

GWDG
NACHRICHTEN
08-09|18

PostgreSQL

Microsoft PowerShell

International Summer
School on Data Science

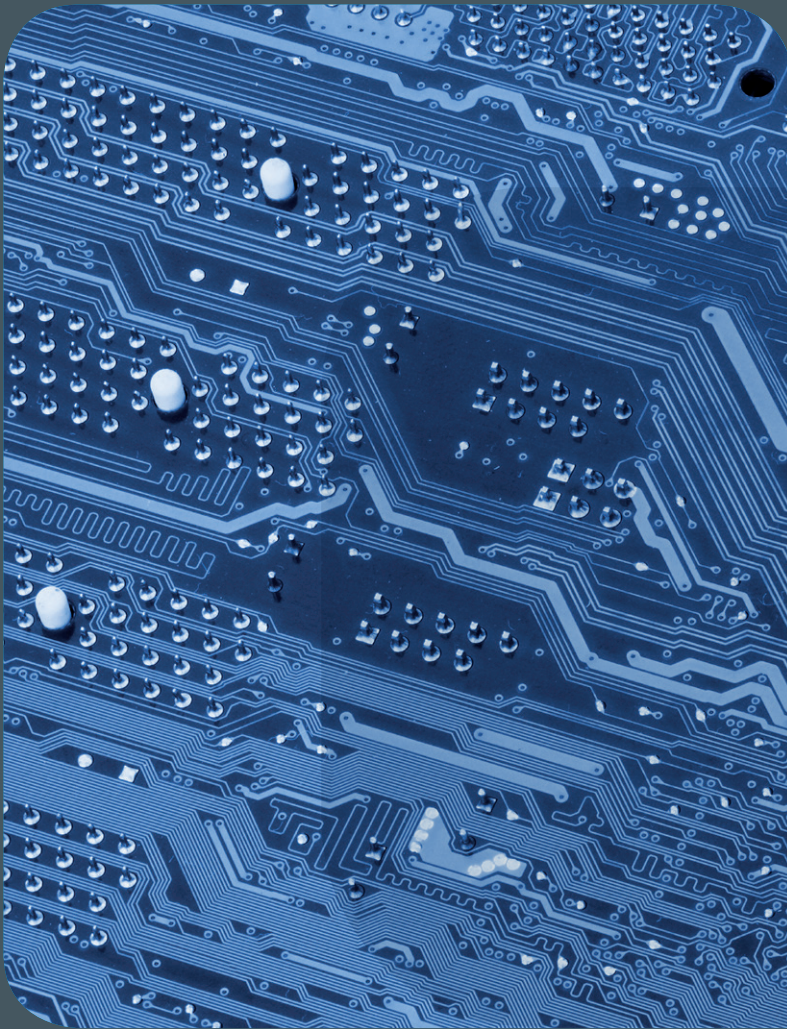
EU-Projekt
„MIKELANGELO“

EU-Projekt „NEPHELE“

ZEITSCHRIFT FÜR DIE KUNDEN DER GWDG



Database



GWDG NACHRICHTEN

08-09|18

Inhalt

.....

4 Neuer Dienst „PostgreSQL“ 6 Die Kraftmuschel selber erweitern – Programmierübung für Auszubildende der Fachinformatik

10 International Summer School on Data Science 2018 12 Kurz & knapp 13 Completion of the MIKELANGELO Project 16 Tipps & Tricks

17 Ending of the EU Horizon 2020 Project “NEPHELE” 20 Personalia 22 Kurse

Impressum

.....

Zeitschrift für die Kunden der GWDG

ISSN 0940-4686
41. Jahrgang
Ausgabe 8-9/2018

Erscheinungsweise:
monatlich

www.gwdg.de/gwdg-nr

Auflage:
550

Fotos:
© Maksim Kabakou - Fotolia.com (1)
© vege - Fotolia.com (5)
© goodluz - Fotolia.com (6)
© Peter Heller (10, 11)
© MPLbpc-Medienservice (3, 20, 21)
© GWDG (2, 22)
© MIKELANGELO Project (15)
© NEPHELE Project Website (18)

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,

um innovative Lösungen voranzutreiben und am Puls der Zeit zu bleiben, betreibt die GWDG erfolgreich eigene Forschung im Umfeld von eScience. Aus zahlreichen Drittmittelprojekten gefördert sind im Durchschnitt 30 – 40 Wissenschaftlerinnen und Wissenschaftler beschäftigt: viele zur Promotion oder im Rahmen eines PostDoc-Programms. In den GWDG-Nachrichten berichten wir daher immer wieder über unsere vielfältigen Forschungsprojekte. Diese erlauben es uns, neue Themen jenseits der Service-Aufgaben eines Rechenzentrums zu betrachten und innovative Lösungen frühzeitig auch in unsere Betriebsstrukturen zu integrieren.

In dieser Ausgabe finden Sie mit den Projekten MIKELANGELO und NEPHELE zwei typische Beispiele hierfür – beides aus dem EU-Rahmenprogramm Horizon 2020 geförderte größere Verbundprojekte. In dem einen Projekt geht es um die Optimierung der Kommunikationsschnittstellen in virtuellen Maschinen für I/O-intensive Anwendungen im Bereich Big Data, Cloud und High Performance Computing, in dem anderen um die Optimierung in neuen optischen SDN-Netzwerkstrukturen. Beides sind Projekte, die hohe Relevanz im Betrieb von großen Rechnerinfrastrukturen haben, wie wir sie bei der GWDG betreiben, und die damit Potenzial haben, um die ständige Weiterentwicklung unserer Systeme zu beeinflussen.

Ich wünsche Ihnen viel Freude beim Lesen dieser Ausgabe.

Ramin Yahyapour

GWDG – IT in der Wissenschaft

Neuer Dienst „PostgreSQL“

Text und Kontakt:

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

Seit dem 1. August 2018 bietet die GWDG ihren Nutzern auch PostgreSQL als weiteren Datenbankdienst neben OracleSQL und MySQL an. OracleSQL ist für große und komplexe Datenbanken gut geeignet, MySQL für Webanwendungen. Nun steht auch PostgreSQL mit anspruchsvollen Statements, speziell in der Bioinformatik und bei Geolokalisationsdaten sinnvoll, zur Verfügung. Neben bekannten Erweiterungen wie PostBIS oder PostGIS stehen noch viele weitere zur Verfügung.

EINSATZMÖGLICHKEITEN

OracleSQL kommt bei großen komplexen Datenbanken zum Einsatz, bei denen es auf Geschwindigkeit, aber nicht so sehr auf die Kosten ankommt. MySQL hat als lizenzfreie und somit günstige Lösung seine Stärken im Webseitenbereich. Dazu bietet die GWDG für MySQL einen hochverfügbaren Dienst für ihre Kunden an, welcher die Möglichkeiten der im Webbereich weitverbreitetsten Datenbank-Engine mit der Geschwindigkeit und Hochverfügbarkeit anderer Lösungen vereint. Nun steht auch PostgreSQL zur Verfügung. Gerade durch seine ausgeklügelten Statements bieten sich mehr Möglichkeiten an, als dies beispielsweise mit MySQL möglich wäre. Einige Beispiele werden weiter unten gezeigt.

Insbesondere in der Bioinformatik wird PostgreSQL häufig eingesetzt, aber auch bei Geolokalisationsdaten. Der Grund dafür ist, dass dort viele Entwickler in diesem Bereich passende und nützliche Erweiterungen geschrieben haben. Neben den dazu bekannten Erweiterungen PostBIS und PostGIS gibt es auch noch etliche andere Erweiterungen.

Auch größere Unternehmen und Behörden sind von den Vorteilen von PostgreSQL überzeugt. So setzen z. B. Apple, BASF, Cisco, Fujitsu und Skype, aber auch das US-Außenministerium PostgreSQL ein. Würde man den Einsatz in Geräten wie Waschmaschinen oder Telefonen mitzählen, so wäre PostgreSQL sogar die am stärksten verbreitetste Datenbank-Engine.

DIENSTNUTZUNG

Interessierte Nutzer können über <https://vweb-antrag.gwdg.de/postgresql> eine PostgreSQL-Datenbank beantragen. Die Datenbank wird danach erstellt und die Zugangsdaten werden per E-Mail zugesendet. Auf Seiten des Kunden muss dazu der Port 5432 frei sein. Aktuell wird die PostgreSQL-Version 10.4 eingesetzt. Zudem steht auf <https://postgresql.gwdg.de> mit phpPgAdmin eine gute grafische Oberfläche für den Datenzugriff zur Verfügung (siehe Abb. 1).

FEATURES

Im Folgendem werden kurz einige Features von PostgreSQL erläutert. Ein Beispiel für eine Erweiterung in PostgreSQL, die

bei der Suche nach Wörtern, bei denen man die genaue Schreibweise nicht kennt, sinnvoll sein kann, ist die *pg_tgrm*-Extension (*CREATE extension pg_tgrm*; siehe Abb. 2).

Komplexere Statements sind mit PostgreSQL möglich; hierzu einige Beispiele:

New Service “PostgreSQL“

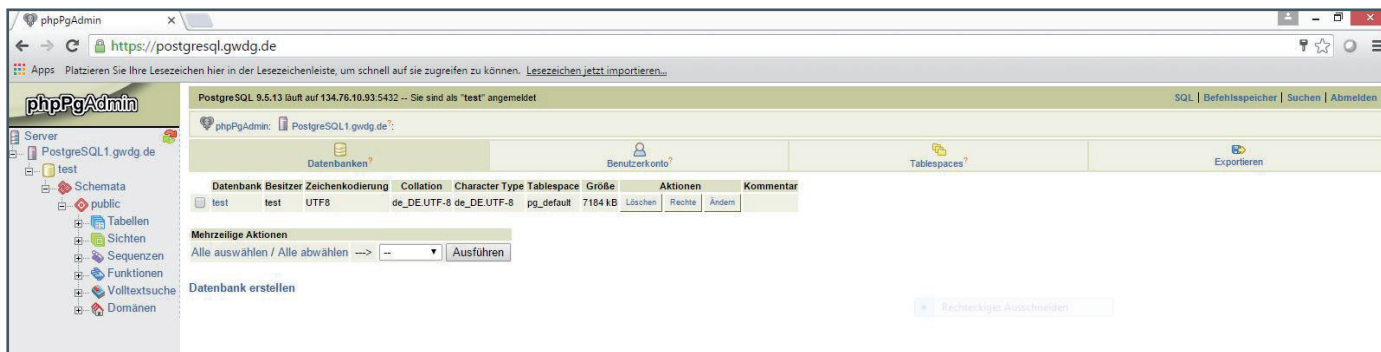
Since the 1th of August 2018 the GWDG offers PostgreSQL as a new service to it's customers. The GWDG already offers OracleSQL as a service for big and complex databases and MySQL appropriate for web services. Now we offer PostgreSQL to our customers. It is suitable for sophisticated statements and is especially useful in the bioinformatics and for geolocation data. Beside well known extensions like PostBIS and PostGIS there are several other extensions available.

Everyone who is interested to use PostgreSQL, just need to fill in the form <https://vweb-antrag.gwdg.de/postgresql> to receive a database with access data. Port 5432 should be free on server site. Actually we use PostgreSQL version 10.4. Also we offer with phpPgAdmin a convenient graphical user interface on <https://postgresql.gwdg.de>.

If you are interested in PostgreSQL, but you have concerns regarding administration, updates and backups, just fill in the formular on <https://vweb-antrag.gwdg.de/postgresql> to source these things out to the GWDG. For your information: We dump all databases daily and we have them available for the last 97 days.

Because of major release updates in PostgreSQL needs downtime, we don't do any major release updates till Christmas. Minor release updates we can perform without any downtime when needed. Only between Christmas and Sylvester we do major release updates; we announce them timely. So in case you need 24/7 also between Christmas and Sylvester we recommend our MySQL service instead (<https://vweb-antrag.gwdg.de/mysql>).

If you have questions, suggestions or issues with PostgreSQL feel free to contact Mr. Roland Groh (e-mail: PostgreSQL@gwdg.de, phone: 0551 201-1838).



1_phpPgAdmin-Oberfläche auf <https://postgresql.gwdg.de>

```
test=# select * from orte order by t_ort <-> 'Sankt Pölten' limit 10;
      t_ort
-----
St. Pölten
Sankt Ilgen
Sankt Gilgen
Sankt Blasen
Sankt Gallen
Sankt Anna am Aigen
Altenmarkt bei Sankt Gallen
Sankt Koloman
Markt Sankt Martin
Sankt Sebastian
(10 rows)
```

2_Anwendungsbeispiel der pg_trgm-Extension in PostgreSQL

Die Möglichkeit des kombinierten *UPDATE* und *SELECT*:

```
UPDATE t_test SET id = id + 1 RETURNING *;
```

Bestimmung des Medianwertes aus einer Spalte:

```
SELECT percentile_disc(0.5) WITHIN GROUP (ORDER BY
column) FROM table;
```

Bei transaktionsbasierten Statements können bereits durch ein *SELECT* eine bis x Zeilen (im Beispiel 2) für andere Transaktionen gesperrt werden:

```
BEGIN;
SELECT * FROM table WHERE id > 0 LIMIT 2 SKIP LOCKED;
...
COMMIT;
```

Wenn Sie PostgreSQL gerne einsetzen möchten, sich aber die Administration, Updates und Backup-Handling ersparen wollen, so bietet die GWDG nun unter <https://vweb-antrag.gwdg.de/postgresql> PostgreSQL zur Nutzung an. Alle Datenbanken werden täglich gedumpt. Die Dumps der letzten sieben Tage werden lokal vorgehalten. Danach werden diese ins TSM verschoben und für 90 Tage aufbewahrt. Somit stehen im Fehlerfall zur Wiederherstellung Dumps der letzten 97 Tage zur Verfügung.

HINWEIS ZU DOWNTIMES UND VERFÜGBARKEIT

Da es auch trotz Replikation (gegenwärtig) nicht möglich wäre, bei PostgreSQL ein Major-Update (Versionswechsel) ohne Downtime durchzuführen, werden jedes Jahr bis Weihnachten keine Major-Updates vorgenommen. Ein Major-Update erfolgt daher nur in der Zeit zwischen Weihnachten und Silvester nach vorheriger Ankündigung. Kleinere ggf. sicherheitsrelevante Minor-Release-Updates sind ohne Probleme jederzeit ohne Downtime möglich. Für alle Nutzer, die eine 24/7-Verfügbarkeit auch zum Jahreswechsel benötigen, ist dieser Dienst daher nicht geeignet. Für diese empfehlen wir den MySQL-Dienst der GWDG (<https://vweb-antrag.gwdg.de/mysql>).

Bei Fragen, Problemen oder Anregungen rund um das Thema „PostgreSQL“ steht Ihnen Herr Roland Groh (E-Mail: PostgreSQL@gwdg.de, Tel.: 0551 201-1838) gern zur Verfügung. ■





Die Kraftmuschel selber erweitern – Programmierübung für Auszubildende der Fachinformatik

Text und Kontakt:

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

In verschiedenen Artikeln der GWDG-Nachrichten wurde schon über die Microsoft PowerShell und ihre Erweiterungsmöglichkeiten berichtet. In diesem Artikel soll näher betrachtet werden, wie sich diese Erweiterungsfähigkeit gut in die Ausbildung von Fachinformatikern der beiden Fachrichtungen Systemintegration und Anwendungsentwicklung einbauen lässt. Dies erweitert die Fähigkeit der beiden Ausbildungsgänge, im Windows-Umfeld Lösungen für spezifische Anforderungen oder Automatisierungen umzusetzen.

DIE AUFGABE

In dem Ausbildungsblock „Compiler/Assembler“ werden die Auszubildenden in mehreren Programmierübungen in die Funktions- und Arbeitsweise dieser Werkzeuge eingewiesen. Im .Net Framework 2.0 SDK (Software Development Kit) ist für diesen Zweck als Übungsprojekt ein einfacher C-Compiler enthalten, der in der .Net Framework-Sprache C# geschrieben ist. Die Auszubildenden haben hier die Möglichkeit, sich einmal anhand eines einfachen C-Compilers im Detail die Funktions- und Arbeitsweise solch eines Werkzeuges anzusehen, das sie für ihre tägliche Arbeit während der Ausbildung und später während ihres Berufslebens nutzen werden.

Die erste Aufgabe war, das im SDK enthaltene C#-C-Compiler-Projekt zu kompilieren, so dass daraus ein ausführbares Programm entsteht. Mittels dieses unter Windows ausführbaren Programms konnten dann einfache C-Quellzeilen-Auflistungen (engl. source codes) ebenfalls in ausführbare Programme übersetzt werden.

Die nächste Aufgabe bestand nun darin, das Hauptprogramm des Beispiel-C-Compilers so umzubauen, dass daraus eine dynamisch ladbare Bibliothek (engl. Dynamic Link Library, kurz DLL)

entsteht, so dass die Compiler-Funktionalität in andere Programme bei Bedarf eingebunden werden kann.

Als Wiederholungsübung aus dem Bereich der Plug-in-Programmierung war nun die Aufgabenstellung, ein PowerShell-Modul zu entwickeln, das die Kommandozeilen-Parameter für den Compiler abfragt und diese der ladbaren dynamischen C-Compiler-Bibliothek übergibt.

Using the Enhancement Options of Microsoft PowerShell for IT Specialist Apprenticeship

Several articles in the GWDG News have already reported on Microsoft PowerShell and its enhancement options. This article will take a closer look at how this expandability can be well integrated into the apprenticeship of IT specialists in the two fields of systems integration and application development. This extends the ability of the two apprenticeship courses to implement solutions for specific requirements or automations in the Windows environment.

DIE AUSFÜHRUNG

Als Projektvorlage wird die Visual Studio C#-Klassenbibliothek genommen. In den Verweisen des Klassenbibliothek-Projekts für das PowerShell-Modul ist noch die *System.Management.Automation.dll* hinzuzufügen, zu finden unter *C:\Program Files (x86)\Reference Assemblies\Microsoft\WindowsPowerShell\3.0*, und die DLL des zuvor genannten Teilprojekts. Nun sind diese beiden eingefügten DLLs noch mittels des Schlüsselworts *using* einzubinden, so dass die komplette Funktionalität zur Verfügung steht (siehe in der Quellzeilen-Auflistung die Zeilen 1 und 2). Wichtig ist, dass die Klasse in Zeile 7 noch mit bestimmten Eigenschaften versehen wird, in Zeile 6 zu sehen, damit die PowerShell beim Laden dieses Moduls die entsprechende Funktionalität zur Verfügung stellt. Das sind im Einzelnen die Eigenschaft *Cmdlet* mit dem Parameter für das Verb *Get* aus der Aufzählung *VerbsCommon*, das als Zeichenkette frei wählbare Hauptwort, in diesem Fall *TMyC*, sowie

als Abschluss der Standard-Übergabe-Parametersatz *Executable*.

Von den vier überschreibbaren Methoden der Automatisierungs-DLL für die PowerShell-Module *BeginProcessing*, *EndProcessing*, *StopProcessing* und *ProcessRecord* wird in diesem Beispiel nur die Methode *ProcessRecord*, wie in der Quellzeilen-Auflistung in den Zeilen 27 bis 30 zu sehen, überschrieben, also durch eigene Quellzeilen erweitert.

Der einfache C-Compiler kann zwei Ausgaben erzeugen, ausführbare Programme, unter Windows mit der Dateiendung *.EXE*, und DLLs mit der Dateiendung *.DLL*. Die Festlegung wird mittels eines Übergabe-Parametersatzes für das jeweiligen Ausgabeformat geregelt. Um auf diese beiden Ausgabeformate reagieren zu können, können für Benutzer verschiedene Übergabe-Parametersätze zusammengestellt werden. Dies ist in der Quellzeilen-Auflistung den Zeilen 11, 12, 14, 15, 17, 18, 20, 21 und 23, 24 zu entnehmen.

Exemplarisch soll nun eine Zeile der Quellzeilen-Auflistung

```

1 using System.Management.Automation;
2 using TMyCCL;
3
4 namespace TMyCCmdLet
5 {
6     [Cmdlet(VerbsCommon.Get, "TMyC", DefaultParameterSetName = "Executable")]
7     public class TMyCmdLet: Cmdlet
8     {
9         TmyCcl _tmyCcl = new TmyCcl();
10
11         [Parameter(Position = 0, Mandatory = true, ParameterSetName =      ↗
12             "Executable", HelpMessage = "Type /debug or /nodebug.")]
13         [Parameter(Position = 0, Mandatory = true, ParameterSetName = "DLL",      ↗
14             HelpMessage = "Type /debug or /nodebug.")]
15         public string target { get; set; }
16         [Parameter(Position = 1, Mandatory = true, ParameterSetName =      ↗
17             "Executable", HelpMessage = "Type /exe.")]
18         [Parameter(Position = 1, Mandatory = true, ParameterSetName = "DLL",      ↗
19             HelpMessage = "Type /dll.")]
20         public string type { get; set; }
21         [Parameter(Position = 2, Mandatory = true, ParameterSetName =      ↗
22             "Executable", HelpMessage = "Type /list.")]
23         [Parameter(Position = 2, Mandatory = true, ParameterSetName = "DLL",      ↗
24             HelpMessage = "Type /list.")]
25         public string listing { get; set; }
26         [Parameter(Position = 3, Mandatory = true, ParameterSetName =      ↗
27             "Executable", HelpMessage = "Type the output path location. For example ↗
28             C:\\Windows\\temp")]
29         [Parameter(Position = 3, Mandatory = true, ParameterSetName = "DLL",      ↗
30             HelpMessage = "Type the output path location. For example C:\\Windows\\ ↗
31             temp")]
32         public string outdirpath { get; set; }
33         [Parameter(Position = 4, Mandatory = true, ParameterSetName =      ↗
34             "Executable", HelpMessage = "Type the path to sourcefile location. For ↗
35             example C:\\Users\\JaneDoe\\Documents")]
36         [Parameter(Position = 4, Mandatory = true, ParameterSetName = "DLL",      ↗
37             HelpMessage = "Type the path to sourcefile location. For example C:\\ ↗
38             Users\\JaneDoe\\Documents")]
39         public string sourcefile { get; set; }
40
41         protected override void ProcessRecord()
42         {
43             _tmyCcl.TmyC(target, type, listing, outdirpath, sourcefile);
44         }
45     }
46 }

```

```
PS C:\Users\hindermath> Import-Module "C:\Users\hindermath\Documents\Visual Studio 2017\Projects\Tool Developers Guide\Samples\myc\src\TMyC.dll"
PS C:\Users\hindermath> Get-Module

ModuleType Version Name ExportedCommands
-----
Script 1.0.0.0 ISE {Get-IseSnippet, Import-IseSnippet, New-IseSnippet}
Manifest 3.1.0.0 Microsoft.PowerShell.Management {Add-Computer, Add-Content, Checkpoint-Computer, Clear-Content...}
Manifest 3.1.0.0 Microsoft.PowerShell.Utility {Add-Member, Add-Type, Clear-Variable, Compare-Object...}
Binary 1.0.0.0 TMyCcmdLet Get-TMyC

PS C:\Users\hindermath>
```

Abb. 1

```
PS D:\hindermath\Documents> Get-TMyC -target /debug -type /exe -listing /list -outdirpath D:\hindermath\Documents -sourcefile .\t01.tmyc
```

Abb. 2

```
Type /debug or /nodebug.
target: /debug
type: !?
Type /exe.
type: /exe
listing: !?
type: /list.
listing: /list
outdirpath: !?
Type the output path location. For example C:\Windows\temp
outdirpath: .
sourcefile: d:\hindermath\documents\t01.tmyc
(1) : error M0000: myc [/debug] [/nodebug] [/list] [/dll] [/exe] [/outdir:path] filename.myc

Compiler aborting: System.ApplicationException: Aborting compilation
bei TMyCCL.Io.Abort(String s) in C:\Users\hindermath\Documents\Visual Studio 2017\Projects\Tool Developers Guide\Samples\myc\src\TMyCCL\Io.cs:Zeile 91.
bei TMyCCL.Io.ParseArgs() in C:\Users\hindermath\Documents\Visual Studio 2017\Projects\Tool Developers Guide\Samples\myc\src\TMyCCL\Io.cs:Zeile 168.
bei TMyCCL.Io.<ctor>(String[] s) in C:\Users\hindermath\Documents\Visual Studio 2017\Projects\Tool Developers Guide\Samples\myc\src\TMyCCL\Io.cs:Zeile 178.
bei TMyCCL.TMyC(String target, String type, String listing, String outdirpath, String sourcefile) in C:\Users\hindermath\Documents\Visual Studio 2017\Projects\Tool Developers Guide\Samples\myc\src\TMyCCL\TMyC.cs:Zeile 68.

PS C:\Users\hindermath>
```

Abb. 3

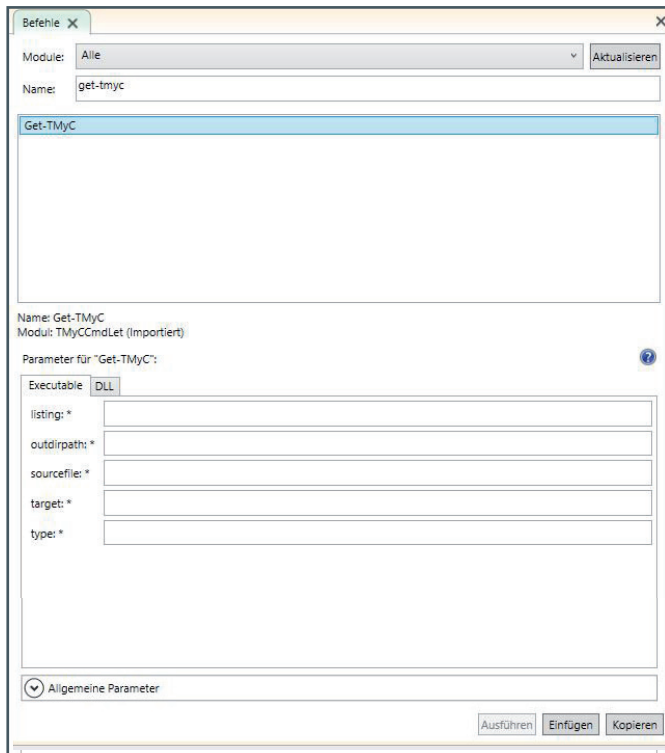


Abb. 4

für die Übergabe-Parameter näher betrachtet werden, in diesem Fall *target* in den Zeilen 11 bis 13. Der eigentliche Übergabe-Parameter *target* in Zeile 13 wird als öffentlich sichtbar deklariert und in die Lage versetzt, Werte mittels *set* entgegenzunehmen oder seinen aktuellen Wert mittels *get* darzustellen. Weiterhin

muss auch die Deklaration des bzw. eines Übergabe-Parameters mit bestimmten Eigenschaften versehen werden, damit die PowerShell auch in diesem Fall bzw. diesen Fällen beim Laden des Moduls für die Übergabe-Parameter die entsprechenden Funktionalitäten zur Verfügung stellt. Das sind im Einzelnen die Eigenschaft *Parameter* mit den Parametern *Position*, an welcher Stelle in der Reihenfolge der Parameter abgefragt wird, *Mandatory*, als Pflichtparameter „Wahr“ (engl. true) oder „Falsch“ (engl. false), *ParameterSetName*, in welchem Parametersatz der Übergabe-Parameter abgefragt werden soll, und *HelpMessage* für die Anzeige eines kleinen Hilfetextes bei Bedarf. In diesem Beispiel ist gut zu sehen, wie für die Verwendung in mehreren Parametersätzen jeweils eine Zeile pro Parametersatz mit bestimmten Eigenschaften versehen werden muss – Zeile 11 *Executable* und Zeile 12 *DLL*.

Wie in diesem Beispiel eines PowerShell-Moduls zu sehen ist, kann durch geschicktes Auslagern von Funktionalitäten das eigentliche Modul sehr übersichtlich gestaltet werden. Das kommt auch der Fehlersuche zugute.

Nachdem das Modul sodann erfolgreich und ohne Fehler übersetzt oder auch kompiliert (engl. compiled) wurde, kann es in eine PowerShell-Instanz zu deren Laufzeit dynamisch geladen, genutzt und entladen werden. Dieses Vorgehen wird im nächsten Abschnitt beschrieben.

DAS ERGEBNIS

Gute Unterstützung zum Testen neuer PowerShell-Module bietet neben der eigentlichen PowerShell, die zur Ausführung fertig entwickelter PowerShell-Skripte gut geeignet ist, die PowerShell ISE (ISE steht für Integrierte Skript-Entwicklungsumgebung).

Diese einfache Skript-Entwicklungsumgebung bietet grundlegende Unterstützung, um ein neues Modul zu testen.

Um das zuvor erstellte PowerShell-Modul zu testen, wird es mithilfe des Befehls *Import-Modul* innerhalb der PowerShell zur Verfügung gestellt. Der Befehl *Remove-Modul* kann gezielt Module entladen. Mithilfe des Befehls *Get-Module* kann die Liste der geladenen Module abgerufen werden. In der letzten Zeile der Abb. 1 zeigt *Get-TMyC* an, dass das Laden erfolgreich war. Um nun die Parametersätze zu prüfen und zu testen, kann in der PowerShell ISE mittels „Ansicht -> Befehl-Add-On anzeigen...“ die Anzeige für die Parameter eines Moduls eingeschaltet werden. Für das Modul, das in diesem Artikel beschrieben wird, sieht dieser Bereich wie in Abb. 4 dargestellt aus. Gut im unteren Bereich ist auch der mittels Reiter-Ansicht separierte Bereich für die Parametersätze *Executable* und *DLL* zu sehen, die weiter oben beschrieben worden sind. Wenn alles richtig eingegeben worden ist, dann sieht der vollständige Befehl wie in Abb. 2 dargestellt aus. Dies kann erreicht werden, indem die Eingabefelder des entsprechenden Parametersatzes ausgefüllt werden und danach mittels der Schaltfläche „Kopieren“ per Strg+V in den interaktiven Bereich eingefügt werden.

Im Laufe der ersten Tests des PowerShell-Moduls haben sich

noch einige Fehler gezeigt. Die Compiler-Funktion bricht mit einer eigenen Fehlermeldung ab (siehe Abb. 3). Die nun beginnende Fehlersuche (engl. debugging) ist nicht Bestandteil dieses Artikels. Aber für die Auszubildenden kann mit dieser Situation ein weiterer wesentlicher Aspekt der Programmierfähigkeit vermittelt werden, nämlich die Fehlersuche. Dies ist neben der kreativen Möglichkeit, Anwendungen oder Lösungen zu entwickeln, ein wesentlicher, aber oftmals auch unbeliebter und frustrierender Teil im Arbeitsleben eines Anwendungsentwicklers oder Systemintegrators.

AUSBLICK

In dieser Übung wurde (noch) nicht betrachtet, ob inzwischen die Möglichkeit besteht, mit dem .Net Core Framework auch PowerShell-Module zu entwickeln, die auch in der plattformübergreifenden Version der Microsoft PowerShell für macOS und verschiedene UNIX-Derivate (siehe hierzu <https://github.com/powershell/powershell>) zur Ausführung gebracht werden können. Dies wird eine der nächsten Anforderungen für zukünftige Programmierübungen in diesem Bereich sein. ●



Using the Parallel Processing Power of the GWDG Scientific Compute Cluster

Upcoming Introductory and Parallel Programming Courses

GWDG operates a scientific compute cluster with currently 17,048 cores and a total compute power of 369 Teraflops ($3.69 \cdot 10^{14}$ floating point operations per second), which can be used by all scientists of the institutes of GWDG's supporting organisations, University of Göttingen and Max Planck Society.

In order to facilitate the access to and the efficient use of these computing resources, GWDG offers introductory and parallel programming courses, held at GWDG's site 'Am Faßberg' (1st and 3rd course) and 'Geiststraße' (2nd course).

The next courses in 2018 are

> **October 22nd, 9:30 am - 4:00 pm**

Using the GWDG Scientific Compute Clusters – An Introduction

This course explains all steps for accessing GWDG's clusters, to compile and install software, and to work with the batch system for the execution of application jobs. The course is intended for new or inexperienced users of the clusters.

> **October 23rd - 24th, 9:15 am - 5:00 pm**

Parallel Programming with MPI (Including MPI for Python)

This course introduces the message passing interface (MPI) for programming parallel applications in FORTRAN, C, and in Python. All concepts will be illustrated with hands on exercises. Examples of parallel applications will be presented and analysed.

> **November 6th, 9:15 am - 5:00 pm**

Programming with CUDA – An Introduction

Graphic processors (GPUs) are increasingly used as computational accelerators for highly parallel applications. This course introduces hardware and parallelization concepts for GPUs and the CUDA programming environment for C and Fortran, including the language elements for controlling the processor parallelism and for accessing the various levels of memory.

These three courses are repeated regularly. Other courses on parallel computing, dealing with more specialized topics can be arranged on demand. The possible subjects include parallel programming for shared memory systems and using extensions of C or Fortran with high level parallel constructs.

More Information about the courses held regularly or on demand at www.gwdg.de/scientific-computing-courses.

Information for registering for the courses at www.gwdg.de/courses.

If you have any further questions please contact support@gwdg.de.

>> www.gwdg.de/courses



International Summer School on Data Science 2018

Text and Contact:

Lena Steilen
lena.steilen@gwdg.de
0551 201-2153

The second International Göttingen Summer School on Data Science was held at the beginning of August 2018. The intense two week course aimed to raise awareness of the chances and challenges of “Data Science” and to prepare the next generation of young researchers for working with digital data.

PROGRAM

From August 2 – 16 2018 the second International Summer School on Data Science took place at the beautiful location “Alte Mensa” in the City of Göttingen. 32 students (mainly MA, PhD) participated in this course. They came from research institutions in nine different countries with diverse disciplinary backgrounds. Together they learned in this intense two week Summer School about the chances and challenges of Data Science. The teaching focus laid on providing an overview of the multilayered topics and methods belonging to Data Science:

- Deal with the Data Lifecycle
- Best Practices in Data Management
- Methods (Modelling, Statistics, Mining, ...)
- Infrastructures and Platforms
- Application Examples
- Ethical, legal, and social Aspects

Building on the good cooperation in 2017 this year’s Summer School was organized jointly by the Institute of Computer Science, Göttingen International, the GWDG’s and SUB’s joint service unit on research data management – Göttingen eResearch Alliance, and the University Medical Centre.

KEYNOTE

This year’s participants were cordially welcomed by greetings of Prof. Dr. Hiltraud Casper-Hehne for the presidential board of the University, Prof. Dr. Ramin Yahyapour for the Institute of Computer Science and the GWDG, Dr. Katharina Jannasch for Göttingen International, Timo Gnadl for the Göttingen eResearch Alliance, and Dr. Sven Bingert for the organizing committee on the first day. Except getting to know each other through short presentations, the keynote “Data Science pipeline, publishing, challenges and the FAIR principles” by Dr. Amrapali Zaveri (Institute of Data Science at Maastricht University) was a further highlight on the first day.

Zweite internationale Data Science Summer School in Göttingen

Anfang August 2018 fand die zweite internationale Data Science Summer School in Göttingen statt. Dieser zweiwöchige Intensivkurs hatte zum Ziel, den Studierenden die Chancen und Herausforderungen im breiten Feld „Data Science“ zu vermitteln und die kommende Generation Forscher auf die Arbeit mit digitalen Daten vorzubereiten.



In her talk she emphasized especially on the challenges of data quality assessment and the importance of ensuring that all digital resources are Findable, Accessible, Interoperable and Reusable (FAIR).

BRING YOUR OWN DATA

Another highlight was the session “Bring your own Data” in the middle of the first full week. In this session the students were asked to bring data from their own current research projects. During the session they could e.g. check their own data quality, proof their methodological approach, or try new tools.



LECTURES

In the course of the Summer School the participants learned and worked with about 20 lecturers representing disciplines like computer science, mathematics, economics, geography, medicine, (statistical) bioinformatics, and engineering sciences. Gratifyingly lecturers from the German-Japanese university cooperation HeKKSaGOn joint the Summer School like in 2017.

The course was conceptualized to reach a balance between theoretical input and giving space to practice and or discuss the presented topics. Especially during the interactive tutorials the discussions were conducted active and very fruitful. The multidisciplinary input was challenge and chance for the participants at the same time. That the students were able to tackle this was shown by their presentations on the last day. They showed lessons learned, own ideas, new insights and also memorable moments of the Summer School.



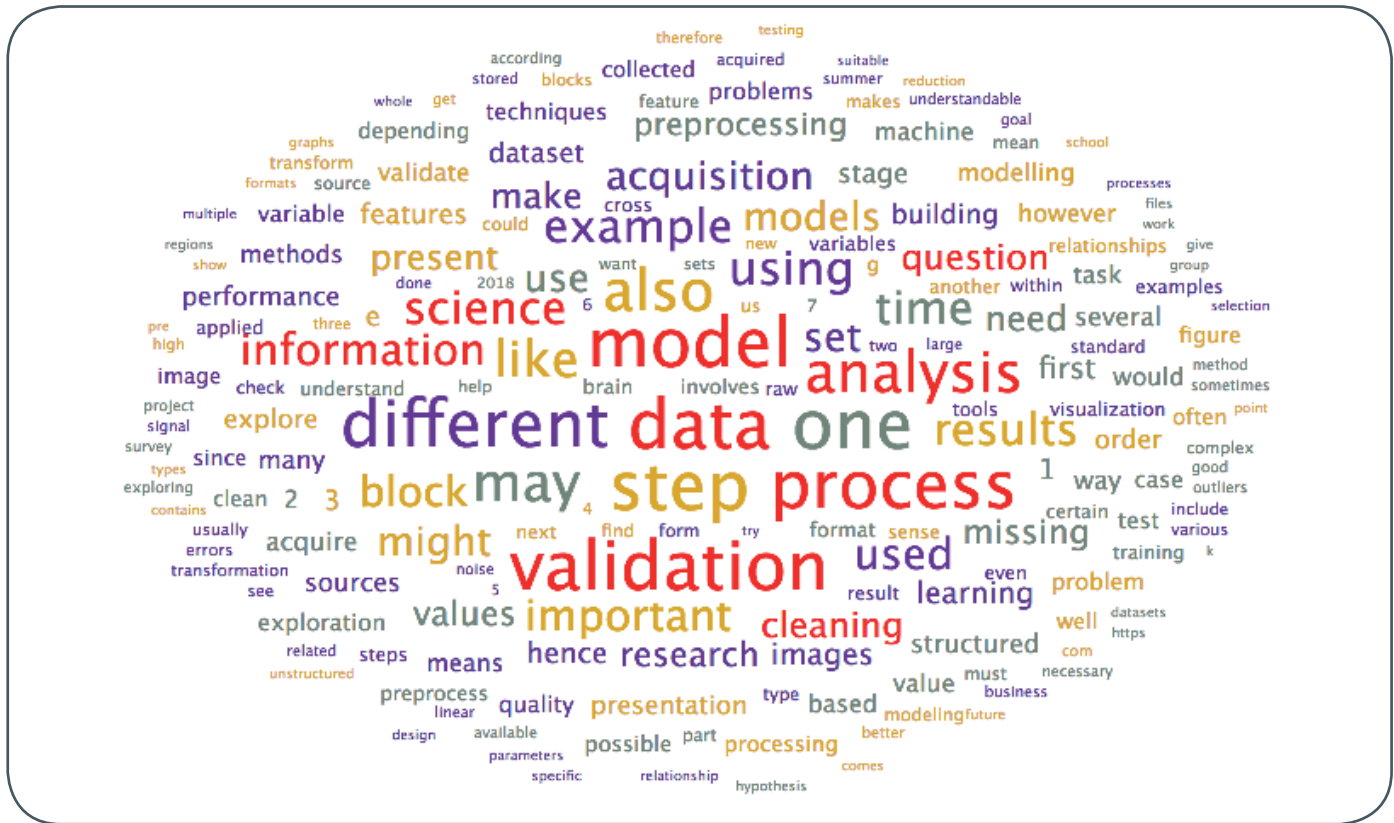


Figure 1: Word cloud based on the students' homework to describe the data science building blocks. More frequent words appear larger; therefore "validation" is regarded as an important topic when doing Data Science.

Beside the intellectual/teaching program the organization committee prepared a diversified social program. The events reached from a Welcome Reception, over guided tours through the City of Göttingen and a hiking tour through the municipal forest to a joint BBQ and a Farewell Dinner. The students had much fun and established contacts to other students and lecturers interested in or doing Data Science from all over the world.

AGAIN IN 2019

After the entirely positive feedback the organization committee plans to offer another Summer School on Data Science in August 2019. We are already looking forward to welcoming the next group of students aiming to become Data Science researchers and experts. ●

Kurz & knapp

Netzwerktreffen zum Thema „Plagiatsprävention und akademische Integrität“

Für alle am Thema „Plagiatsprävention“ Interessierten findet am Donnerstag, den 18. Oktober 2018, von 14:00 – 18:00 Uhr ein „Netzwerktreffen zum Thema Plagiatsprävention und akademische Integrität“ im Tagungs- und Veranstaltungshaus „Alte Mensa“ der Universität Göttingen statt. Das Treffen wird von der GWDG und der Firma Turnitin LLC veranstaltet. Alle Interessierten sind herzlich eingeladen daran teilzunehmen. Nähere Informationen zur Veranstaltung sind unter dem URL <https://www.eventbrite.co.uk/e/netzwerktreffen-zum-thema-plagiatspravention-und-akademische-integritat-tickets-48365289884> zu finden. Bei Fragen zur Veranstaltung können Sie sich gerne an Herrn Roland Groh (E-Mail: roland.groh@gwdg.de, Tel.: 0551 201-1838) wenden.

Groh

Öffnungszeiten des Rechenzentrums am Tag der Deutschen Einheit

Das Rechenzentrum der GWDG ist am Mittwoch, den 3. Oktober 2018, dem Tag der Deutschen Einheit, 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.

Pohl

Completion of the MIKELANGELO Project

Text and Contact:

Peter Chronz
peter.chronz@gwdg.de
0551 201-2199

Maik Srba
maik.srba@gwdg.de
0551 201-2186

The Horizon 2020 project “MIKELANGELO” has successfully concluded on 31.12.2017. The project included nine international partners, ran for 36 months, and had an EU contribution of nearly € 6 million. The project’s goals revolved around improved agility of cloud computing, improved I/O performance of virtual machines (VMs), and the convergence of cloud computing, big data, and high performance computing (HPC). The project’s results include many new components, which lay the foundation for further research, some of them are slated for upstreaming into the Linux kernel.

MOTIVATION AND GOALS

The project was motivated by performance losses in cloud computing, especially for I/O operations, a lack of agility in cloud computing, lack of cross-layer optimization, and an insufficient compatibility of HPC, cloud computing, and big data computing.

The main issues with performance in cloud computing stem from the pervasive use of compute virtualization in form of VMs. VMs have been optimized for most compute operations to have negligible overhead, through hypervisor implementations and hardware support in CPUs. Nevertheless, I/O operations in VMs still incur a high overhead. Some hypervisors – the components running VMs – approach the performance problem of I/O in VMs with para-virtualization, which requires special drivers inside the guest OS, that is, inside the VM. This approach limits the types of guest VMs and needs to be maintained to stay in sync with new guest and host kernels. Thus, one of MIKELANGELO’s goals was to find a generic solution to the I/O performance problem, relying only on hypervisor code, that is in the host OS.

Another motivation of MIKELANGELO was to improve the lack of agility in cloud computing. This lack of agility was apparent to anyone trying to dynamically deploy a complex service in the cloud. Complex services often rely on many virtual infrastructure components and on a set of applications running on those resources. Furthermore, their deployment needs to be monitored and repaired in case of a failure. When we set out with MIKELANGELO, there were no integrated and easy-to-use solutions for this problem. The hype around Docker was in its infancy and Kubernetes was not yet public. So another goal of MIKELANGELO was to develop simple to use mechanisms to deploy complex software stacks in the cloud.

The lack of cross layer optimization in cloud computing refers to inefficiencies in cloud computing due to configurations across the layers of a cloud stack, which do not match. A cloud stack can be seen to comprise a hardware layer, a virtual infrastructure layer, and application/service layers. Each layer usually consists of many different components, such as physical hosts in the hardware

layer, virtual machines in the virtual layer, and web applications in the service layer. These components interact within and across those layers and influence each other. Furthermore, most components have a set of parameters, which can be configured for security and performance. One of the goals of MIKELANGELO was the investigation of optimization of parameters across the whole cloud stack to improve target metrics such as quality of service, utilization, and energy efficiency.

Finally, MIKELANGELO had the goal to investigate a convergence of cloud computing, high performance computing, and big data. Over time cloud computing developed as an alternative to perform some of the tasks that are traditionally tackled by high performance computing. Both paradigms offer certain advantages. For example, high performance computing offers a specialized and optimized environment, based on hardware, software, and middleware configurations. On the other hand cloud computing offers elasticity and flexible configuration by allowing customers to use their own OS, including drivers.. MIKELANGELO set out to investigate how one could combine the benefits of both paradigms, by running HPC workloads in the cloud and by virtualizing HPC deployments.

Abschluss des Projektes „MIKELANGELO“

Das Horizon-2020-Projekt „MIKELANGELO“ wurde zum 31.12.2017 erfolgreich abgeschlossen. Das Projekt bestand aus neun internationalen Partnern, lief über 36 Monate und wurde durch die EU mit knapp 6 Millionen Euro gefördert. Ziele des Projektes waren die Verbesserung der Agilität von Cloud Computing, die verbesserte I/O-Leistung virtueller Maschinen und die Konvergenz von Cloud Computing, Big Data und Hochleistungsrechnen. Die Projektergebnisse umfassen eine Vielzahl neuer Komponenten, die Grundlagen für weitere Forschung legen und teilweise für ein Upstreaming in den Linux-Kernel vorgesehen sind.

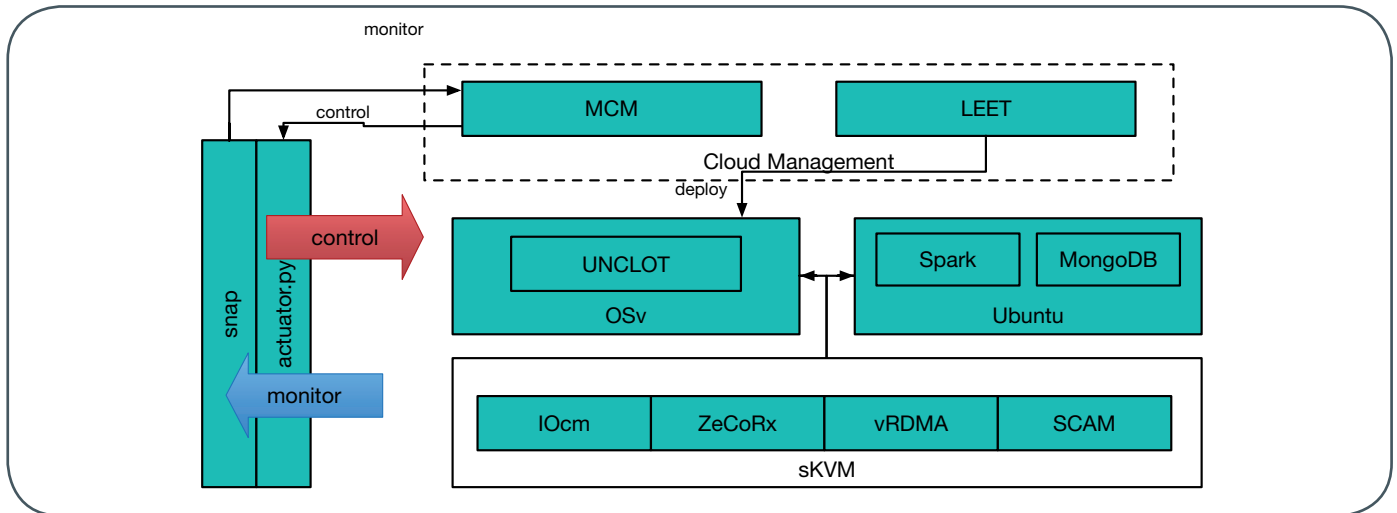


Figure 1: The MIKELANGELO architecture

TECHNICAL RESULTS

To tackle our goals the project has developed a set of components, which approach individual issues and which can be integrated into a whole stack as shown in Figure 1. Although some components tackle multiple issues at once they can be broadly categorized by their motivations and goals: improving performance of virtual machines, improving the agility of cloud deployments, cross-layer optimization, cloud/HPC convergence. The project's components are available as open source on MIKELANGELO's GitHub account (<https://github.com/mikelangelo-project>) and in GWDG's GitLab installation (<https://gitlab.gwdg.de/mikelangelo> and <https://gitlab.gwdg.de/scotty>).

Performance

Because the biggest inefficiency in VM performance lies in I/O operations MIKELANGELO has focused on this issue. Furthermore, we chose to work on KVM as target hypervisor for our optimizations. KVM is the most popular choice to run VMs in OpenStack and is used, for example, in GWDG Cloud Server. Other popular hypervisors are Xen and VMware ESX. However, Xen limits the guest OS and its configuration, while VMware is a proprietary, closed-source product.

Our work on KVM includes two extensions. First, our project partner IBM worked on a component called IOcm (IO core manager). This component dedicates a set of physical CPU cores to perform only virtualized I/O operations. This approach led to vastly improved performance for I/O-heavy workloads, such as stream processing. Unfortunately, these great results could only be replicated for special use cases and not in a general setting. To provide a significant speed up for I/O operations IOcm requires the workload to exchange large packages, which is not typical for many applications. Instead most applications exchange many small packages. There turned out to be more problems with IOcm that surfaced through discussions in the KVM and Linux kernel community, especially with RedHat. These problems would have made it unlikely for IOcm to be upstreamed into the Linux kernel. These drawbacks led to the development of ZeCoRx.

ZeCoRx stands for "Zero Copy Rx" and follows another approach to optimize the I/O operations in the hypervisor. The main idea is to avoid copying incoming messages from kernel space to

the VM's memory and instead providing the VM with direct access to the original copy of an incoming message. Avoiding this copy operation has yielded promising speedups in early experiments. By the end of MIKELANGELO ZeCoRx has reached a prototypical stage. It is now being developed further by IBM with the goal to upstream the result into the Linux kernel.

The third approach to improve IO performance in VMs in MIKELANGELO is called vRDMA, which stands for virtual Remote Direct Memory Access. RDMA is a mechanism used in high performance networks, such as Infiniband and converged Ethernet (via RoCE). RDMA allows incoming messages to be copied directly into the recipients memory, so that no copy operation from the networking queue buffer into the memory is required. However, when using virtual machines the received data still needs to be copied from kernel space into the VMs memory. To eliminate this costly copy operation MIKELANGELO's partner Huawei has developed a special RDMA driver that will store incoming RDMA messages into memory that can directly be accessed by the VM by using IVSHMEM (https://doc.dpdk.org/guides-16.04/prog_guide/ivshmem_lib.html).

Key Facts

Duration: 01.01.2015 – 31.12.2017 (36 months)

Programme: H2020

Partners: 9 Partners across Europe and Israel: Xlab (Slovenia), University of Stuttgart (Germany), IBM (Israel), Intel (Ireland), GWDG (Germany), Pipistrel (Slovenia), Scylla (Israel), Ben-Gurion University (Israel), Huawei (Germany)

Funding: H2020-ICT-2014-1

EC Contribution: Grant amount of nearly 6 million EUR

Website: <https://www.mikelangelo-project.eu>



Co-funded by the Horizon 2020
Framework Programme of the European Union



Figure 2: Project participants

Agility

To improve agility in cloud computing we have followed multiple approaches in MIKELANGELO. First, our project partner Scylla have advanced their unikernel operating system OSv (<http://osv.io>). Second, Xlab, another partner, have developed an orchestration tool called LEET for OSv. Third, Scylla have developed a high performance asynchronous library called Seastar (<http://seastar.io>) and they have developed a Cassandra clone on top of Seastar, called ScyllaDB (<https://www.scylladb.com>).

OSv is a new type of operating system, developed by Cloudius Systems, which was later renamed to Scylla. OSv is a unikernel, which means that it is made to run only one process at a time. This constraint allows one to skip many complex mechanisms, which a multiprocess system, such as Linux, needs to include. For example, there is no distinction between kernel space and user space. System calls are normal function calls and thus very cheap. There is no need for memory fencing. Furthermore, OSv relies on running in the cloud. So there is no need to develop a large set of drivers to accommodate hardware. The main abstraction that needs to be supported is that provided by the VM. OSv's benefits are a very fast start up and boot time of less than a second, tiny VM images of few MBs, and low memory requirements. However, at the beginning of MIKELANGELO OSv had severe restrictions in term of usability and supported runtimes. During the project major features, such as multithreading, usability improvements, and support for many platforms have been developed. Among others the support for Python, Go, nodejs, and a new NFS client have been implemented.

To simplify the deployment of services project partner Xlab have implemented LEET, which is a package manager and deployment system for OSv. Without LEET the deployment of new applications for OSv was rather involved and thus required special skills. With LEET the initial deployment needs to be specified in a manifest. The deployment then is performed automatically, akin to Docker.

To improve agility Scylla have developed two additional components. First, Seastar has been developed to take advantage of new asynchronous mechanisms in C++14. Leveraging those features, Scylla have developed a NoSQL database as a proof of concept. This proof of concept grew into a prototype and finally a full product, called ScyllaDB. By leveraging Seastar's asynchronous mechanisms, ScyllaDB reaches a speedup of 10 over Cassandra for common workload mixes. ScyllaDB improves agility by furthermore allows dynamic scale-out with stable QoS. Whereas Cassandra's QoS degrades drastically when scaling under load, ScyllaDB keeps a nearly stable performance even while scaling and re-balancing its shards.

Cross-layer Optimization

Cross-layer optimization in MIKELANGELO revolved around resource management and resource control in the cloud stack, security enhancements, and full-stack monitoring.

First, to perform infrastructure experiments GWDG has developed an experimental system called Scotty. Scotty automates the tedious process of deploying resources, data collection, data publication, and managing experiment artifacts. Using Scotty researchers can save one to two orders of magnitude of effort when performing infrastructure experiments. This speedup comes from the automation of complex and repetitive tasks, such as deploying virtual resources, workloads, and services for the experiment. Furthermore, a significant part of the speedup comes from the re-use of workloads, which are treated like apps.

To provide a control mechanism for the full cloud stack GWDG has further developed actuator.py. Actuator.py is analogous to a distributed monitoring system with the difference that it issues control commands instead of providing metering data. To keep the system scalable and extensible actuator.py uses RabbitMQ and a plugin mechanism. The deployment of actuator.py includes a master node and many slave agents deployed across the whole cloud

stack, including the physical hosts and virtual machines. Plugins allow the control of any thinkable parameter inside the hosts and VMs. `actuator.py` is used by Scotty to control an experiment's environment and by MCM (MIKELANGELO Cloud Manager) as a control tool for live resource management.

MCM is a tool for live resource management. MCM is a controller for the cloud, which receives live metering data from a data source, such as InfluxDB, and issues control commands using `actuator.py` and a middleware, such as OpenStack Nova. The control logic can be plugged in as Python module, which makes MCM convenient to use in conjunction with Scotty.

Although MCM and `actuator.py` are independent components, which can be used in production, they shine particularly, when combined with Scotty. A typical experiment in Scotty will describe an experiment including workloads, system configuration, and MCM strategies. Scotty will then use `actuator.py` to configure the infrastructure for an experiment and it will load the control strategy into MCM. Once the experiment runs, MCM will use `actuator.py` to adjust the configuration across the cloud stack to manage the QoS.

To improve the security of colocated VMs in a cloud deployment, MIKELANGELO looked into side channel attacks on VMs. Side channel attacks presented the best option to attack colocated VMs. The project partners at Ben Gurion University have implemented a side channel attack on the shared cache, which they called SCAM (Side Channel-based Attack and Mitigation). This attack allows a malicious VM to steal private information from a VM that runs on the same physical CPU. This case is important in practice because the attacker can steal information, such as the private key of a TLS configuration. In the project this attack has been implemented and verified. To deal with this pervasive vulnerability our partners have devised a monitoring system and a mitigation mechanism. The monitoring system observes VM operations for an attack profile, such as deliberate cache thrashing. The mitigation system adds certain noise patterns, which make it much harder for an attacker to extract private information from a colocated VM.

Finally, the project partner Intel have open sourced and advanced their new monitoring system called Snap (<https://github.com/intelsdi-x/snap>). Snap features massive scalability, comprehensive management, and an advanced processing pipeline. In the project new monitoring plugins, such as the attack monitoring for SCAM, have been developed for Snap. Furthermore, Snap is used to collect live metering data used by MCM.

Cloud-HPC Convergence

To advance the integration of cloud computing and HPC our partners at the University of Stuttgart (HRLS) have developed an

extension to the HPC batch system Torque. The extended system is called vTorque and allows the use of VMs for job deployments. Instead of deploying jobs directly onto HPC nodes, VM images are booted and jobs are run inside those VMs. Leveraging the project's base improvements, such as vRDMA and IOcm/ZeCoRx the otherwise unbearable performance hit on I/O through compute virtualization has been reduced to a tolerable level. The upside is huge. Using VMs in HPC deployments allows users to install their own software stack. This in turn allows HPC users to build on new systems, which otherwise would not be available in an HPC environment. Often HPC environments significantly lag behind the state of the art in terms of kernel version, available drivers, and by extension available software. Another benefit of encapsulating the execution environment is the possibility to migrate or even scale-out HPC workloads. Migration to another HPC site becomes possible by decoupling the job implementation from the hardware and its OS configuration. At the same time a versatile VM image allows to scale out workloads to the cloud or even to migrate a whole job to the cloud.

SUSTAINABILITY

At the end of a research project there is always the question whether the project's results will lead to any sustainable impact. In MIKELANGELO, we have worked towards the goal of sustainability from the beginning. Fortunately, a lot of results from MIKELANGELO are taken up in industry collaborations and other research projects.

The work on ZeCoRx and vRDMA has led to ongoing industry collaborations between MIKELANGELO partners and major players outside of MIKELANGELO. In the case of ZeCoRx IBM works on upstreaming the results into the Linux kernel. Getting this result into the mainline Linux code would be a huge success for MIKELANGELO, even if the process takes some more time.

ScyllaDB has gained in popularity thanks to its good performance, graceful scaling, and its compatibility with Cassandra. At the same time OSv is being investigated by other projects as a unikernel guest OS for virtualized microservice architectures. In this context LEET is also being further developed to provide a virtualized alternative to containers and an integration with Kubernetes.

At GWDG we are continuing to develop Scotty, `actuator.py`, and MCM. The goal for those components is to provide a powerful testbed, which can be used across projects internally and for collaborative research through BMBF and Horizon 2020 projects.

If you would like to read more about our work, please visit our website, where you can find blog posts and our project reports: <https://www.MIKELANGELO-project.eu>. ■

Tipps & Tricks

Passwortänderung im Kundenportal auch nach Passwortablauf möglich

Ab sofort gibt es für Nutzer, deren GWDG-Account aufgrund der nicht rechtzeitigen, aus Sicherheitsgründen regelmäßig notwendigen Passwortänderung gesperrt wurde, eine einfache Möglichkeit, diese Änderung im Selfservice über das

GWDG-Kundenportal <https://www.gwdg.de> vorzunehmen bzw. nachzuholen. Die Anmeldung am Kundenportal (und nur noch dort) ist nun nämlich auch nach Passwortablauf noch mit dem abgelaufenen Passwort möglich, so dass dort dann das Passwort selbst geändert werden kann, um wieder arbeitsfähig zu sein.

Krull

Ending of the EU Horizon 2020 Project “NEPHELE”

Text and Contact:

Muzzamil Aziz
muzzamil.aziz@gwdg.de
0551 201-2114

NEPHELE was an international project that had been operated under the umbrella of EU Horizon 2020 framework from February, 2015 till January 2018. The project was led by the Institute of Communications and Computer Systems (ICCS) of the National Technical University of Athens (NTUA). The rest of the project consortium was comprised of GWDG from Germany, Seagate Systems from the UK, Mellanox Technologies from Israel, Nextworks from Italy, the University of Patras from Greece, and Interoute from Italy. This article shortly reports on the overall outcome and the achievements of the project.

WHY NEPHELE?

A growing trend of big data's burden is visible in the data centre (DC) traffic. According to the Cisco GCI Index, the data centre IP traffic growth in 2021 will reach to 20.6 ZB per year, which is a threefold of the volume estimated in 2016 [1]. A higher percentage of this data centre traffic is termed to be generated from within the data centre, knowing the fact that big data applications as being one of the significant drivers of it. NEPHELE addressed the same issue of data centres explosion and proposed a novel hybrid electronic-optical network architecture, in contrast to traditional “fat-tree” architecture, that scales linearly with the number of servers in the data centre and is capable of up to 94% power savings. Additionally, along its control path NEPHELE utilized the strength of SDN (software defined networking) paradigm to support the phenomena of application aware networking in cloud data centres. The SDN control plane enables the efficient allocation of network resources in DC networks in order to ensure the quality of service (QoS) guarantees required by the cloud applications running over the virtual infrastructure and to perform real-time optimizations based on the global traffic matrixes generated by those applications.

NEPHELE DEMO

On October 20th, 2017, the NEPHELE Demo review was held at NTUA Photonics lab in Greece. Mellanox Technologies presented a first live end-to-end demo of NEPHELE Intra DC architecture in front of the EU review panel. As a matter of fact, the EU reviewers acknowledged the fact that NEPEHELE project was one of its kind among the EU projects which realized some developments at almost every single layer of OSI reference model and was a pioneer project which combined the time slotted operations of data plane devices with the control plane management of SDN. The following subsection presents the details of the NEPHELE Intra DC architecture demonstrated in the review.

EU-Horizon-2020-Projekt „NEPHELE“ beendet

NEPHELE war ein internationales Forschungsprojekt, das im Zuge des EU-Rahmenprogramms „Horizon 2020“ mit drei Millionen Euro gefördert wurde. Es begann im Februar 2015. Die Projektlaufzeit betrug 36 Monate. Das Projektziel war die Konzeption einer Netzwerkarchitektur für Rechenzentren der nächsten Generation, die die Barrieren aktueller Netzwerktechnologien überwindet. Hierzu erforschte und entwickelte die GWDG Innovationen im Bereich Software-defined Networking (SDN). Die wichtigsten Neuerungen stellen die Kombination optischer und elektronischer Netzwerke und die Entwicklung einer fortgeschrittenen, holistischen Kontrollschicht auf Grundlage von SDN dar. Um das hybride elektronisch-optische Netzwerk zu realisieren, wurden neue Hardwarekomponenten entwickelt. Diese Geräte umfassen Netzwerkkarten für Server, Top-of-Rack-Switches und sogenannte POD-Switches. Die Kontrolllogik des NEPHELE-Netzwerks wird auf Grundlage von SDN in einem Cloud-Orchestrator implementiert. Die bedeutendsten Ergebnisse des Forschungsprojekts sind ein skalierbares Netzwerk mit stark verbesserter Energieeffizienz sowie die Steigerung der Kosteneffizienz. Zusätzlich ermöglicht das NEPHELE-Netzwerk, Rechen- und Speichernetze in einem gemeinsamen Netzwerk zu konsolidieren. Die Forschungsergebnisse des NEPHELE-Projektes sind dabei nicht auf ein einzelnes Rechenzentrum beschränkt. Die NEPHELE-Technologie ermöglicht auch Dienstgütegarantien zwischen Endknoten in föderierten Rechenzentren. Das Projekt wurde durch das Institute of Communication and Computer Systems (ICCS) der School of Electrical and Computer Engineering der National Technical University of Athens (NTUA) koordiniert. Partner im Konsortium waren Seagate Systems (UK), die GWDG, Mellanox Technologies (Israel), Nextworks (Italien), die University of Patras (Griechenland) und Interoute (Italien).

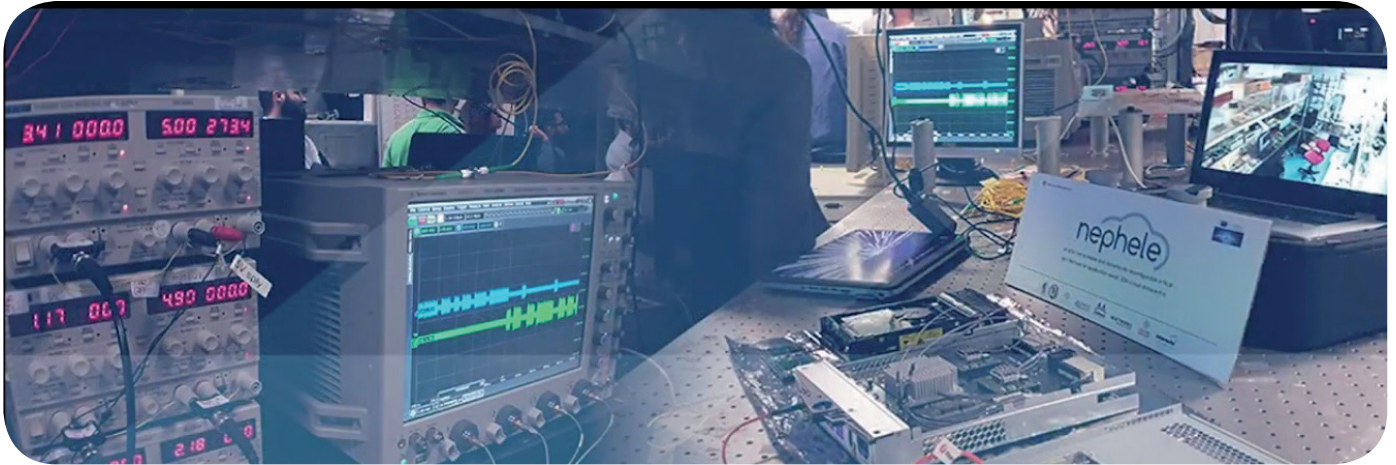


Figure 1: A snapshot of test bed from NEPHELE Demo review [2]

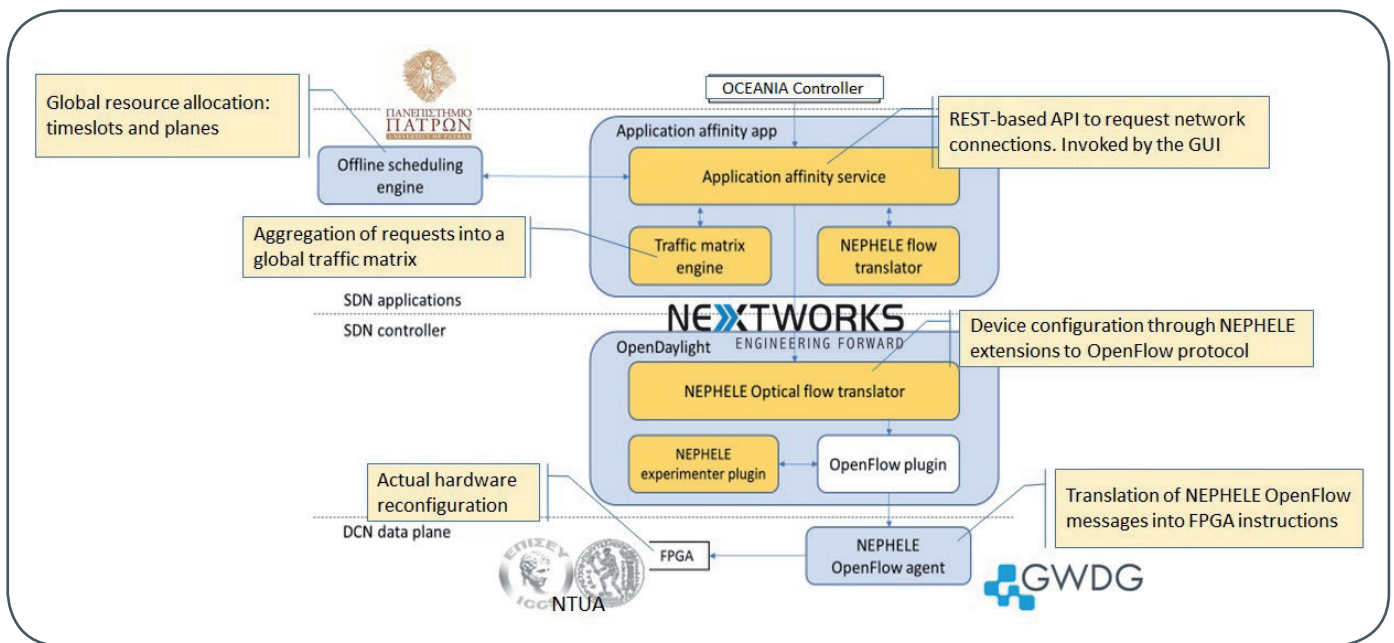


Figure 2: NEPHELE's control plane overview [3]

CONTROL PLANE VIEW

Figure 2 depicts a control plane view of NEPHELE Intra DC demo. It is comprised of three major components. A scheduler, a SDN controller and a variety of OpenFlow agents. The heart of the control plane is the network controller, named as OCEANIA. OCEANIA is a prototype developed on top of an open source OpenDaylight SDN controller, which is mainly used to dynamically configure the traffic connections at the data plane devices using the Open Flow commands. The configuration policy is fed into the OCEANIA controller by the NEPHELE scheduling engine, which continuously computes the current traffic matrix collectively based on the overall network condition and the new traffic requests generated by the cloud applications. The Open Flow commands by the controller are sent to the device specific Open Flow agents, which translates the Open Flow instructions to the corresponding FPGA schedules.

DATA PLANE VIEW

Figure 3 depicts a network view of NEPHELE Intra DC demo. It is a hybrid optical/electrical network comprises of four optical

Key Facts

Duration: 01.02.2015 – 31.01.2018 (36 months)

Programme: H2020

Partners: 7 Partners across Europe and Israel: Institute of Communications and Computer Systems (ICCS) of the National Technical University of Athens (NTUA; Greece), GWDC (Germany), Seagate Systems (UK), Mellanox Technologies (Israel), Nextworks (Italy), University of Patras (Greece), Interoute (Italy)

Funding: H2020-ICT-2014

EC Contribution: Grant amount of 3 million EUR

Website: <http://www.nepheleproject.eu>



Co-funded by the Horizon 2020
Framework Programme of the European Union

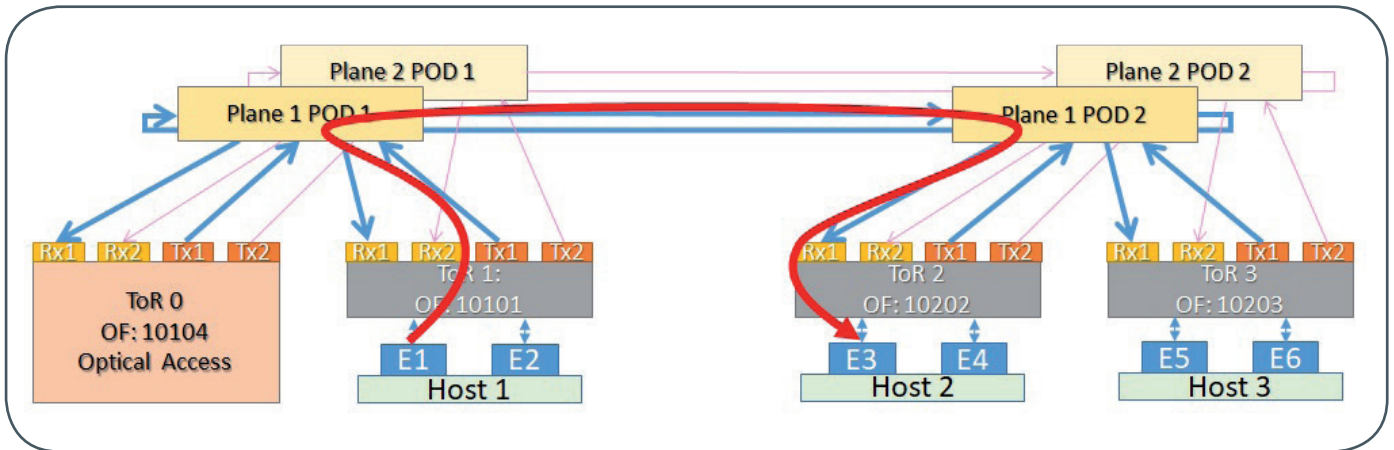


Figure 3: NEPHELE's data plane overview [4]

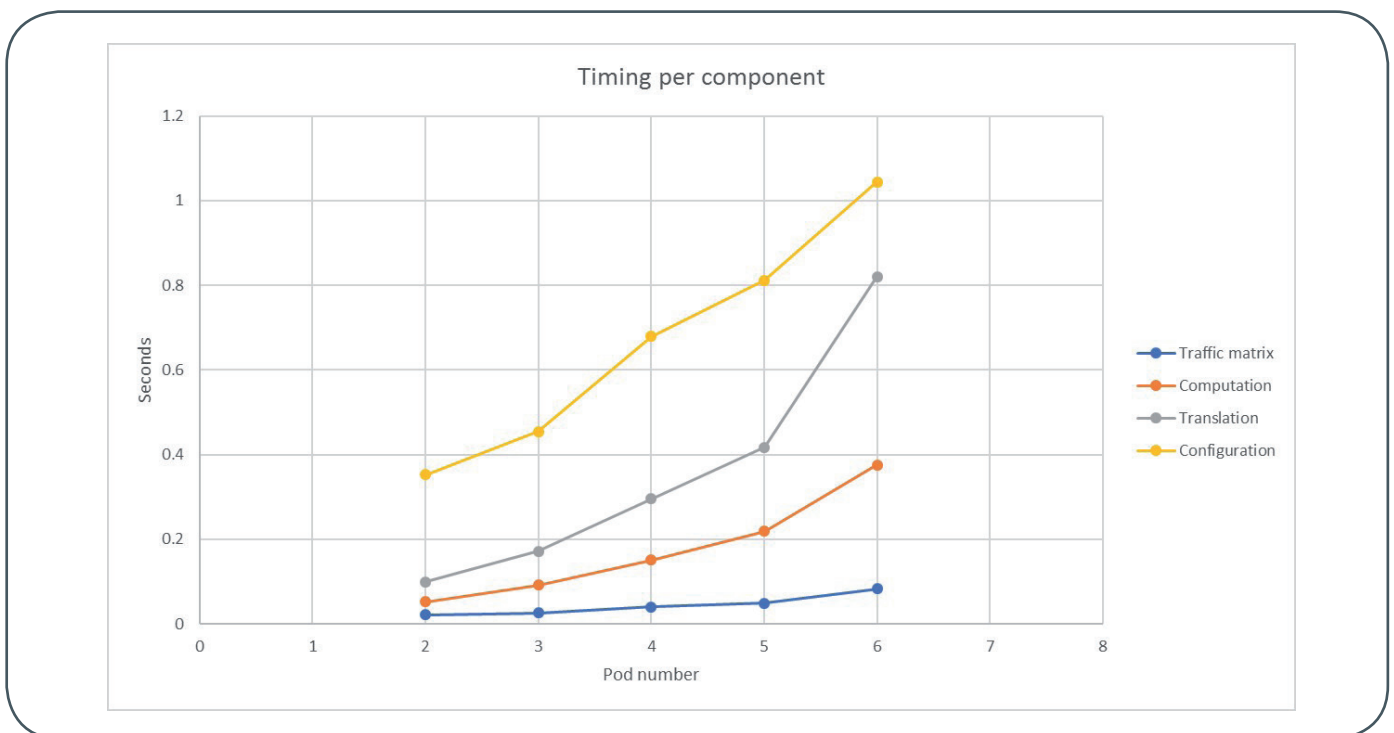


Figure 4: Processing times of OCEANIA components [3]

SCENARIO	2 PODS	3 PODS	4 PODS	5 PODS	6 PODS
Number of Planes	2	2	2	2	2
Number of Wavelengths = Number of ToRs per POD	80	80	80	80	80
Number of switches controlled by OCEANIA	164	246	328	410	492

Table 1: Test scenarios for OCEANIA intra-DC controller [3]

switches (as mentioned here POD switches) and three electrical ToR (top of the rack) switches. Each ToR is connected with one or more POD. So, for example, short lived packet flow such as intra ToR traffic between the E1 and E2 servers will be handled by the electrical packet switched network, whereas the long distanced, such as Inter POD traffic between E1 and E5 will be handled by optical circuit switching.

Here, the red line (in figure 3) shows an exemplary connection path that has been demonstrated in the demo review at NTUA lab in Athens. A connection is requested in the demo between ToR 1 server 1 (E1) and ToR 2 server 1 (E3), with 5 Gbit/s bandwidth. The traffic matrix gets updated with the request, and sent to the off-line scheduling engine. The scheduling engine has to allocate 40

timeslots (half of the total bandwidth of a link) for this connection, and it chooses plane 1, timeslots 20-59. The Flow translator sends the corresponding Open Flow configuration to the device agents. The device agents translate the messages building the new schedule, and forward it to the FPGAs. At the next reconfiguration period, the FPGAs apply the new schedule on the hardware, establishing the end-to-end light-path.

PERFORMANCE TESTING OF OCEANIA CONTROLLER

The OCEANIA controller has been extensively tested under different test scenarios, presented in table 1 with around 240

requests. The performance is mainly computed based on the execution times taken by various OCEANIA operations under load, namely the generation of traffic matrix, computation of network allocation and translating to Open Flow rules and, finally pushing the Open Flow configurations to the data plane devices. The instantiation time of an individual flow has not been explicitly measured, as it does not really effects the overall network optimizations.

Figure 4 presents a summary of the test results. The major portion to the processing time is taken by the configuration of network devices, which scales linearly with the number of PODs as compared to other processing times. For instance, the other controller operations entail a sharp increase in their processing times after the number of PODs is taken above 5. This can be seen as a fact of deprivation of VM resources at the corresponding level.

ACKNOWLEDGEMENT

This project had received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645212 (NEPHELE).

DISCLAIMER

The results of this article reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

- [1] Cisco, Cisco Global Cloud Index: Forecast and Methodology, 2012 -2017 White Paper.
- [2] NEPHELE Project website: <http://www.nepheleproject.eu/>
- [3] Aziz, Muzzamil et. al, D5.3: Fully functional intraDC SDN controller framework prototype, 30th September 2016.
- [4] Mellanox technologies, NEPHELE Intra-DC scenario demonstrator, 20th October 2017. ●

Personalia

NEUER MITARBEITER ROBERT WILLER

Seit dem 15. Juli 2018 verstärkt Herr Robert Willer die Arbeitsgruppe „Nutzerservice und Betriebsdienste“ (AG H) als wissenschaftliche Hilfskraft. Nach Abschluss seines Bachelor-Studiums studiert Herr Willer an der Georg-August-Universität Göttingen im Master-Studiengang Physik. Sein Aufgabenschwerpunkt bei der GWDG wird in der Kundenbetreuung und Unterstützung bei der Administration des eduroam-Dienstes der GWDG liegen. Weitere Aufgabenschwerpunkte sind die Betreuung und Unterstützung bei der Administration des Voucher- und Guest On Campus-Dienstes. Herr Willer ist per E-Mail unter robert.willer@gwdg.de und telefonisch unter 0551 201-1843 zu erreichen. Körner



NEUER MITARBEITER MARCO MEDEN

Seit dem 15. August 2018 verstärkt Herr Marco Meden die Arbeitsgruppe „Anwendungs- und Informationssysteme“ (AG A) als studentische Hilfskraft. Er studiert zurzeit Angewandte Informatik mit Schwerpunkt Bioinformatik an der Georg-August-Universität Göttingen. Seine Aufgabenschwerpunkte bei der GWDG werden im Bereich des Apple-Beratungszentrums und bei CrashPlan PROe liegen. Herr Meden ist per E-Mail unter marco.meden@gwdg.de und telefonisch unter 0551 201-1830 zu erreichen. Heider

NEUE AUSZUBILDENDE DANIEL HEISE, JAKOB HOLLWEDEL UND NILS BUCZIOR

Am 1. August 2018 hat Herr Daniel Heise seine Ausbildung zum Fachinformatiker in der Fachrichtung Systemintegration bei der GWDG begonnen. Herr Heise hat zuvor das Berufliche Gymnasium Technik mit dem Schwerpunkt Informationstechnik an der BBS II in Göttingen besucht und dort sein Abitur erworben. Herr Heise ist per E-Mail unter daniel.heise@gwdg.de und telefonisch unter 0551 201-2183 zu erreichen. Herbold



Darüber hinaus haben am 1. August 2018 zwei weitere Auszubildende ihre Ausbildung zum Fachinformatiker in der Fachrichtung Anwendungsentwicklung in der Arbeitsgruppe „Basisdienste und Organisation“ (AG O) begonnen. Herr Jakob Hollwedel (E-Mail: jakob.hollwedel@gwdg.de, Tel.: 0551 201-2191) hat zuletzt die Fachhochschulreife im Bereich Technik erlangt und im Rahmen eines einjährigen Praktikums das Berufsbild des Fachinformatikers kennengelernt. In seiner Freizeit hat Herr Hollwedel bereits Erfahrungen mit der Programmiersprache Java sowie im Bereich der Web-Entwicklung gesammelt.



Herr Nils Buczior (E-Mail: nils.buczior@gwdg.de, Tel.: 0551 201-2190) wechselt im zweiten Ausbildungsjahr zur GWDG und bringt für die Ausbildung Kenntnisse in den Programmiersprachen Java und C# mit, die er sich bereits im Rahmen eines Informatik-Studiums und im ersten Ausbildungsjahr angeeignet hat. Pohl





INFORMATIONEN:
support@gwdg.de
0551 201-1523

September bis
Dezember 2018

Kurse

KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
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
USING THE GWDC SCIENTIFIC COMPUTE CLUSTER – AN INTRODUCTION	Dr. Boehme, Dr. Ehlers	22.10.2018 9:30 – 16:00 Uhr	15.10.2018	4
PARALLELRECHNER-PROGRAMMIERUNG MIT MPI	Prof. Haan	23.10. – 24.10.2018 9:15 – 17:00 Uhr	16.10.2018	8
PHOTOSHOP FÜR FORTGESCHRITTENE	Töpfer	23.10. – 24.10.2018 9:30 – 16:00 Uhr	16.10.2018	8
PROGRAMMING WITH CUDA – AN INTRODUCTION	Prof. Haan	06.11.2018 9:15 – 17:00 Uhr	30.10.2018	4
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

KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
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