann mit Hilfe dieser sogenannten HPC-Cluster (engl. High Performance Computing, kurz HPC) durchzuführen, die ja auch die GWGD zur Verfügung stellt.

Mit Hilfe des vereinheitlichten Dateisystems bei der GWGD kann nun die reine Quellzeilen-Datei *host.cpp* dorthin kopiert werden. Wenn das Windows-System ins Active Directory der GWGD integriert ist, sollte dem Benutzer das persönliche Laufwerk *P:* zur Verfügung stehen.

Nun mittels einer UNIX-Shell und zuvor gestartetem X-Server den folgenden Befehl absetzen (Den Account-Namen, im vorliegenden Beispiel *thinder*, entsprechend austauschen):

```
ssh -X thinder@gwdu101.gwdg.de
```

Neben der *gwdu101* stehen noch die *gwdu102* und *gwdu103* zur Verfügung. Jetzt die passende bzw. aktuelle CUDA-Umgebung laden. Dies wird mittels folgendem Befehl durchgeführt:

```
module load cuda90
```

Jetzt kann die CUDA IDE **Nsight** mit diesem Befehl aufgerufen werden:

```
nsight &
```

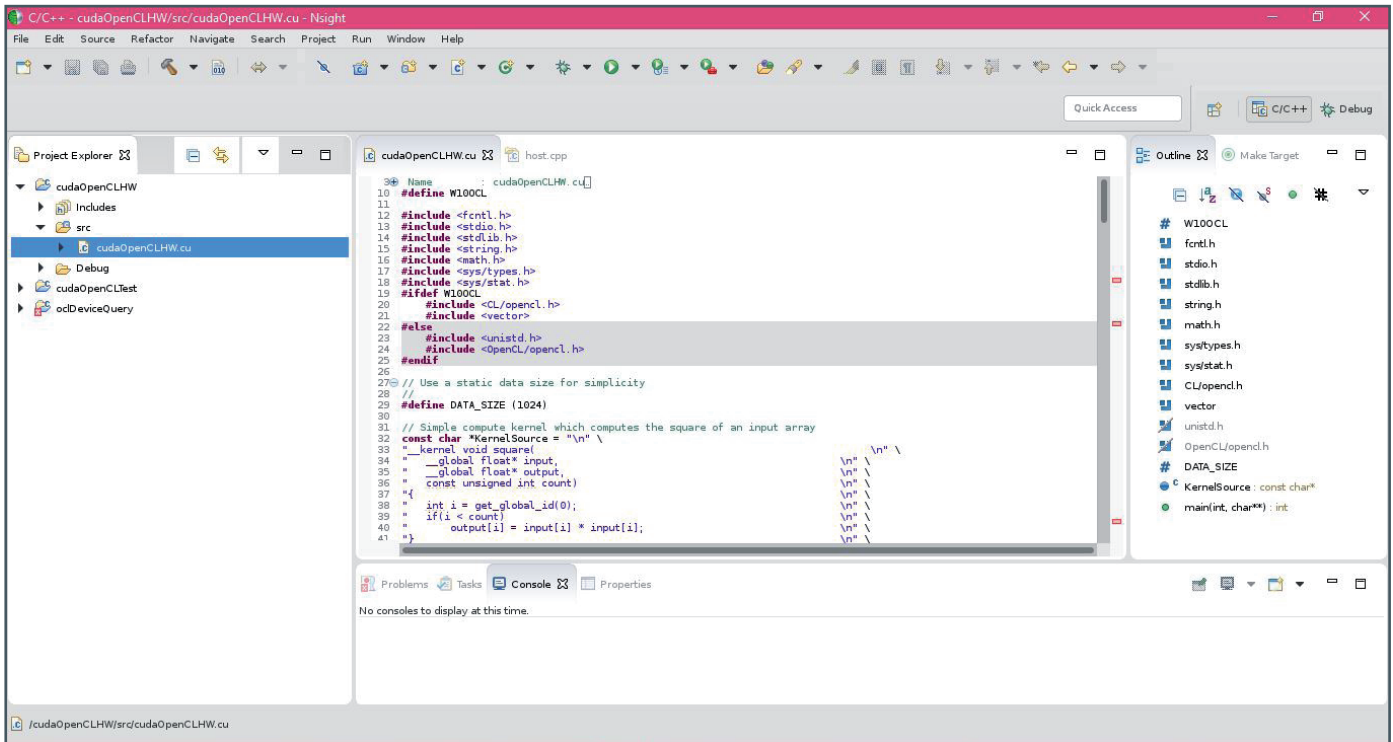


Abb. 2

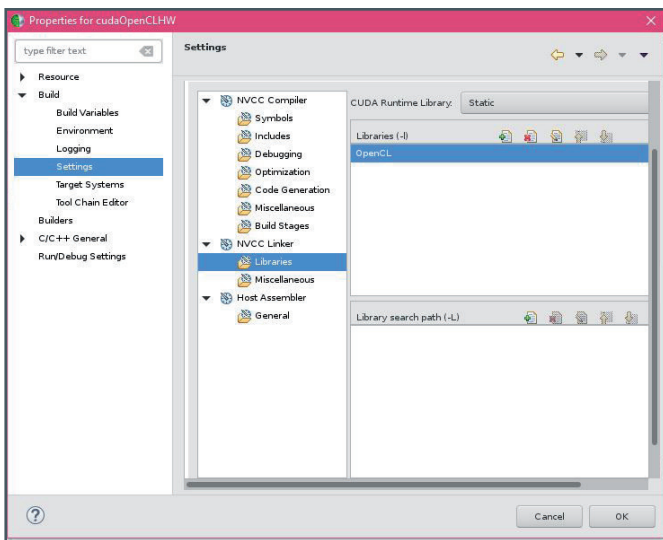


Abb. 3

Mittels „File -> New -> CUDA C/C++ Project“ ein neues Projekt anlegen, die Quellzeilen-Datei *host.cpp* öffnen und mittels „Kopieren und Einfügen“ die Quellzeilen in dieses Projekt übertragen. Das Ergebnis ist in Abb. 2 zu sehen. In den Projekteigenschaften im Bereich des NVCC Linkers noch den Eintrag *OpenCL* hinzufügen (siehe Abb. 3). So lässt sich dann das OpenCL-Projekt auf den interaktiven Rechnern der HPC-Plattform übersetzen.

Um das gerade erfolgreich übersetzte Programm zu testen,

eine entsprechende Kommandozeile mittels folgendem Befehl auf dem HPC-Cluster aufrufen:

```
bsub -ISs -q gpu -n 1 -R "rusage[ngpus_shared=1]/bin/bash
```

In das Projekt-Verzeichnis wechseln. Im vorliegenden Beispiel ist das der Pfad:

```
cd ~/cuda-workspace/cudaOpenCLHW/Debug
```

Und hier das Programm wie folgt aufrufen:

```
./cudaOpenCLHW
```

## AUSBLICK

Bisher waren die jetzigen Testläufe des Programms leider noch nicht erfolgreich. Sie brachen immer wieder mit einem Segment-Zugriffsfehler (engl. Segmentation Fault) ab. Die genaue Analyse der Ursache für diese Programmabbrüche läuft noch. Falls das OpenCL-Programm auf der HPC-Plattform der GWDG zum Laufen gebracht werden kann, wird hierüber in einem entsprechenden Folgeartikel in den GWDG-Nachrichten berichtet. ■

## QUELLZEILEN-AUFLISTUNG

```

1 #define W100CL
2 #include <fcntl.h>
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6 #include <math.h>
7 #include <sys/types.h>
8 #include <sys/stat.h>
9 #ifdef W100CL
10 #include <CL/opencl.h>
11 #include <vector>
12 #else
13 #include <unistd.h>
14 #include <OpenCL/opencl.h>
15 #endif
16 // Use a static data size for simplicity
17 #define DATA_SIZE (1024)
18 // Simple compute kernel which computes the square of an input array
19 const char *KernelSource = "\n" \
20 "    _kernel void square(                \n" \
21 "        __global float* input,         \n" \
22 "        __global float* output,       \n" \
23 "        const unsigned int count)     \n" \
24 "{                                     \n" \
25 "    int i = get_global_id(0);          \n" \
26 "    if(i < count)                     \n" \
27 "        output[i] = input[i] * input[i]; \n" \
28 "}" \
29 "\n";
30
31 int main(int argc, char** argv)
32 {
33     int err; // error code returned from api calls
34
35     float data[DATA_SIZE]; // original data set given to device
36     float results[DATA_SIZE]; // results returned from device
37     unsigned int correct; // number of correct results returned
38
39     size_t global; // global domain size for our calculation
40     size_t local; // local domain size for our calculation
41
42     cl_device_id device_id; // compute device id
43     cl_context context; // compute context
44     cl_command_queue commands; // compute command queue
45     cl_program program; // compute program
46     cl_kernel kernel; // compute kernel
47
48     cl_mem input; // device memory used for the input array
49     cl_mem output; // device memory used for the output array
50 #ifdef W100CL
51     cl_uint numPlatforms = 0;
52     cl_uint numDevices = 0;
53 #endif
54
55     // Fill our data set with random float values
56     int i = 0;
57     unsigned int count = DATA_SIZE;
58     for (i = 0; i < count; i++)
59         data[i] = rand() / (float)RAND_MAX;
60
61 #ifdef W100CL
62     // Get (in numPlatforms) the number of OpenCL platforms available
63     // No platform ID will be returned, since platforms is NULL
64     err = clGetPlatformIDs(0, NULL, &numPlatforms);
65     if (err != CL_SUCCESS)
66     {
67         printf("Error: Failed to get num platforms returned!\n");
68         return EXIT_FAILURE;
69     }
70     if (0 == numPlatforms)
71     {
72         printf("Error: No platforms found!\n");
73         return EXIT_FAILURE;
74     }
75     std::vector<cl_platform_id> platforms(numPlatforms);
76     // Now, obtains a list of numPlatforms OpenCL platforms available
77     // The list of platforms available will be returned in platforms
78     err = clGetPlatformIDs(numPlatforms, &platforms[0], NULL);
79     if (err != CL_SUCCESS)
80     {
81         printf("Error: Failed to get platforms returned!\n");
82         return EXIT_FAILURE;
83     }
84 #endif
85     // Connect to a compute device
86     int gpu = 1; //GPU = 1; CPU = 0
87 #ifdef W100CL
88     err = clGetDeviceIDs(platforms[1], gpu ? CL_DEVICE_TYPE_GPU :
89         CL_DEVICE_TYPE_CPU, 1, &device_id, &numDevices);
90 #else
91     err = clGetDeviceIDs(NULL, gpu ? CL_DEVICE_TYPE_GPU : CL_DEVICE_TYPE_CPU, 1,
92         &device_id, NULL);
93 #endif
94 #endif
95     if (err != CL_SUCCESS)
96     {
97         printf("Error: Failed to create a device group!\n");
98         return EXIT_FAILURE;
99     }
100     // Create a compute context
101     context = clCreateContext(0, 1, &device_id, NULL, NULL, &err);
102     if (!context)
103     {
104         printf("Error: Failed to create a compute context!\n");

```

```

102         return EXIT_FAILURE;
103     }
104     // Create a command commands
105     commands = clCreateCommandQueue(context, device_id, 0, &err);
106     if (!commands)
107     {
108         printf("Error: Failed to create a command commands!\n");
109         return EXIT_FAILURE;
110     }
111     // Create the compute program from the source buffer
112     program = clCreateProgramWithSource(context, 1, (const char **)&
113         KernelSource, NULL, &err);
114     if (!program)
115     {
116         printf("Error: Failed to create compute program!\n");
117         return EXIT_FAILURE;
118     }
119     // Build the program executable
120     err = clBuildProgram(program, 0, NULL, NULL, NULL);
121     if (err != CL_SUCCESS)
122     {
123         size_t len;
124         char buffer[2048];
125         printf("Error: Failed to build program executable!\n");
126         clGetProgramBuildInfo(program, device_id, CL_PROGRAM_BUILD_LOG, sizeof
127             (buffer), buffer, &len);
128         printf("%s\n", buffer);
129         exit(1);
130     }
131     // Create the compute kernel in the program we wish to run
132     kernel = clCreateKernel(program, "square", &err);
133     if (!kernel || err != CL_SUCCESS)
134     {
135         printf("Error: Failed to create compute kernel!\n");
136         exit(1);
137     }
138     // Create the input and output arrays in device memory for our calculation
139     input = clCreateBuffer(context, CL_MEM_READ_ONLY, sizeof(float) * count,
140         NULL, NULL);
141     output = clCreateBuffer(context, CL_MEM_WRITE_ONLY, sizeof(float) * count,
142         NULL, NULL);
143     if (!input || !output)
144     {
145         printf("Error: Failed to allocate device memory!\n");
146         exit(1);
147     }
148     // Write our data set into the input array in device memory
149     err = clEnqueueWriteBuffer(commands, input, CL_TRUE, 0, sizeof(float) *
150         count, data, 0, NULL, NULL);
151     if (err != CL_SUCCESS)
152     {
153         printf("Error: Failed to write to source array!\n");
154         exit(1);
155     }
156     // Set the arguments to our compute kernel
157     err = 0;
158     err = clSetKernelArg(kernel, 0, sizeof(cl_mem), &input);
159     err |= clSetKernelArg(kernel, 1, sizeof(cl_mem), &output);
160     err |= clSetKernelArg(kernel, 2, sizeof(unsigned int), &count);
161     if (err != CL_SUCCESS)
162     {
163         printf("Error: Failed to set kernel arguments! %d\n", err);
164         exit(1);
165     }
166     // Get the maximum work group size for executing the kernel on the device
167     err = clGetKernelWorkGroupInfo(kernel, device_id, CL_KERNEL_WORK_GROUP_SIZE,
168         sizeof(local), &local, NULL);
169     if (err != CL_SUCCESS)
170     {
171         printf("Error: Failed to retrieve kernel work group info! %d\n", err);
172         exit(1);
173     }
174     // Execute the kernel over the entire range of our 1d input data set
175     // using the maximum number of work group items for this device
176     global = count;
177     err = clEnqueueNDRangeKernel(commands, kernel, 1, NULL, &global, &local, 0,
178         NULL, NULL);
179     if (err)
180     {
181         printf("Error: Failed to execute kernel!\n");
182         return EXIT_FAILURE;
183     }
184     // Wait for the command commands to get serviced before reading back results
185     clFinish(commands);
186     // Read back the results from the device to verify the output
187     err = clEnqueueReadBuffer(commands, output, CL_TRUE, 0, sizeof(float) *
188         count, results, 0, NULL, NULL);
189     if (err != CL_SUCCESS)
190     {
191         printf("Error: Failed to read output array! %d\n", err);
192         exit(1);
193     }
194     // Validate our results
195     correct = 0;
196     for (i = 0; i < count; i++)
197     {
198         if (results[i] == data[i] * data[i])
199             correct++;
200     }
201     // Print a brief summary detailing the results
202     printf("Computed '%d/%d' correct values!\n", correct, count);
203     // Shutdown and cleanup
204     clReleaseMemObject(input);
205     clReleaseMemObject(output);
206     clReleaseProgram(program);
207     clReleaseKernel(kernel);
208     clReleaseCommandQueue(commands);
209     clReleaseContext(context);
210     return 0;
211 }

```

# Kurz & knapp

## Öffnungszeiten des Rechenzentrums am Maifeiertag und an Christi Himmelfahrt

Das Rechenzentrum der GWDG ist am Maifeiertag, 01.05.2018, und an Christi Himmelfahrt, 10.05.2018, 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](mailto: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

## Treffen des ZKI-Arbeitskreises „Zentrale Systeme“ in Göttingen

Auch in diesem Jahr fand das Jahrestreffen des ZKI-Arbeitskreises „Zentrale Systeme“ (<https://www.zki.de/arbeitskreise/zentrale-systeme/>) wieder am Max-Planck-Campus in Göttingen statt. Vom 06. bis zum 09.03.2018 wurde ausgiebig über die Veränderungen seit dem letzten Treffen und die kommenden Herausforderungen im Bereich der zentralen Systeme, also vor allem Server und Speichersysteme, diskutiert. Hierbei wurde vom üblichen Rahmen mit Sitereports und anschließender Diskussion abgewichen und stattdessen das Verfahren des „BarCamps“ ausprobiert. Als Abwandlung wurden einige Site-reports als Ideengeber genutzt und aus diesen Ideen dann die Reihenfolge für die nächsten Diskussionen bestimmt.

Das Thema „Proprietäre Software und wie löst man sich davon?“ ist offensichtlich noch immer aktuell. Insbesondere die

Frage, wie man für akademische Nutzer eine E-Mail- und Groupware-Umgebung betreibt, ist nicht einfach zu beantworten, da in der Praxis die als „Allheilmittel“ beworbenen kommerziellen Lösungen nicht alle Anforderungen abdecken können.

Ebenso spannend ist die Frage nach der Zukunft der Speichertechnologien. Hier ist in den Hochschulrechenzentren eine Abkehr vom bzw. Reduzierung des Speichernetzes (SAN) zu beobachten. Object Storage scheint diese Lücke zu füllen. Auch Filer entwickeln sich wohl in Richtung einer (großen) Nische, für die Object Storage nicht geeignet ist. In diesen Themenbereich gehören natürlich auch die Fragen, wie mit den zunehmenden Datenmengen umgegangen werden soll und wo bekannte Datensicherungskonzepte nicht mehr anwendbar sind.

Im Bereich der Servervirtualisierung sind zahlreiche landesweite Initiativen zu beobachten: Während teilweise noch konzeptioniert wird, sind die „Cloud Server“ bereits in NRW (WWU Münster), Baden-Württemberg (bwCloud) und auch bei der GWDG (GWDG Cloud Server) im Test- oder sogar schon Produktivbetrieb. Dennoch besteht weiter großer Bedarf bei der Abstimmung und beim Erfahrungsaustausch. So sollen im laufenden Jahr noch ein Workshop für Betreiber mit dem Fokus Betriebskonzept veranstaltet und in einem zweiten Workshop Hilfestellungen für Interessenten weitergegeben werden.

Den Abschluss der Jahrestagung bildete ein Workshop zum Thema „Docker“.

Nachdem die GWDG nun zum dritten Mal das Jahrestreffen ausgerichtet hat, wird 2019 die TU Dresden diese Aufgabe übernehmen.

Im Rahmen des Jahrestreffens standen turnusgemäß die Wahlen des Sprechers und seiner Stellvertreter an. Die beiden „Amtsinhaber“ wurden für die nächsten zwei Jahre wiedergewählt; Henrik Bloch von der TU Braunschweig stößt als weiterer Stellvertreter ins Team hinzu.

Nachtwey



INFORMATIONEN:  
support@gwdg.de  
0551 201-1523

April bis  
Dezember 2018

# Kurse



KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
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
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 GWGD	Quentin	24.05.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	17.05.2018	4

KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
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 VORKENNTNISSEN – 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 BIBDBEARBEITUNG 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
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

KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
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](mailto: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](mailto: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