

GWDG NACHRICHTEN 08|24

Neue Großformatdrucker

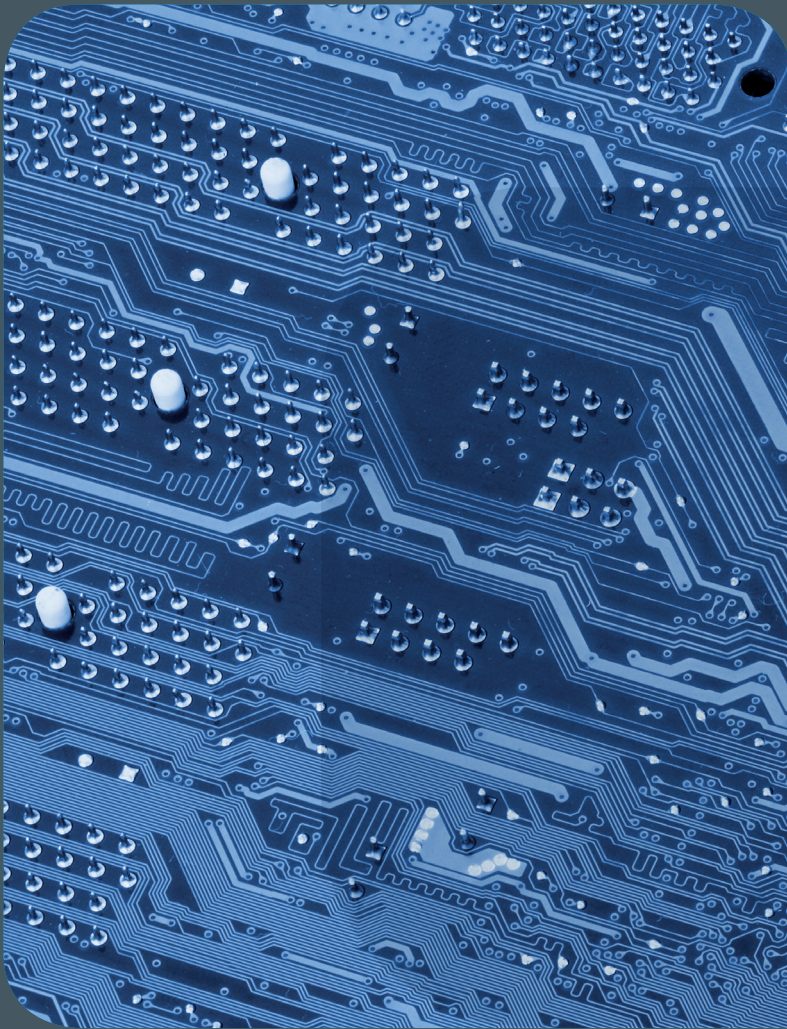
Performance Engineering
and Analysis Tools for
GWDG HPC Systems

Ausbildungsinhalte
„Algorithmen und
Datenstrukturen“

KonKIS 2024

ZEITSCHRIFT FÜR DIE KUNDEN DER GWDG





GWDG NACHRICHTEN

08|24 Inhalt

.....

**4 HP DesignJet T1700ps – drei neue
Großformatdrucker für die externen
Druckerstandorte** **6 Performance Engineering
and Analysis Tools for GWDG HPC Systems**
**15 Ausbildungsinhalte „Algorithmen und
Datenstrukturen“ für Fachinformatiker*innen**
Anwendungsentwicklung **21 Kurz & knapp**
22 Stellenangebote **24 Personalia** **25 Academy**

Impressum

.....

Zeitschrift für die Kunden der GWDG

ISSN 0940-4686
47. Jahrgang
Ausgabe 8/2024

Erscheinungsweise:
10 Ausgaben pro Jahr

www.gwdg.de/gwdg-nr

Auflage:
550

Fotos:

© peterschreiber.media - stock.adobe.com (1)
© Nataliya Kalabrina - Fotolia.com (14)
© nito - Fotolia.com (22-23)
© Robert Kneschke - Fotolia.com (25)
© Momius - Fotolia.com (27)
© MPI-NAT-Medienservice (3, 24 unten)
© GWDG (2, 24)

Herausgeber:

Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen
Burckhardtweg 4
37077 Göttingen
Tel.: 0551 39-30001
Fax: 0551 39-130-30001

Redaktion und Herstellung:

Dr. Thomas Otto
E-Mail: thomas.otto@gwdg.de

Druck:

Kreationszeit GmbH, Rosdorf



Prof. Dr. Ramin Yahyapour
ramin.yahyapour@gwdg.de
0551 39-30130

*Liebe Kund*innen und Freund*innen der GWDG,*

*der effiziente Umgang mit modernen Hochleistungsrechnern erfordert viel Verständnis im Zusammenspiel von Software und Hardware. Das hat sich über die Jahrzehnte nicht geändert. Die verfügbare Hardware hat eher an Vielfalt und Komplexität gewonnen, so dass der Prozess der Software-Optimierung immer wichtiger geworden ist. Erfreulicherweise gibt es dazu Unterstützung. Es gibt diverse Werkzeuge und Plattformen, die bei der Leistungsoptimierung unterstützen können. In dieser Ausgabe finden Sie einen Artikel, in dem beispielhaft die verschiedenen Schritte und möglichen Werkzeuge vorgestellt werden. Ebenso gibt es Unterstützung in Form von Schulung und Beratung. Die GWDG ebenso wie die anderen Nationalen Hochleistungsrechenzentren haben den Auftrag, auch Kompetenzvermittlung und Beratung zum Umgang mit Hochleistungsrechnern zu übernehmen. Dazu finden Sie bei uns entsprechende Angebote. Dies wird immer wichtiger, da solche Rechnersysteme nicht mehr nur für wenige Expert*innen relevant sind, sondern auch immer mehr Anwender*innen in neuen Disziplinen finden.*

Ramin Yahyapour

GWDG – IT in der Wissenschaft

HP DesignJet T1700ps – drei neue Großformatdrucker für die externen Druckerstandorte

Text und Kontakt:
Uwe Nolte
uwe.nolte@gwdg.de
0551 39-30325

Seit Kurzem stehen drei neue leistungsfähige Großformatdrucker des Typs HP DesignJet T1700ps an den drei externen Druckerstandorten in den beiden Bereichsbibliotheken Medizin und Physik sowie im LRC/SUB zur Verfügung. Sie ermöglichen aufgrund ihres Sechs-Farbsystems, innovativer Druckkopftechnik sowie der eingebauten „Adobe PDF Print Engine“ qualitativ hochwertige Ausdrücke.

EINLEITUNG

Die neuen Großformatdrucker des Typs HP DesignJet T1700ps sind mit HP-Bright-Office-Tinten ausgestattet. Das ermöglicht laut Hersteller „brillante Farben und gestochen scharfe Bildqualität – auch auf Normalpapier“. Ein Tintensatz besteht aus den Farben Cyan, Magenta, Gelb, Matt- und Fotoschwarz sowie Grau. Außerdem sind alle drei Geräte mit der „Adobe PDF Print Engine“ ausgestattet. Dabei handelt es sich um eine Hochgeschwindigkeits- und High-Fidelity-Druckplattform von Adobe. Version 4 erweitert die innovative APPE-Technologie um neue Leistungsmerkmale und bahnbrechende Rendering-Funktionen. Darüber hinaus ermöglichen sein 128-GByte-Speicher und die 500-GB-HDD problemloses Drucken selbst komplexer Druckdateien. Der Drucker kann Dokumente in den Formaten HP-GL/2, HP-RTL, TIFF (unkomprimiert!), JPEG, CALS G4 und HP PCL 3 GUI, PostScript und PDF verarbeiten. Daher können Sie unter UNIX JPEG-Dateien direkt per *lpr*-Befehl an den Drucker schicken.

Die neuen Drucker können unter Windows über folgende Warteschlangen (Queues) angesprochen werden:

- Standort LRC/SUB
`\\gwdg-print3.gwdg.de\bcip4s` oder `\\gwdg-print3.gwdg.de\bcixls`
- Standort Bereichsbibliothek Medizin
`\\gwdg-print3.gwdg.de\mcip4s` oder `\\gwdg-print3.gwdg.de\mcixls`
- Standort Bereichsbibliothek Physik
`\\gwdg-print3.gwdg.de\pcip4s` oder `\\gwdg-print3.gwdg.de\pcixls`

Die alten Queuenamen mit der „13“ und der „79“ am Ende werden für eine Übergangszeit weiterhin gültig bleiben.

Auf unseren UNIX-Dialogservern (z. B. `login.gwdg.de`, `gwdg20.gwdg.de` und `gwdg60.gwdg.de`) stehen noch die Warteschlangen `bcip4s`, `mcip4s` sowie `pcip4s` zur Verfügung, die ein

DIN-A4-Dokument automatisch auf das Format DIN A0 hochskalieren. Der Drucker wird mit gestrichenem 120-g/m²-Normalpapier betrieben. Die Rollenbreite beträgt 106 cm. Pro Ausdruck werden 0,36 AE (das entspricht 12,00 EUR) vom jeweiligen Institutskontingent abgebucht. An den genannten Standorten ist zudem eine Schneidemaschine vorhanden, mit der Sie DIN-A0-Poster zuschneiden können.

Die folgenden drei Tabellen geben eine Übersicht über die Queues der Posterdrucker an den drei externen Standorten. ●

HP DesignJet T1700ps - Three New Large Format Printers for the External Printer Locations

Three new high-performance large format printers of the type HP DesignJet T1700ps are now available at the three external printer locations in the two departmental libraries Medizin and Physics as well as in the LRC/SUB. Their six-color system, innovative print head technology and built-in „Adobe PDF Print Engine“ enable high-quality printouts. These printers can be used under Windows with the following printer queues:

- Location LRC/SUB
`\\gwdg-print3.gwdg.de\bcip4s` or `\\gwdg-print3.gwdg.de\bcixls`
- Location Departmental Library Medicine
`\\gwdg-print3.gwdg.de\mcip4s` or `\\gwdg-print3.gwdg.de\mcixls`
- Location Departmental Library Physics
`\\gwdg-print3.gwdg.de\pcip4s` or `\\gwdg-print3.gwdg.de\pcixls`



1_HP DesignJetT1700ps (© Hewlett-Packard)

QUEUE	BESCHREIBUNG
mcip1s (Alias-Namen: mcip1s17, mcip1s79)	„Normale“ Warteschlange; kein Skalieren der Druckdaten; Eingabeformat PostScript via GWDGPS-Treiber; Ausgabeformat an Drucker: JPEG; Rip mittels ghostscript auf print.gwdg.de; Warteschlange für Windows-, UNIX- und Mac-Systeme.
mcip4s (Alias-Namen: mcip4s17, mcip4s79)	Hochskalieren der Druckdaten auf 400 %; Eingabeformat PostScript; Ausgabeformat an Drucker: JPEG; Rip mittels ghostscript auf print.gwdg.de; Warteschlange nur auf UNIX-Dialog-Servern nutzbar
mcix1s (Alias-Namen: mcix1s17, mcix1s79)	Spezial-Warteschlange; kein Skalieren der Druckdaten; Eingabeformat: PostScript mittels Original HP-Druckertreiber via gwdg-print3.gwdg.de; Rip mittels druckerinterner „Adobe PDF Print Engine“

Tabelle 2: Queues für den HP DesignJetT1700ps in der Bereichsbibliothek Medizin

QUEUE	BESCHREIBUNG
bcip1s (Alias-Namen: bcip1s17, bcip1s79)	„Normale“ Warteschlange; kein Skalieren der Druckdaten; Eingabeformat PostScript via GWDGPS-Treiber; Ausgabeformat an Drucker: JPEG; Rip mittels ghostscript auf print.gwdg.de; Warteschlange für Windows-, UNIX- und Mac-Systeme.
bcip4s (Alias-Namen: bcip4s17, bcip4s79)	Hochskalieren der Druckdaten auf 400 %; Eingabeformat PostScript; Ausgabeformat an Drucker: JPEG; Rip mittels ghostscript auf print.gwdg.de; Warteschlange nur auf UNIX-Dialog-Servern nutzbar
bcix1s (Alias-Namen: bcix1s17, bcix1s79)	Spezial-Warteschlange, kein Skalieren der Druckdaten; Eingabeformat: PostScript mittels Original HP-Druckertreiber via gwdg-print3.gwdg.de; Rip mittels druckerinterner „Adobe PDF Print Engine“.

Tabelle 1: Queues für den HP DesignJetT1700ps im LRC/SUB

QUEUE	BESCHREIBUNG
pcip1s (Alias-Namen: pcip1s17, pcip1s13)	„Normale“ Warteschlange; kein Skalieren der Druckdaten; Eingabeformat PostScript via GWDGPS-Treiber; Ausgabeformat an Drucker: JPEG; Rip mittels ghostscript auf print.gwdg.de; Warteschlange für Windows-, UNIX- und Mac-Systeme
pcip4s (Alias-Namen: pcip4s17, pcip4s13)	Hochskalieren der Druckdaten auf 400 %; Eingabeformat PostScript; Ausgabeformat an Drucker: JPEG; Rip mittels ghostscript auf print.gwdg.de; Warteschlange nur auf UNIX-Dialog-Servern nutzbar
Pcix1s (Alias-Namen: pcix1s17, pcix1s79)	Spezial-Warteschlange, kein Skalieren der Druckdaten; Eingabeformat: PostScript mittels Original HP-Druckertreiber via gwdg-print3.gwdg.de; Rip mittels druckerinterner „Adobe PDF Print Engine“

Tabelle 3: Queues für den HP DesignJetT1700ps in der Bereichsbibliothek Physik

Register now!



Konferenz der deutschen KI-Servicezentren

Sept 18.-19., 2024



KonKIS 24

GEFÖRDERT VOM











Performance Engineering and Analysis Tools for GWDC HPC Systems

Text and Contact:
Dr. Jack Ogaja
jack.ogaja@gwdg.de

Software performance engineering is important for maximizing the benefits of complex HPC systems. The process requires tools developed to gain insights into the performance limits of different components of HPC hardware. At the GWDC we employ sophisticated tools to help customers solve their software performance issues and use the available HPC resources efficiently and effectively. In this article we present usage of selected performance tools available for GWDC customers and show some of their differences.

INTRODUCTION

Performance engineering, analysis and optimization is the process of investigating and improving the design, functionality and applicability of software to exploit maximum performance of the hardware infrastructure in which they are used. This process is critical for HPC service provision especially considering the huge amount of resources spent on assembling and operating large computing systems with complex microprocessors and memory architectures. The process involves mainly three steps: application instrumentation, run-time measurements of key events, and visual analysis of profiles and events traces.

At GWDC, we support customers on both local (Scientific Compute Cluster – SCC) HPC systems and national NHR HPC systems with their software performance issues using some of the most sophisticated tools available. In this article we show basic usage of performance engineering tools currently available for users of the GWDC HPC systems, what insights can be gained from each, and how they differ.

LIKWID

LIKWID (Like I Knew What I'm Doing) is a collection of command-line utilities used for performance optimization, modeling, and architecture analysis. Developed by Erlangen Regional Computing Center (RRZE), LIKWID helps in exploring processor architecture by revealing the underlying thread and cache topology which can be used to enforce task-affinity. The suite also includes tools to measure energy consumption and different metrics based on hardware performance counters. It can also be used to manipulate processor frequencies. LIKWID is open-source and supports x86 (Intel and AMD) and GPUs (Nvidia and AMD) architectures.

Example: Node-Level Performance Analysis Using LIKWID

Compute nodes are computer units with topological layouts as shown in Figures 1 and 2, consisting of microprocessors mounted

on sockets (the blue blocks) interconnected via high-speed low-latency communication channels (QPI or UPI). Within the sockets are multiple processor cores with basic units including Floating Point and Arithmetic units (FPU and ALU), memory registers, control units and multilevel caches making the socket a powerful multi-core node component. Each socket has its main memory accessible by all processors inside the compute node.

For Nvidia GPU nodes, a custom high-speed interface,

Performance Engineering und Analyse-Tools für die HPC-Systeme der GWDC

Performance Engineering, Analyse und Optimierung ist der Prozess der Untersuchung und Verbesserung des Designs, der Funktionalität und der Anwendbarkeit von Software, um die maximale Leistung der Hardware-Infrastruktur, in der sie eingesetzt wird, zu nutzen. Dieser Prozess ist für die Bereitstellung von HPC-Diensten von entscheidender Bedeutung, vor allem wenn man bedenkt, wie viele Ressourcen für die Zusammenstellung und den Betrieb großer Rechensysteme mit komplexen Mikroprozessoren und Speicherarchitekturen aufgewendet werden. Die Leistungsanalyse und -optimierung von HPC-Anwendungen umfasst im Wesentlichen drei Schritte: Instrumentierung der Anwendung, Laufzeitmessungen von Schlüsselereignissen und visuelle Analyse von Profilen und Ereignis Spuren. Bei der GWDC unterstützen wir Kund*innen sowohl auf lokalen (Scientific Compute Cluster – SCC) HPC-Systemen als auch auf nationalen NHR-HPC-Systemen bei bestehenden Software-Leistungsproblemen mit einigen der anspruchsvollsten verfügbaren Tools. In diesem Artikel zeigen wir die grundlegende Verwendung von Performance-Engineering-Tools, die derzeit für Nutzer*innen der HPC-Systeme der GWDC zur Verfügung stehen, welche Erkenntnisse aus den einzelnen Tools gewonnen werden können und wie sie sich unterscheiden.

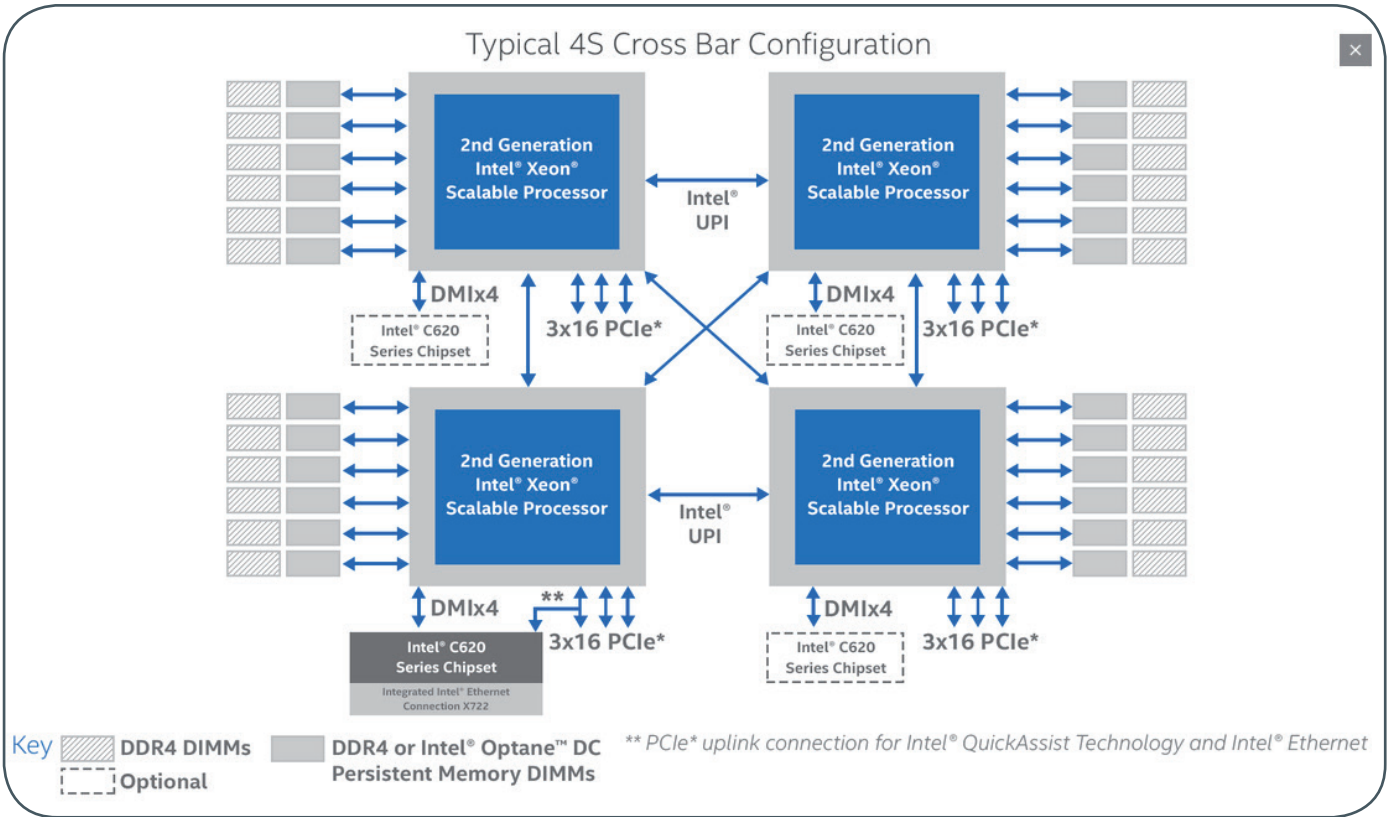


Figure 1: A compute node configuration courtesy of Intel

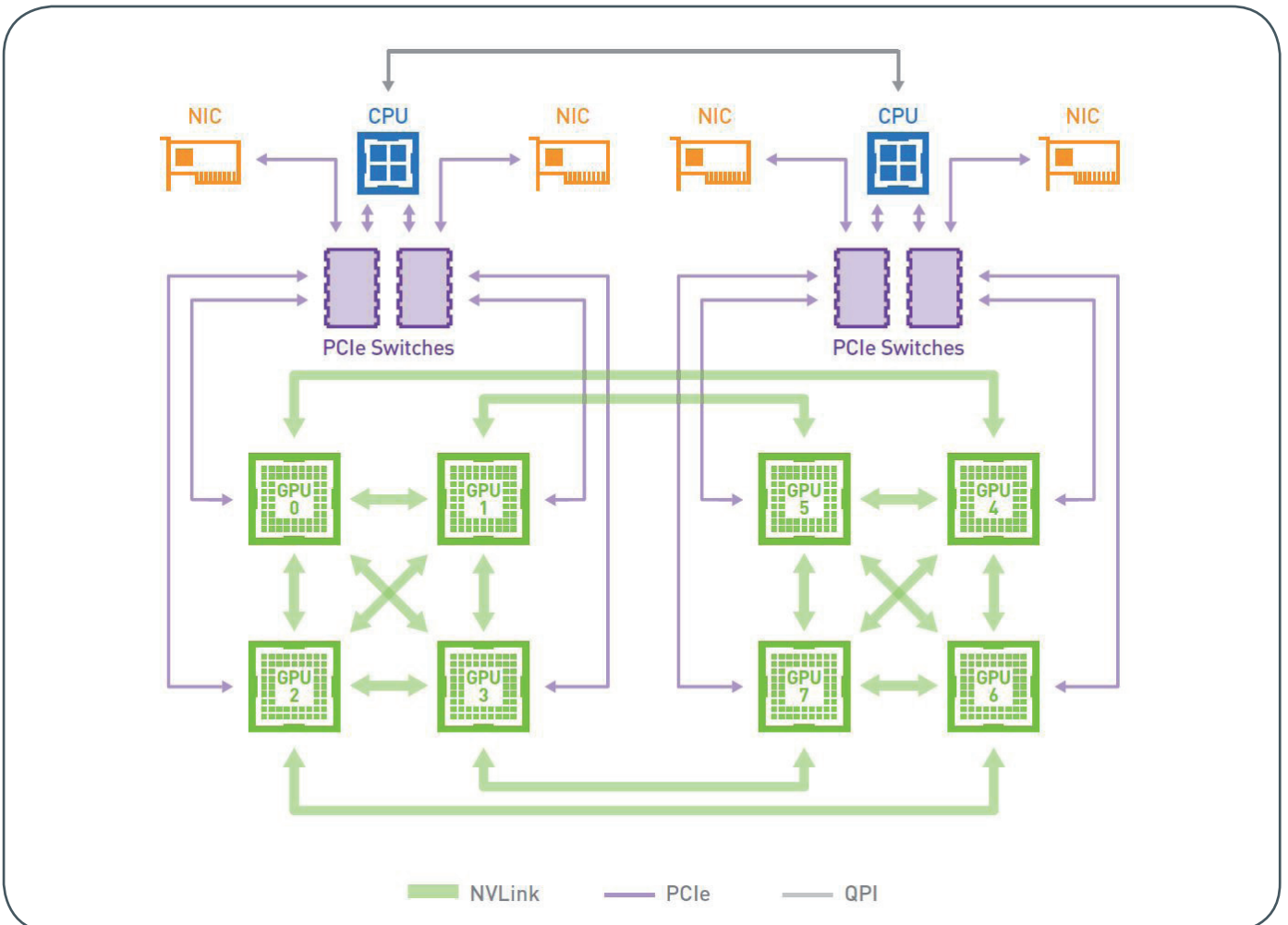


Figure 2: A GPU node configuration courtesy of Nvidia

NVLink, is used to connect GPUs within the nodes.

We can use LIKWID to characterize performance limits of a node by measuring the compute intensity and memory bandwidth when executing a parallel application. In such analysis multiple tools within LIKWID are used: *likwid-topology* is applied to understand the node architecture, and *likwid-perfctr* for profiling the application at run-time. We use these tools to understand the interaction between the application and the hardware. The latter tool configures and reads hardware performance data.

One way of characterizing node-level performance is using Roofline model to determine compute and memory bounds of the node when executing a program. LIKWID has the tools to measure necessary metrics for Roofline modeling.

In the following paragraphs, we show usage of selected tools among available with LIKWID tool-set. All applications and software provided for GWDG HPC clusters including the performance tools are available via environment modules. Before use, they must be loaded into your compute environment.

Node Architecture Information

The *likwid-topology* tool is used to explore hierarchical topology of compute nodes (example depicted in Figures 1 and 2). Depending on the level of complexity of the compute node, the following levels of topology can be exposed using the tool:

- Thread topology: How processor IDs map on physical compute resources
- Cache topology: How processors share the cache hierarchy
- Cache properties: Detailed information about all cache levels
- NUMA topology: NUMA (Non-Uniform Memory Access) domains and memory sizes
- GPU topology: GPU information

Application Performance Profiling

LIKWID comes with an outstanding feature, performance group, for profiling applications. Performance groups, which are preconfigured with useful event sets and derived metrics, organizes and combines micro-architecture events and counters with e.g. run-time and clock speed. Performance groups provide a set of derived metrics which are read on the fly without compilation by command-line selection for efficient analysis. The feature also provides a set of derived metrics for efficient analysis. The *likwid-perfctr* tool uses available hardware counters to measure events that characterises the interaction between software and hardware. A light-weight marker API can be applied for code instrumentation

To profile a program, the following command can be executed in the command line of a compute node (see Code 1).

```
1 | likwid-perfctr -g <performance group of interest> myProgram
```

Code 1

The *likwid-perfctr* tool can also be used as a wrapper (no modification in the application) or by adding *Marker_API* functions inside the program's code. Since *likwid-perfctr* measures all events on specified microprocessors, it is necessary for processes and threads to dedicate resources. This can be done by pinning the application manually or using the built-in functionality of *likwid-pin* to enable thread-, core-, or task-affinity. *likwid-pin*

explicitly supports Pthread and the OpenMP implementations of Intel and GNU gcc.

Micro-benchmarking

LIKWID also contains a micro-benchmarking framework which enables:

- Quantification of sustainable performance limits
- Separation of influences considering isolated instruction code snippets
- Reverse-engineering of processor features
- Discovery of hardware bugs

To run the micro-benchmarking, the following command can be executed (see Code 2).

```
1 | likwid-bench -a # reveals available options
2 | likwid-bench <options>
```

Code 2

SCORE-P

Score-P (Scalable Performance Measurement Infrastructure for Parallel codes) is a powerful tool for profiling and tracing events during execution of parallel applications.

With its support for different data formats to instrument, write and read trace data, it supports a number of analysis tools including Vampir, Scalasca and TAU. Data formats supported by Score-P include, Open Trace format 2 – OTF2, Cube4 profiling format and the Opari2 instrumenter.

Before generating profiles and trace data of applications using Score-P, it is necessary to instrument their codes. In GWDG HPC systems, this is done by first load the specific environment module before using the Score-P wrapper (see Code 3)

```
1 | $ module load scorep
```

Code 3

In the GWDG clusters, loading the Score-P module also enables links to both OTF2 and CUBE libraries.

Example: Automatic and Manual Instrumentation Using ScoreP

For automatic instrumentation use the *scorep* wrapper before the compiler, for example (see Code 4).

```
1 | $# Environment variable setup examples
2 |
3 | $ export SCOREP_ENABLE_TRACING=true
4 |
5 | $ scorep --user gcc -c test.c -o test
```

Code 4

For manual instrumentation the *--user* flag is added to the *scorep* command when compiling, i.e. (see Code 5).

```
1 | $ scorep --user gcc -o test test.c
```

Code 5

The environment variable *SCOREP_ENABLE_TRACING* allows tracing measurements. Current Score-P environment variables currently set can be checked by *score-p-info* (see Code 6).


```

1  $ scorep-info config-vars --full
2
3  #Output
4
5  SCOREP_ENABLE_PROFILING
6  Description: Enable profiling
7  Type: Boolean
8  Default: true
9
10 SCOREP_ENABLE_TRACING
11 Description: Enable tracing
12 Type: Boolean
13 Default: false
14
15 SCOREP_VERBOSE
16 Description: Be verbose
17 Type: Boolean
18 Default: false
19
20 .....

```

Code 6

To produce trace and profile files, the compiled program is executed normally.

Note: Heavy and complex applications should be profiled and their traces generated in compute nodes using resource schedulers e.g. SLURM to avoid burdening front-end nodes.

VAMPIR TOOL SUITE

Vampir Tool Suite is a collection of tools for analysing performance of parallel applications. It includes instrumentation, measurement and visualization tools. The tools give the user an insight into the dynamic run-time behaviour of their applications. The instrumentation and measurement tools are Score-P and Vampir-Trace - which is no longer under active development and will not be covered in this documentation, and the Vampir visualization tool.

With a powerful graphical framework which displays state diagram, statistics and timelines, Vampir (the visualization tool) uses run-time events collected by the instrumentation and measurements tools and libraries. Vampir has zooming and scrolling functionalities which allow parallel applications developers to analyse performance of the applications at different levels of details. Filtering capabilities for processes, functions, messages, and collective operations are also available to identify performance bottle-necks.

The tools can also be used to interactively navigate through the timelines of applications execution in accelerators. This enables an extensive performance analysis and optimization of hybrid programs written in CUDA, OpenACC, OpenCL, and PyCUDA.

Some features of Vampir include:

- Powerful zooming and scrolling in all displays
- Adaptive statistics for user selected time ranges
- Filtering of processes, functions, messages, collective operations
- Hierarchical grouping of threads, processes, and nodes
- Support of source code locations
- Integrated snapshot and printing for publishing
- Customizable displays
- Server: Vampir also offer a client-server model for a highly scalable distributed performance data visualization, and remote visualization of performance data.

Vampir in GWDG Systems

The following Vampir versions are currently available in the GWDG's clusters (see Table 1).

SYSTEM	VERSION	COMMENTS
Emmy	10.4.1	available as a module 'vampir/10.4.1'
SCC	10.4.1	available as a module 'vampir/10.4.1'

Table 1: Available Vampir versions in the GWDG's clusters

Example: A Quick Start in SCC Cluster

1. To use Vampir in the GWDG's clusters, the module must be loaded first to set the environment correctly.
2. Start Vampir
After executing the command *vampir* in the command line, the GUI will open with "Open Recent" dialogue box by default as shown in Figure 3. One the need to navigate to the OTF files, and select the desired file. The default timeline when the data is loaded is the master timeline from which other timelines, statistical charts, I/O summary and call tree can be viewed (see figure 4)

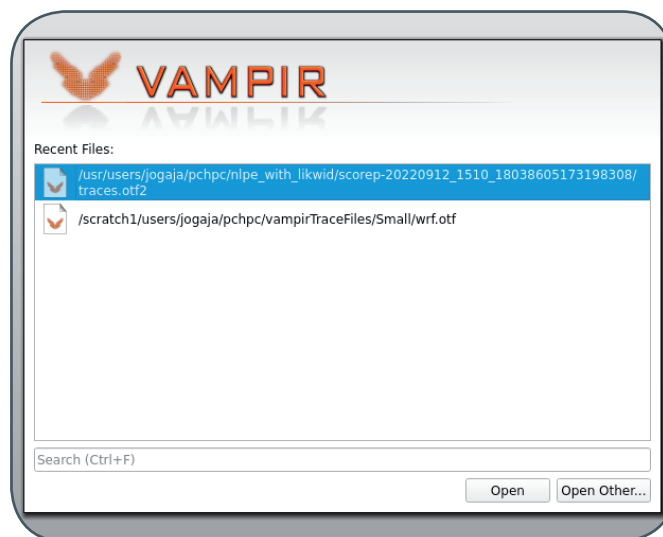


Figure 3: Vampir open dialogue box

NVIDIA NSIGHT SYSTEM

Nsight is a collections of GPU profiling tool set provided by Nvidia as part of developer tools. Nsight consist of three distinct tools, system, compute and graphics (see Figure 5).

- Nsight Systems: Used for analysing application algorithms system-wide
- Nsight Compute: Used for CUDA kernels analysis
- Nsight Graphics: For debugging and analysing graphics workloads

Nsight Systems

To identify the largest optimization opportunities a developer or user should start with a system-wide performance analysis of their applications. This can be done using Nsight Systems. Nsight Systems provides developers and users a system-wide visualization of an applications performance from which they can optimize bottlenecks to scale efficiently across any number of size of CPUs and GPUs from large servers to a SoC (System on a Chip).

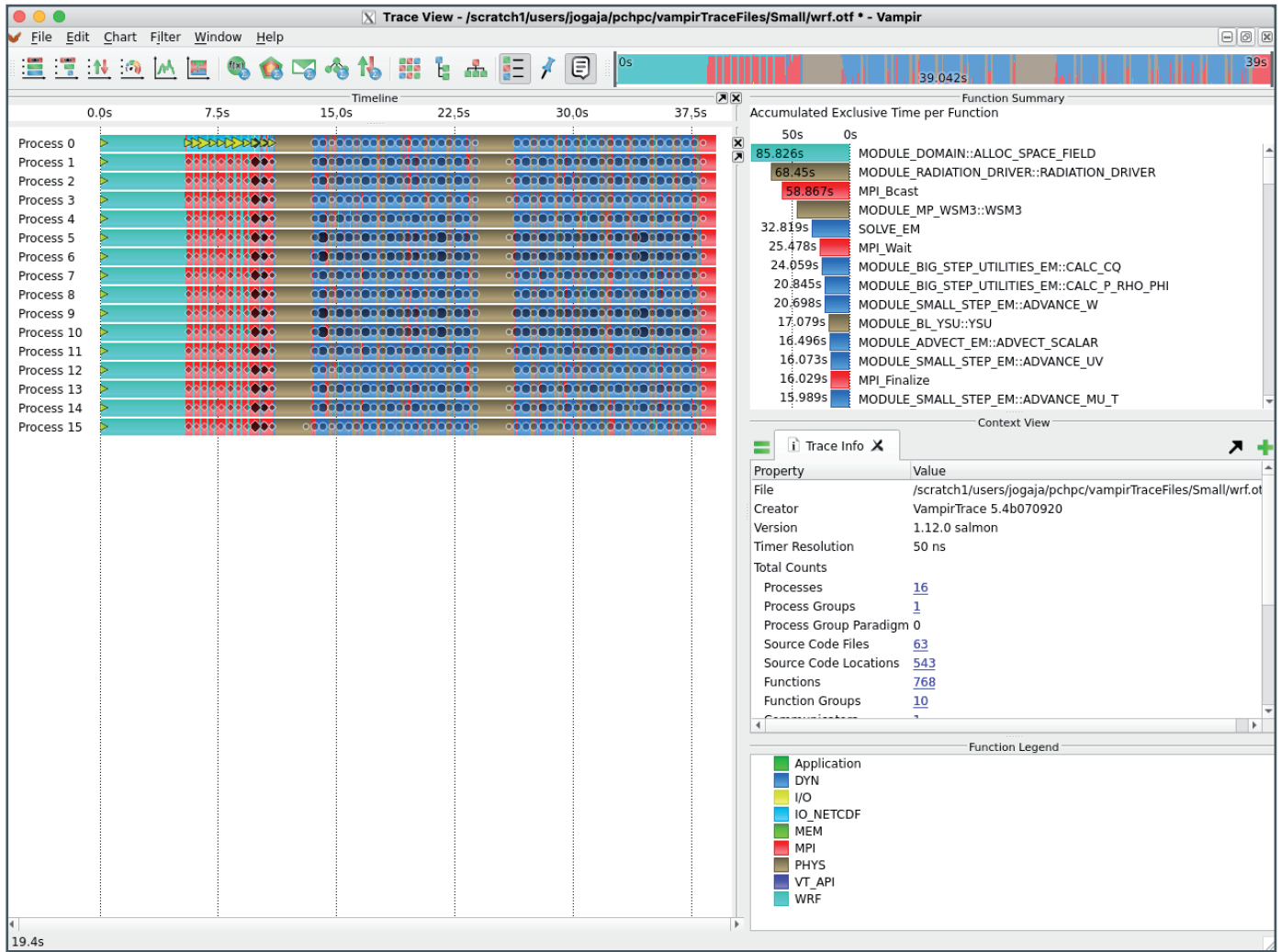


Figure 4: Master timeline for example trace files

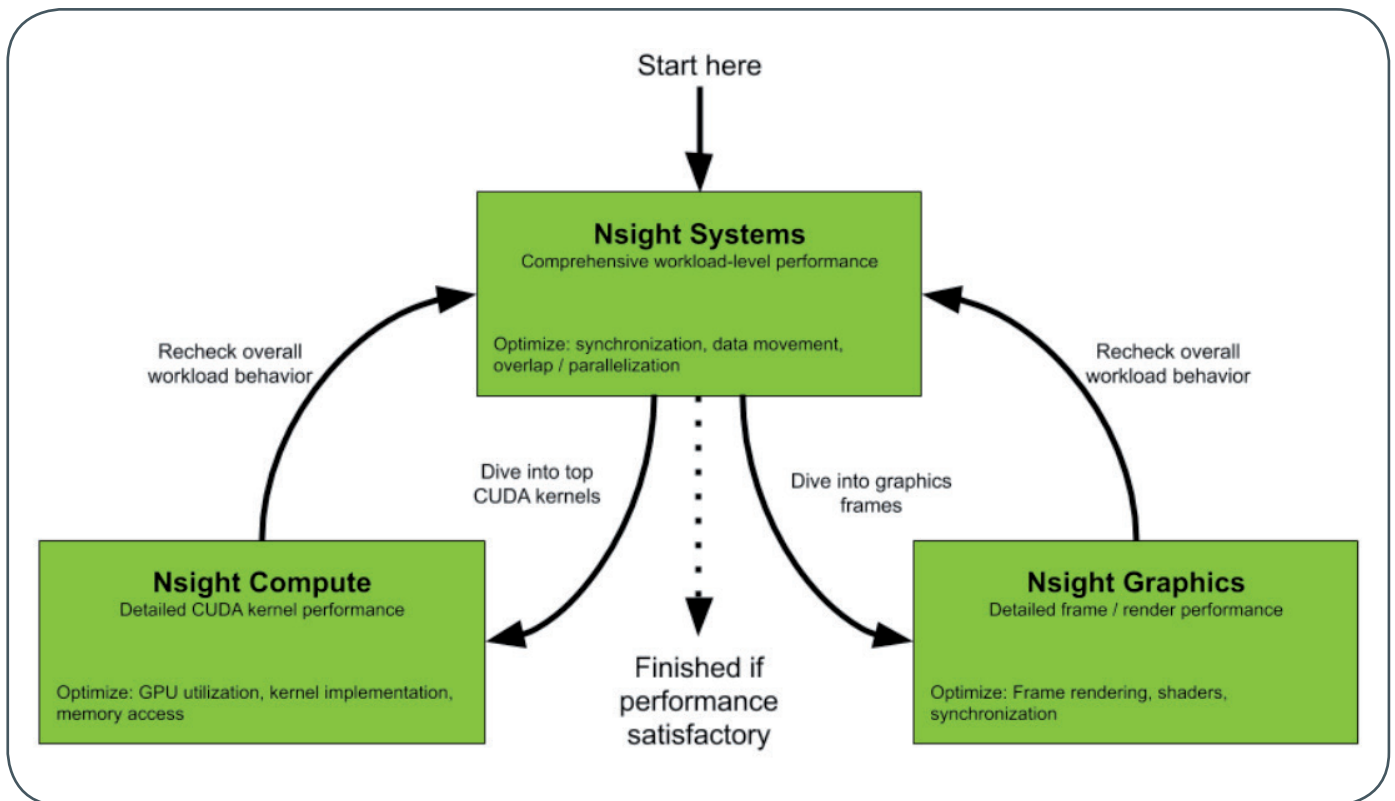


Figure 5: Flowchart for Nsight compute, system and graphics

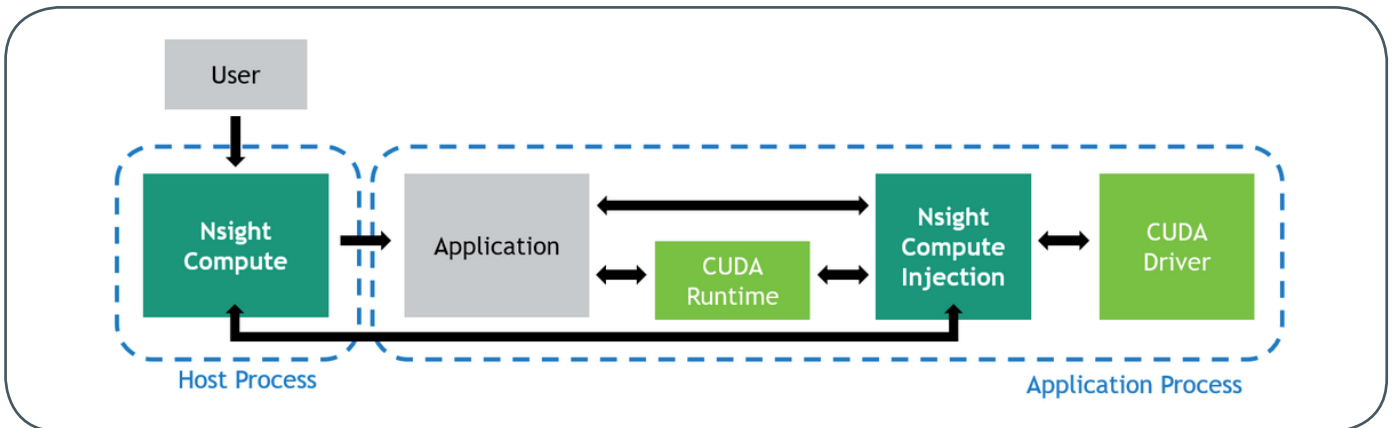


Figure 6: Profiling process using Nsight Compute

Nsight Compute

As an interactive kernel profiler for CUDA applications, Nsight Compute provides detailed performance metrics and API debugging via a user interface and command line tool. Nsight Compute also provides customizable and data-driven user interface and metric collection that can be extended with analysis scripts for post-processing results. Process for profiling a CUDA kernel using Nsight Compute is depicted in Figure 6.

Nsight Graphics

This is a standalone development tool for debugging, profiling and analysing graphics applications. Nsight Graphics allows optimization of performance of Direct3D 11, Direct3D 12, DirectX Raytracing 1.1, OpenGL, Vulkan, and KHR Vulkan Ray Tracing Extension based applications.

Example: Performance Roofline of Grete GPUs

Quick Start in the Grete Cluster

Nsight is included with the CUDA development package and to use Nsight in the Grete (GPU) cluster, the preferred CUDA module must be loaded first to set the environment correctly – currently *cuda/12.1* (see Code 7).

```
[u10805@ggpgpu136 ~]$ module load cuda/12.1
```

Code 7

In this example we use a simple CUDA Kernel, *kernel_strided.cu*, with strided memory access (*kernel_c*) to gain insights into some causes of performance bottlenecks in the Grete cluster. We use Nvidia A100 GPUs currently installed in Grete. To achieve this aim we use Nsight Compute (*ncu*) to measure the *arithmetic intensity* and hierarchical *memory bandwidth* to develop a hierarchical roofline model for all memory levels (see Code 8).

Compile the kernel against *nvcc* (see Code 9).

```
1 | module load cuda/12.1
2 | nvcc -o kernel_strided kernel_strided.cu
```

Code 9

Run the batch script – *run_roofline_grete.sh* in Grete (see Code 10).

Results are plotted in Figure 7 which shows the effect of strided memory access in performance. The strided access leads to

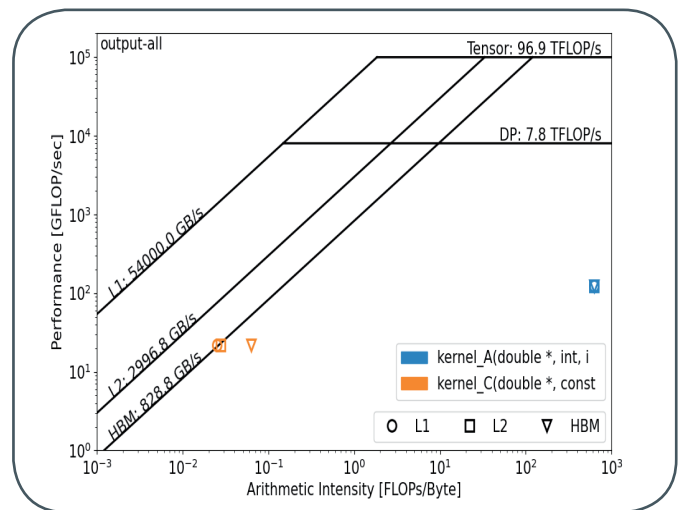


Figure 7: Grete roofline

TOOL	SYSTEM	COMMENTS
LIKWID	Emmy, Grete, SCC	A collection of command-line tools for Performance Optimization, Modeling, and Architecture Analysis. It is a light-weight tool-set for both CPU and GPU systems. Suitable for Node-level performance engineering.
ScoreP	Emmy, Grete, SCC	Instrumentation tool for generating profiles and trace events during execution of parallel programs
Vampir	Emmy, Grete, SCC	A collection of instrumentation, measurement and visualization tools for parallel applications. Uses traces generated by other tools e.g. ScoreP
Nsight	Grete, SCC	A collection of Nvidia development tools targeting Nvidia GPUs

Table 2: Summary of performance tools available in the GWDG HPC systems

poor bandwidth utilization as the hardware fetches chunks of data which are not fully used by the threads.

ADDITIONAL OPTIONS

Considering the ever-growing and sophisticated HPC landscape, the tools presented in this article are not exhaustive, in fact, Scalasca (<https://www.scalasca.org/>), another tool which can be used in both profile and trace modes is available across the GWDG systems and will be presented in another article. Other tools which are OS-native are also available in GWDG systems e.g. Linux Perf. We also perform regular updates on the collection and intergrate new developments for the existing tools.

```
#include <stdio.h>

__global__ void kernel_A(double* A, int N, int M)
{
    double d = 0.0;
    int idx = threadIdx.x + blockIdx.x * blockDim.x;

    if (idx < N) {
#pragma unroll(100)
        for (int j = 0; j < M; ++j) {
            d += A[idx];
        }

        A[idx] = d;
    }
}

__global__ void kernel_C(double* A, const double* B, int N)
{
    int idx = threadIdx.x + blockIdx.x * blockDim.x;

    // Strided memory access: warp 0 accesses (0, stride, 2*stride, ...), warp 1 accesses
    // (1, stride + 1, 2*stride + 1, ...).
    const int stride = 16;
    int strided_idx = threadIdx.x * stride + blockIdx.x % stride + (blockIdx.x / stride) * stride * blockDim.x;

    if (strided_idx < N) {
        A[idx] = B[strided_idx] + B[strided_idx];
    }
}

int main() {

    double* A;
    double* B;

    int N = 80 * 2048 * 100; // 100 * maximum number of resident threads on V100
    size_t sz = N * sizeof(double);

    cudaMalloc((void**) &A, sz);
    cudaMalloc((void**) &B, sz);

    cudaMemset(A, 0, sz);
    cudaMemset(B, 0, sz);

    int threadsPerBlock = 64;
    int numBlocks = (N + threadsPerBlock - 1) / threadsPerBlock;

    int M = 10000;

    kernel_A<<<numBlocks, threadsPerBlock>>>(A, N, M);

    kernel_C<<<numBlocks, threadsPerBlock>>>(A, B, N);

    cudaDeviceSynchronize();
}
```

Code 8

SUMMARY

A number of factors will determine your choice of tools in analysing and optimizing performance of your applications. The presented tools are used in various ways considering; the engineering and performance task, design and functionality of the tools, and the microprocessor architecture in which the tools are applied.

In Table 2, we summarize some worthy-to-consider information about the presented tools considering the before mentioned factors.

REFERENCES

- [1] HPC Performance Documentation: https://docs.gwdg.de/doku.php?id=en:services:application_services:high_performance_computing:performance_engineering_and_analysis
- [2] LIKWID: <https://github.com/RRZE-HPC/likwid/wiki>
- [3] Vampir: <https://vampir.eu/>
- [4] Scalasca: <https://www.scalasca.org/>
- [5] Nsight: <https://developer.nvidia.com/tools-overview>

```
1  #!/bin/bash
2  #SBATCH --job-name=roofline
3  #SBATCH -t 00:30:00
4  #SBATCH -p grete
5  #SBATCH -G A100:1          # requesting GPU slices,
6  #SBATCH --nodes=1         # total number of nodes
7  #SBATCH --ntasks=1       # total number of tasks
8  #SBATCH --cpus-per-task 4 # number of CPU cores per task
9  #SBATCH --mail-type=all   # send mail when job begins and ends
10 #SBATCH --output=./slurm_files/slurm-%x-%j.out
11 #SBATCH --error=./slurm_files/slurm-%x-%j.err
12
13 module load cuda/12.1
14
15 # Time
16 metrics="sm__cycles_elapsed.avg,\
17 sm__cycles_elapsed.avg.per_second,"
18
19 # DP
20 metrics+="sm__sass_thread_inst_executed_op_dadd_pred_on.sum,\
21 sm__sass_thread_inst_executed_op_dfma_pred_on.sum,\
22 sm__sass_thread_inst_executed_op_dmul_pred_on.sum,"
23
24 # SP
25 metrics+="sm__sass_thread_inst_executed_op_fadd_pred_on.sum,\
26 sm__sass_thread_inst_executed_op_ffma_pred_on.sum,\
27 sm__sass_thread_inst_executed_op_fmuls_pred_on.sum,"
28
29 # HP
30 metrics+="sm__sass_thread_inst_executed_op_hadd_pred_on.sum,\
31 sm__sass_thread_inst_executed_op_hfma_pred_on.sum,\
32 sm__sass_thread_inst_executed_op_hmul_pred_on.sum,"
33
34 # Tensor Core
35 metrics+="sm__inst_executed_pipe_tensor.sum,"
36
37 # DRAM, L2 and L1
38 metrics+="dram__bytes.sum,\
39 lts__t_bytes.sum,\
40 lltex__t_bytes.sum"
41
42 dir=./output
43 mkdir -p $dir
44
45 # Baseline
46 output=output.csv
47 profilestr="ncu --metrics $metrics --csv"
48 profilestr_kA="ncu -k kernel_A --metrics $metrics --csv"
49 profilestr_kC="ncu -k kernel_C --metrics $metrics --csv"
50
51 #echo Baseline version
52
53 $profilestr ./kernel_strided > $dir/$output 2>&1
```



GWDG Pad

KOLLABORATION LEICHT GEMACHT!

Ihre Anforderung

Sie möchten allein oder gemeinsam mit Ihrem Team unkompliziert an Textdokumenten arbeiten oder Präsentationen erstellen und dabei auf eine Vielzahl nützlicher Funktionen zurückgreifen. Ihre Änderungen sollen sowohl für Sie als auch Ihre Teamkolleg*innen direkt und in Echtzeit einsehbar sein. Sie möchten die Lese- oder Schreibberechtigung für Ihre Dokumente einschränken können, sodass Sie Ihre Daten vor unbefugtem Zugriff schützen können. Zudem wollen Sie in der Auswahl Ihrer Endgeräte flexibel sein, sowohl mobiler Zugriff als auch Desktop-Varianten sollen unterstützt werden.

Unser Angebot

Auf Basis der freien Software „HedgeDoc“ bieten wir Ihnen einen Dienst, mit dem Sie schnell und unkompliziert Dokumente erstellen, mit anderen Personen teilen und gemeinsam bearbeiten können.

Ihre Vorteile

- > Kollaborativer Echtzeit-Editor
- > Übersicht über alle Ihre Dokumente nach Login

- > Unterstützung von UML-Diagrammen, mathematischen Formeln, Syntax-Highlighting, Musiknoten und vielem mehr
- > Modi zum Erstellen und Vorführen von Präsentationen
- > Einbinden externer Ressourcen wie Videos, PDF-Dateien oder SlideShare
- > Autovervollständigungs-Funktion für Markdown-Ausdrücke
- > Zugriffsbeschränkungen für jedes Dokument einstellbar
- > Veröffentlichung von Dokumenten möglich
- > Webbrowser ausreichend zur Benutzung, keine weitere Installation von Software nötig

Interessiert?

Wenn Sie unseren Dienst „GWDG Pad“ unter <https://pad.gwdg.de> nutzen möchten, benötigen Sie lediglich einen aktuellen Webbrowser. Um eigene Dokumente erstellen zu können, ist zusätzlich die Verwendung eines gültigen GWDG-Accounts oder die einmalige Registrierung unter <https://www.gwdg.de/registration> erforderlich.

Ausbildungsinhalte „Algorithmen und Datenstrukturen“ für Fachinformatiker*innen Anwendungsentwicklung

Text und Kontakt:

Thorsten Hindermann
thorsten.hindermann@gwdg.de
0551 39-30307

In den GWDG-Nachrichten 3/2023 und 4-5/2023 wurde schon einmal über die Ausbildungsinhalte der Fachinformatiker*innen/Anwendungsentwicklung, kurz FI/AE, der GWDG berichtet. In weiteren, in loser Reihenfolge erscheinenden Artikeln soll die Vorstellung von Lehrinhalten dieses Ausbildungsberufs der IHK fortgesetzt werden.

EINLEITUNG

In dem vorliegenden Artikel soll die inhaltliche Ausgestaltung der Lehrinhalte für die Fachinformatiker*innen/Anwendungsentwicklung, kurz FI/AE, nach dem Rahmenausbildungsplan vorgestellt werden. Die hier behandelten Lehrinhalte befassen sich mit Algorithmen und Datenstrukturen. Im Rahmenplan heißt es im Abschnitt „Erstellen, Entwickeln und Betreuen von IT-Lösungen (§ 4 Absatz 2 Nummer 4)“ dazu in Literal d) wie folgt: „Algorithmen formulieren und Anwendungen in einer Programmiersprache erstellen“ und im Abschnitt „Programmieren von Softwarelösungen (§ 4 Absatz 2 Nummer 10)“ dazu in Literal a) wie folgt: „Programmspezifikationen festlegen, Datenmodelle und Strukturen aus fachlichen Anforderungen ableiten sowie Schnittstellen festlegen“. Weiterhin heißt es noch, dass die Fähigkeit zur Auswahl und Anwendung von Programmiersprachen entweder prozedural oder objektorientiert vermittelt werden soll. Auch Werkzeuge zur Versionsverwaltung und Tests sollen möglichst zum Einsatz kommen.

In dem hier beschriebenen Projekt bzw. der Aufgabe werden die im vorherigen Absatz genannten Sachverhalte beleuchtet und erläutert, wie diese Inhalte erstellt und vermittelt werden, damit die Auszubildenden ihre handwerklichen Fertigkeiten und Fähigkeiten in der Fachrichtung Anwendungsentwicklung vermittelt bekommen.

VON DER IDEE ...

Anforderungen

Zusammen mit der Ausbilderin, die für die FI/AE bei der GWDG zuständig ist, wurden in einem Notizenbereich auf dem von der GWDG selbst betriebenen Pad <https://pad.gwdg.de> die Inhalte des Rahmenplans zusammengetragen. Aus den dort gesammelten Stichworten wurden nach gemeinsamer Beratung

die Algorithmen und Datenstrukturen bestimmt, für die die inhaltlichen Vermittlung ausgestaltet werden sollte.

Datenstrukturen

Die ausgewählte Datenstruktur ist eine eindimensionale Anordnung, engl. Array, die Ganzzahlen, in Programmiersprachen meist der Datentyp Integer, aufnimmt und das Erstellen einer einfach verketteten Liste, engl. Linkedlist, ermöglicht. Diese Liste wird von den Auszubildenden in einer vorgegebenen Programmiersprache als selbst erstellter Datentyp *LinkedList* programmiert, wenn nicht auf den vorgefertigten Datentyp *LinkedList* des Rahmenwerks (engl. Framework) zurückgegriffen wird. In der täglichen Praxis wird dies getan werden, aber im vorliegenden Fall sollen sich ja die Auszubildenden mit diesen Lerninhalten auseinandersetzen.

Algorithmen

Für die eindimensionale Ganzzahlanordnung, engl. Integer-Array, soll mit Hilfe der vorgegebenen Sortieralgorithmen Bubble- und Quicksort die gegebene Anordnung des Integer-Array sortiert werden. Die beiden Sortieralgorithmen werden für die weitere

Training Content „Algorithms and Data Structures“ for IT Specialists in Application Development

The GWDG News 3/2023 and 4-5/2023 have already reported on the training content for IT specialists/application developers, FI/AE for short, at the GWDG. The presentation of the training content of this IHK training profession will be continued in further articles that will appear in irregular order.

Verwendung als Programmierschnittstellen, engl. **Application Programmers Interfaces**, kurz **APIs**, zur Verfügung gestellt. Diese können sowohl vom bzw. von der Entwickler*in selbst als auch von anderen Entwickler*innen genutzt werden, so dass das „Rad“ nicht zweimal erfunden werden muss.

Für die **Linkedlist** sollen die **APIs** für die folgenden Fälle Erstellen (engl. **Create**), Lesen (engl. **Read**), Ändern (engl. **Update**) und Löschen (engl. **Delete**), kurz **CRUD**, erstellt werden. Weiterhin sollen zwei Suchalgorithmen erstellt werden, und zwar eine lineare Suche, für kleinere Datenmengen ausreichend, und eine binäre Suche, bei größeren Datenmengen schneller als die lineare Suche, die über die Inhalte der **Linkedlist** suchen und ermitteln, ob gegebene Inhalte in dieser Liste enthalten sind.

Zeitkomplexität

Im Bereich der zu erstellenden Algorithmen wird auch bei den einzelnen Verfahren untersucht, wie viel Zeit es braucht, wenn bei der **Linkedlist** die Datenmengen steigen oder aber beim Hinzufügen eines weiteren Datums bei größer werdender Datenmenge in der **Linkedlist**.

Hier soll vermittelt werden, sich über diese größer werdenden Zeiten Gedanken zu machen und wie man den steigenden Zeitverbrauch durch geschickte Auswahl der Algorithmen optimieren, also verringern kann.

Zyklomatische Komplexität

Unter diesem Stichwort soll untersucht und vermittelt werden, wie kompliziert womöglich die selbst erstellte **API** geworden ist. Wenn diese mit zu vielen Abfragen und durch deren Verzweigungen zu viele Pfade in den Quellzeilen aufweist, die das Programm nehmen könnte, und somit zu kompliziert wird, soll dies von den Auszubildenden erkannt werden und durch Umstrukturierung der Quellzeilen, im Englischen **Refactoring** genannt, gegengesteuert werden. Somit wird auch wieder die Lesbarkeit und Übersichtlichkeit der Quellzeilen verbessert.

Generischer Ansatz von Datentypen

Hier soll von den Auszubildenden erkannt und erarbeitet werden, ob es sinnvoll ist, für mögliche Datentypen, wie z. B. *Integer* oder *Double*, eine auf den zu verwendenden Datentyp spezialisierte **Linkedlist** zu erstellen, oder ob es Möglichkeiten eines generischen Ansatzes gibt, der mit möglichst vielen Datentypen zurechtkommt und somit eine generische **Linkedlist** erstellt, die mit möglichst vielen Datentypen zurechtkommt und deswegen nicht verändert oder angepasst werden muss.

... ÜBER DIE WERKZEUGE ...

Das Hauptwerkzeug in der täglichen Programmierarbeit ist die **Integrierte Entwicklungsumgebung**, engl. **Integrated Development Environment**, kurz **IDE**. Dieses Tool erleichtert die Erstellung der zu entwickelnden Programme, Bibliotheken und deren **APIs** mit Hilfe der verwendeten Programmiersprache und des genutzten **Rahmenwerks**, engl. **Framework**.

Sprache und Rahmenwerk

Bei der **GWDG** ist die hauptsächliche Programmiersprache **Java** mit dem **JDK** (**Java Development Kit**) als **Framework**. Im **IHK-Rahmenplan** ist auch vorgesehen, dass nicht nur eine

Compiler-Sprache erlernt werden soll, sondern auch zusätzlich eine zweite. Diese zweite Sprache ist für **FI/AE-Auszubildende** bei der **GWDG C#** mit dem **.Net Framework**. Beide Sprachen und **Rahmenwerke** ähneln sich in großen Teilen, so dass es für die **FI/AE** recht gut gelingt, mit beiden Sprachen zu entwickeln. Auch mit **C#** ist es seit dem Erscheinen des **.Net Core Framework** möglich, neben **Windows** auch für die Betriebssysteme **Linux** und **macOS** damit Anwendungen zu entwickeln.

Da bei den Auszubildenden und Ausbilder*innen alle drei Betriebssysteme auf deren Entwicklungsrechnern im Einsatz sind, ist mit der Auswahl dieser Sprachen und **Rahmenwerke** keine Einschränkung verbunden und das Erlernen der Entwicklung von Anwendungen plattformunabhängig geworden.

Integrierte Entwicklungsumgebung

Die im vorherigen Absatz beschriebenen Bedingungen in Bezug auf die plattformübergreifende Verfügbarkeit der Programmiersprachen und **Rahmenwerke** schränkt die zur Verfügung stehenden und einsetzbaren **Integrierten Entwicklungsumgebungen** etwas ein. Die **IDEs** der Firma **JetBrains**, die bei der **GWDG** bei den **FI/AE** zum Einsatz kommen, gibt es für die drei genannten Betriebssysteme **Windows**, **Linux** und **macOS**. Somit kann von allen das gleiche Programm erstellt werden, egal welches Betriebssystem verwendet wird. Das gibt den Auszubildenden die Möglichkeit und Freiheit, mit ihrem bevorzugten Betriebssystem zu arbeiten, schränkt aber nicht die zu erlernenden Inhalte ein.

Funktionale Tests

Mit **funktionalen Tests**, engl. **Units-Tests**, soll getestet werden, ob die programmierten Methoden, die zuvor ja schon als **API** benannt und vorgestellt wurden, genau das ausführen, was die **Anwendungsentwickler*innen** erreichen und an **Funktionalität** zur Verfügung stellen möchten. Dies ist bekannt unter **testgetriebener Entwicklung**, engl. **Test Driven Development**, kurz **TDD**.

Dabei wird wie folgt vorgegangen. Zuerst werden die **Tests** zur Überprüfung der Methoden entwickelt. Diese schlagen fehl, weil die zu überprüfenden Methoden ja noch gar nicht fertig entwickelt sind. In den **IDEs** werden die fehlgeschlagenen **Tests** in der Farbe **Rot** angezeigt. Wenn dann nach und nach die Methoden fertig entwickelt werden und die Überprüfung durch die **Tests** bestehen, werden diese bei Erfolg in der Farbe **Grün** angezeigt. Das ist die sogenannte **Rot-nach-Grün-Entwicklung**, engl. **Red-To-Green-Development**, in der **TDD**.

Ein einzelner **Test** ist dabei so aufgebaut, dass dieser dem **Muster Einrichten, Ausführen, Behaupten bzw. Überprüfen**, engl. **Arrange, Act und Assert**, kurz **AAA**, folgt.

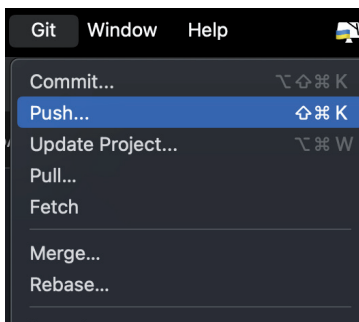
Welches der vielen existierenden **Testrahmenwerke** für das Projekt ausgewählt wird, ist nicht ganz so entscheidend. Alle **Testrahmenwerke** sind sehr ähnlich aufgebaut und sind somit für die **Testzwecke** geeignet. Allein die eingesetzte **IDE** entscheidet, welche **Testrahmenwerke** zum Einsatz kommen, weil die **IDE** diese entsprechend integrieren und unterstützen muss.

Versionskontrolle

Die **Versionierung** der Quellzeilen der verschiedenen Projekte im Themenblock „Algorithmen und Datenstrukturen“ soll mit der **Versionverwaltung Git** erfolgen. Hierbei wird das Augenmerk darauf gelegt, dass die **Versionskontrolle** mit den eingebauten Möglichkeiten der **IDE** erfolgt. Somit muss diese für diesen

Arbeitsschritt nicht verlassen werden, aber durch die nahtlose Integration in die IDE ist dies auch nicht notwendig.

Dies umfasst das Erstellen von Übergaben an das lokale Git-Repository, *git commit*, sowie dessen sinnvolle Kommentierung, bestehend aus einer zusammenfassenden Überschrift und einem erklärenden Text. Weiterhin *git push*, um das beim Erstellen der Projekte lokale Git-Repository, engl. Repository, kurz Repo, auf der Festplatte des Entwicklungsrechners in das entfernte Git-Repository auf dem selbstbetriebenen GitLab-Server der GWDG <https://gitlab.gwdg.de> hochzuladen. Damit haben andere Entwickler*innen ebenfalls Zugriff auf dieses Repository, wo nun gemeinsam daran gearbeitet werden kann. Entweder für die Erweiterung der Quellzeilen für neue Funktionen oder um anzuschauen, was der/die Entwickler*in entwickelt hat, um für



1_Die wichtigsten und gebräuchlichsten Git-Befehle, die direkt über diese Menüpunkte ausgeführt werden können, ohne dafür die IDE verlassen zu müssen

diese Quellzeilen Rückmeldung und Verbesserungsvorschläge zu geben und gemeinsam zu besprechen.

Daneben wird aber auch eine Vielzahl von weiteren wichtigen Git-Kommandos in die IDE integriert. Welche das im Einzelnen sind, hängt vom Hersteller der IDE ab und unterscheidet sich jeweils in kleinen Details, die zu zahlreich wären, um diese hier abschließend aufzuzählen.

Für die JetBrains IDE Rider sind hier die wichtigsten Git-Befehle, die Rider unterstützt, in einem Bildschirmausschnitt in Abbildung 1 gezeigt.

Somit wird hiermit demonstriert, dass eine gute IDE für alle Aufgaben rund um die Anwendungsentwicklung genutzt werden kann. Einmal beim Arbeitsbeginn gestartet, kann mit ihr der komplette Entwicklungszyklus an einem Arbeitstag bewerkstelligt werden, ohne sie zwischendurch zu verlassen.

... ZU QUELLZEILEN

In diesem Abschnitt werden auszugsweise Quellzeilenbeispiele oder Bildschirmausschnitte anreißern, was mit den anschließenden Unterabschnittsüberschriften gemeint ist.

Die gesamte Quellzeilenbasis dieses Projekts sowie dessen Teilprojekte befinden sich im internen GitLab-Ausbildungsrepository der GWDG. Auf Anfrage anderer Ausbilder*innen der Institute der Max-Planck-Gesellschaft oder der Universität Göttingen können diese nach Rücksprache mit der Gruppen- oder Geschäftsleitung gerne als Kopie zur Verfügung gestellt werden.

Sortieren

Mit den Sortieralgorithmen wird sich bei diesem gesamten Aufgabenkomplex zuerst beschäftigt. Hier werden die beiden Algorithmen Bubblesort und Quicksort näher betrachtet. Auf Implementierung und Arbeitsweise wird hierbei ein großes Augenmerk gelegt, so dass die Auszubildenden die Vor- und Nachteile der beiden Algorithmen selbst erarbeiten, natürlich von den Ausbilder*innen begleitet, die für Anleitung, Fragen und Antworten während der gemeinsamen Programmierübungen zur

Verfügung stehen.

Quellzeilenauflistung 1 zeigt die Kernarbeitsmethode des Bubblesort-Algorithmus.

```
private void BubbleSort(int[] anArray, SortingOrder sortingOrder)
{
    for (var outerLoop = anArray.Length; outerLoop > 1; outerLoop--)
        for (var innerLoop = 0; innerLoop < anArray.Length - 1; innerLoop++)
            switch (sortingOrder)
            {
                case SortingOrder.Ascending:
                    if (anArray[innerLoop] > anArray[innerLoop + 1])
                        SwapElements(anArray, innerLoop);
                    break;
                case SortingOrder.Descending:
                    if (anArray[innerLoop] < anArray[innerLoop + 1])
                        SwapElements(anArray, innerLoop);
                    break;
            }
}
```

Quellzeilenauflistung 1: Kernarbeitsfunktion des Bubblesort-Algorithmus

Die aufgerufene Methode *SwapElements* ist in der Quellzeilenauflistung 2 zu sehen.

```
private void SwapElements(int[] anArray, int index)
{
    int temp = anArray[index];
    anArray[index] = anArray[index + 1];
    anArray[index + 1] = temp;
}
```

Quellzeilenauflistung 2: Tausch-Funktion, die während des Sortiervorgangs im Bubblesort-Algorithmus aufgerufen wird

Verkettete Liste

Der zweite große Bereich dieses in diesem Artikel beschriebenen Ausbildungsbereichs sind Datenstrukturen. Datenstrukturen werden genutzt, um Daten in strukturierter Art und Weise abzuspeichern und wieder abrufen zu können.

In diesem Bereich erarbeiten sich die Auszubildenden eine verkettete Liste. Auch hier wird das Augenmerk auf die Implementierung und Arbeitsweise gelegt und, um genau wie bei den Sortieralgorithmen auch hier die Vor- und Nachteile im Vergleich zu anderen Datenstrukturen/-typen (wie letztere in Programmiersprachen genannt werden), wie z. B. *Array* oder *List*, zu erarbeiten.

Quellzeilenauflistung 3 zeigt die Knotendatenstruktur, engl. Node, der verketteten Liste, die für die Speicherung des Datums

```
/// <summary>
/// Represents a node in a generic linked list.
/// </summary>
/// <typeparam name="T">The type of the node.</typeparam>
public sealed class GenericLinkedListNode<T> : IDisposable
{
    /// <summary>
    /// Represents the value of a node in a generic linked list. The
    /// value can be of any type.
    /// </summary>
    /// <typeparam name="T">The type of the value.</typeparam>
    public T Value { get; private set; }

    /// <summary>
    /// Represents the next node in a generic linked list.
    /// </summary>
    /// <typeparam name="T">The type of the value of the next
    /// node.</typeparam>
    public GenericLinkedListNode<T>? Next { get; set; }

    /// <summary>
    /// Represents a node in the generic linked list.
    /// </summary>
    /// <typeparam name="T">The type of the node.</typeparam>
    public GenericLinkedListNode(T value)
    {
        Value = value;
        Next = null;
    }
}
```

Quellzeilenauflistung 3: Knotendatenstruktur der verketteten Liste

oder der Daten zuständig ist, je nach Art und Komplexität der zu speichernden Daten. Weiterhin enthält der Knoten noch einen Speicherplatz auf den nächsten Knoten. So bilden zwei oder mehr Knoten dann eine Kette, die zur Namensgebung dieser Datenstruktur beigetragen hat.

Quellzeilenauflistung 4 zeigt die Hinzufügen-Funktion, die für das Hinzufügen eines neuen Knotens in die verkettete Liste zuständig ist. Darin wird das Datum, repräsentiert durch die Variable *value*, gespeichert sowie der neue Knoten an dem nächsten freien (*current.Next = newNode*) und letzten (*Last = newNode*) Knotenplatz gespeichert.

```

/// <summary>
/// Adds a new node with the specified value to the end of the
/// linked list.
/// This operation has a time complexity of O(n).
/// </summary>
/// <typeparam name="T">The type of the value of the
/// node.</typeparam>
/// <param name="value">The value of the node to be added.</param>
public void AddNode(T value)
{
    var newNode = new GenericLinkedListNode<T>(value);

    if (First == null)
    {
        First = newNode;
        Last = newNode;
    }
    else
    {
        var current = First;
        while (current.Next != null)
        {
            current = current.Next;
        }
        current.Next = newNode;
        Last = newNode;
    }
    Count++;
}

```

Quellzeilenauflistung 4: Hinzufügen-Funktion der verketteten Liste

Suchen

Beim Suchen von Daten in Datenstrukturen, in diesem Fall der verketteten Liste, werden zwei Suchmethoden von den Auszubildenden implementiert und deren Arbeitsweise näher betrachtet, damit sie entscheiden können, welche Methode sich am besten für wenige oder viele Daten eignet.

```

/// <summary>
/// Searches for an item in the generic linked list based on the
/// specified value and search method.
/// </summary>
/// <param name="value">The value to search for.</param>
/// <param name="searchMethod">The search method to use. The default
/// value is SearchMethod.Linear.</param>
/// <returns>True if the item is found in the linked list;
/// otherwise, false.</returns>
public bool FindItem(T value, SearchMethod searchMethod =
SearchMethod.Linear)
{
    var newNode = new GenericLinkedListNode<T>(value);

    if (searchMethod == SearchMethod.Linear)
    {
        var currentNode = First;
        while (currentNode != null)
        {
            if (currentNode.Value.Equals(newNode.Value))
                return true;
            currentNode = currentNode.Next;
        }
    }
    else if (searchMethod == SearchMethod.Binary){...}

    return false;
}

```

Quellzeilenauflistung 5: Einfache iterative Suche über die Knoten der verketteten Liste. Der *else if*-Zweig wird nicht betrachtet.

Quellzeilenauflistung 5 zeigt eine einfache Vorgehensweise bei der Suche von Daten, wobei bei dieser Vorgehensweise iterativ über alle Knoten der verketteten Liste gegangen wird, um das Datum zu finden, nach dem gesucht wird, repräsentiert durch die Variable *value*. Enthält die Liste das gesuchte Datum, wird Wahr (*return true*) aus der Funktion zurückgegeben, sonst Falsch (*return false*).

Zeitkomplexität

Die Zeitkomplexität in der Programmierung ist ein Maß dafür, wie sich die Laufzeit eines Algorithmus oder einer Funktion in Abhängigkeit von der Eingabegröße verändert. Die Laufzeit wird oft als eine Funktion ausgedrückt, die die obere Grenze für die Anzahl der Operationen angibt, die der Algorithmus oder die Funktion ausführen muss.

Beim Hinzufügen von Knoten in eine verkettete Liste hängt die Zeitkomplexität davon ab, ob die Liste eine kleine oder schon große Menge verketteter Knoten enthält und ob ein bestimmter Index oder Wert gesucht wird. Wenn die Liste unsortiert ist und kein bestimmter Index oder Wert angegeben wird, kann ein neuer Knoten am Anfang oder am Ende der Liste in konstanter Zeit $O(1)$ hinzugefügt werden, da keine Suche erforderlich ist. Wenn die Liste sortiert ist oder ein bestimmter Index oder Wert angegeben wird, muss die Liste durchlaufen werden, um die richtige Position für den neuen Knoten zu finden. Dies erfordert eine lineare Zeit $O(n)$, wobei n die Länge der Liste ist. Die Zeitkomplexität für das Hinzufügen von Knoten in eine verkettete Liste kann also zwischen $O(1)$ und $O(n)$ variieren, je nachdem, wie die Liste organisiert ist und was die Eingabeparameter sind.

Im Laufe der Zeit haben sich aber Wege gefunden, weitere Knoten in eine Liste in konstanter Zeit, also mit $O(1)$, einzufügen. Quellzeilenauflistung 6 zeigt nochmals die *AddNode*-Funktion, die wir schon in der Quellzeilenauflistung 4 vorgestellt haben. In dieser Auflistung werden weitere Knoten mit der Zeitkomplexität von $O(n)$ eingefügt, da hier iterativ über alle Knoten entlang gegangen wird, um den freien Speicherplatz für den neuen eingefügten Knoten zu finden. Die Bedingung dafür ist, dass *current.Next = null* ist. Wenn das nicht der Fall ist, wird weiterhin die *while*-Schleife durchlaufen und der nächste Knoten geladen. Das wird mit der Bedingung in der *while*-Schleife *current.Next != null* überprüft.

```

/// <summary>
/// Adds a new node with the specified value to the end of the
/// linked list.
/// This operation has a time complexity of O(1).
/// </summary>
/// <typeparam name="T">The type of the node.</typeparam>
/// <param name="value">The value of the node to be added.</param>
public void AddNode(T value)
{
    var newNode = new GenericLinkedListNode<T>(value);

    if (First == null)
    {
        First = newNode;
        Last = newNode;
    }
    else
    {
        Last!.Next = newNode;
        Last = newNode;
    }
    Count++;
}

```

Quellzeilenauflistung 6: Hinzufügen von Knoten zur verketteten Liste mit der konstanten Zeitkomplexität $O(1)$

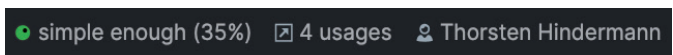
Zyklomatische Komplexität

Die zyklomatische Komplexität ist eine Metrik, die die Anzahl der linearen unabhängigen Pfade durch die Quellzeilen einer Funktion angibt. Sie wird ermittelt, indem die Anzahl der Verzweigungs- und Schleifenanweisungen (if, else, switch, for, while, do etc.) in den Quellzeilen gezählt wird und 1 hinzuaddiert wird. Eine höhere zyklomatische Komplexität weist auf eine höhere Komplexität und ein höheres Risiko für Fehler und Wartungsprobleme im Code hin.

Dieser Vorgang kann in der IDE automatisch ermittelt werden, entweder durch Hinzufügen eines zusätzlichen Softwaresteck-Moduls, engl. Plugin, oder die IDE verfügt schon eingebaut über diese Möglichkeit, die zyklomatische Komplexität anzuzeigen.

In JetBrains Rider wird diese Funktion durch ein Plugin bereitgestellt (siehe Beschreibung und Installationsanleitungen hierzu unter <https://plugins.jetbrains.com/plugin/10395-cyclomaticcomplexity>).

Wenn in der IDE installiert, wird diese Komplexität im Quellzeilenbearbeitungsbereich dargestellt (siehe Abbildung 2).



2_Grafische Darstellung der zyklomatischen Komplexität durch das hinzugefügte Plugin

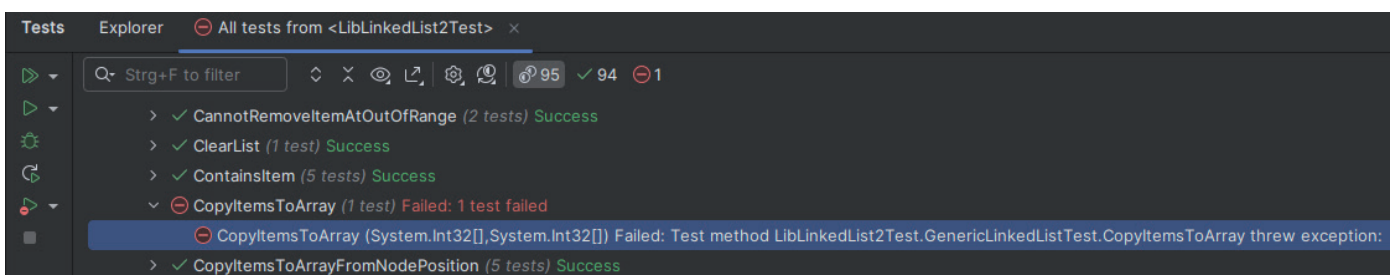
Funktionale Tests

Mit den funktionalen Tests wird überprüft und soll sichergestellt werden, dass die Implementierung der Funktionen die Testbedingungen erfüllt. Dazu wird der Test vorbereitet (Arrange), auch mit den Testdaten für den Aufruf der zu testenden Funktion. Anschließend wird der Test der Funktion ausgeführt (Act). Abschließend wird überprüft, ob die implementierte Funktion das richtige Ergebnis liefert (Assert). Wenn das der Fall ist, wird der Test grün angezeigt. Wenn die Implementierung noch nicht korrekt ist, wird der Test rot angezeigt.

Man kann kaum glauben, dass es so „einfach“ ist, ist es aber. Der Mehrwert der funktionalen Tests liegt darin, dass bei der Verbesserung und/oder Umstellung (Refaktorisierung) der Quellzeilen der gerade bearbeiteten Funktion mit den Tests sofort überprüft werden kann, ob die Logik beschädigt worden ist und der Test nach der Veränderung auf den Zustand rot geht. Wenn alles gut gelaufen ist, bleibt der Test weiterhin grün.

In Quellzeilenauflistung 7 ist auch schön zu sehen, dass die Testfunktion für das Testsystem markiert wird, und zwar mit der Eigenschaft [TestMethod]. In der Eigenschaft [DataRow(...)] werden dem Testsystem die Testdaten für die Eingabe und das erwartete Ergebnis übergeben. Weiterhin sind mit den drei Kommentaren die schon zuvor erwähnten drei Schritte gekennzeichnet: Arrange, Act und Assert.

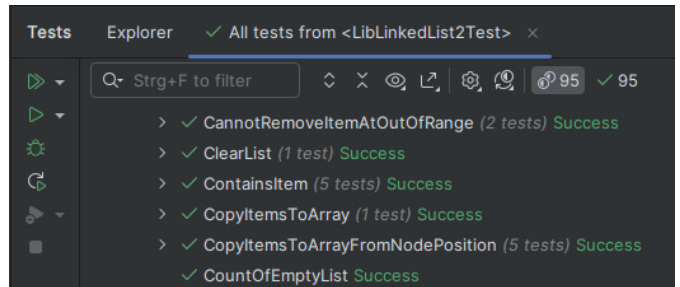
Abbildung 3 zeigt den fehlgeschlagenen Test der Funktion CopyTo. Dieser Status in rot zeigt Entwickler*innen an, dass die Funktion noch nicht richtig implementiert ist und weiterer



3_Zeigt den fehlgeschlagenen Test der Funktion CopyTo. Dieser wird deshalb in der Testübersicht rot angezeigt.

```
[TestMethod]
[DataRow(new int[] { 55, 07, 78, 12, 42 }, new int[] { 55, 07, 78, 12, 42 })]
public void CopyItemsToArray(int[] values, int[] expected)
{
    int[] tempArray = new int[values.Length];
    // Arrange
    foreach (var value in values)
        _genericIntList.AddNode(value);
    // Act
    _genericIntList.CopyTo(tempArray);
    // Assert
    for (int index = 0; index < _genericIntList.Count; index++)
        Assert.AreEqual(expected[index], tempArray[index],
            $"The value at position {index} is incorrect after sorting");
}
```

Quellzeilenauflistung 7: Test der Funktion CopyTo



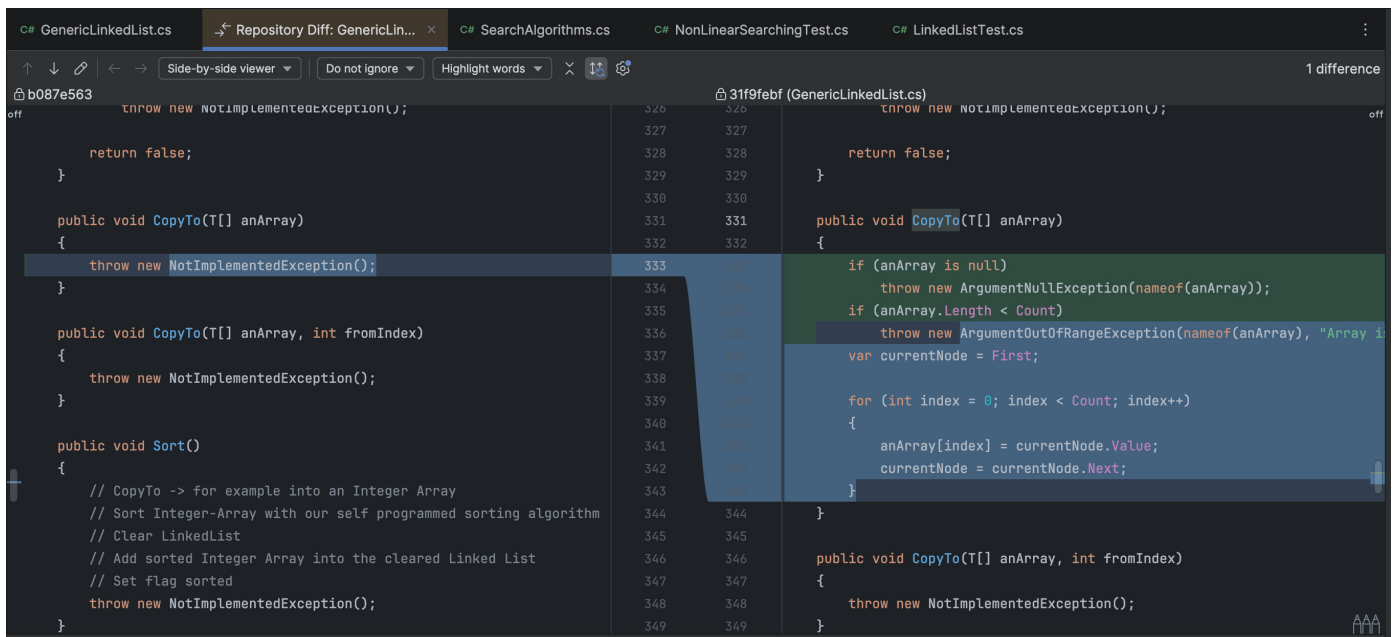
4_Erfolgreicher Test der Funktion CopyTo. Nachdem Entwickler*innen die Implementierung erstmalig implementiert und/oder weiter verbessert haben, wird diese nun in der Testübersicht endlich grün angezeigt.

Überarbeitung bedarf. Und zwar so lange überarbeiten bzw. weiter implementieren, bis der Test dann grün wird, wie in Abbildung 4 zu sehen. Dies zeigt Entwickler*innen an, dass die Funktion nun allen Anforderungen entspricht, die mit den Test abgedeckt worden sind

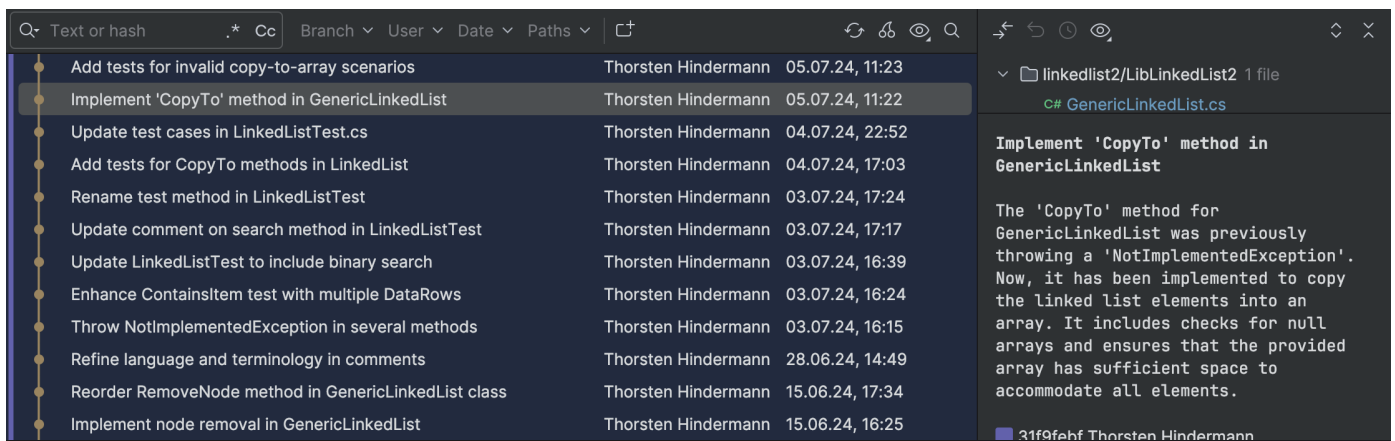
Git-Integration

Neben den schon weiter oben im Abschnitt „Versionskontrolle“ beschriebenen Aspekten ist ein weiterer, wichtiger und praxisrelevanter Punkt, dass Entwickler*innen ihre tägliche Arbeit von ihrem lokalen Entwicklungsrechner auf einen entfernten Server hochladen. Wenn das eigene Arbeitsgerät kaputt geht, sind Entwicklungsprogramm und Quellzeilen getrennt. Die IDE ist auf dem neuen Ersatzrechner schnell installiert und mit einem Auschecken der Quellzeilen aus dem entfernten Repositoryum auf den lokalen Entwicklungsrechner sind Entwickler*innen somit schnell wieder in der Lage, dort weiterzuentwickeln, wo sie mit ihrem letzten git commit und git push aufgehört haben.

Das Thema Git ist zu komplex, um es im Rahmen dieses Artikels abzuhandeln. Im Bildschirmausschnitt in Abbildung 5 ist exemplarisch eine Differenzansicht dargestellt, wie in diesem Arbeitsschritt die Funktion CopyTo implementiert worden ist. Diese Funktion ist erst eingchecked worden, nachdem die funktionalen Tests für diese Funktion von Rot nach Grün gewechselt sind.



5_Differenzansicht der erfolgreich implementierten Funktion, nachdem alle Tests grün geworden sind und somit *git commit* und *git push* aus der IDE ausgeführt werden konnten



6_Ansicht der zahlreich übergebenen Arbeitsschritte in das Repository (links zu sehen und rechts zu der jeweiligen Übergabe ein erklärender Text)

Im Bildschirmausschnitt in Abbildung 6 sind ausschnittsweise die zahlreichen, mittels *git commit* an das lokale und mittels *git push* an das entfernte Repository, übergebenen Arbeitsschritte für Entwickler*innen dokumentiert. Zu jeder Übergabe gibt es jeweils auch eine zusammenfassende Überschrift und einen erklärenden Text des übergebenen Arbeitsschrittes.

MÖGLICHE AUFGABENSTELLUNGEN

Nach dem begleiteten Teilprojekt „Sortieren von eindimensionalen Ganzzahlanordnungen“, engl. Integer Arrays, mit Hilfe der Bubblesort- und Quicksort-Algorithmen und dem Hauptprojekt „Verkettete Liste“, engl. Linkedlist, sind die Auszubildenden in der Lage, das Teilprojekt für den binären Suchalgorithmus, engl. Binary Search Algorithm, nach diesen vorangegangenen begleiteten Vorbildprojekten selbst zu erstellen. Dabei sollen sie das Projekt selbst und die Klasse mit der Methode, zuvor schon als API vorgestellt, für den binären Suchalgorithmus erstellen.

Weitere Aufgabenstellungen bzw. Teilprojekte sind:

- Die gleichzeitige Anlage des Teilprojekts „Funktionale Tests“, Unit-Tests, sowie die Erstellung der möglichen Tests der zu untersuchenden Testszenerien für den

binären Suchalgorithmus.

- Das Hochladen des bei dem Anlegen der Projekte erstellten lokalen Git-Repositories in das entfernte Git-Repository mit *git-push* auf einen git-fähigen Repositoriums-Server, wenn hier bei der Entwicklung entsprechende *git-commits* mit aussagekräftigen Beschreibungen erstellt wurden.
- Die Einbindung der beiden Teilprojekte, wie zuvor auch schon die beiden Teilprojekte „Sortieren und zugehörige Unit-Tests“ sowie „Binärer Suchalgorithmus und dessen korrespondierende Units-Tests“, in das Hauptprojekt „Linkedlist“.
- Ausreichende Dokumentation der Methoden bzw. APIs der Klasse binärer Suchalgorithmus in den jeweiligen Quellzeilendateien und eine ausführliche Gesamtdokumentation in der *README.md*-Datei, die jedem Projekt hinzugefügt werden sollte, im entsprechenden Repository.

UNTERSTÜTZUNG DER AUSBILDUNG

Noch ein Wort zum Autor dieses Artikels. Dieser unterstützt seit 2011 die Ausbildung der Fachinformatiker*innen Anwendungsentwicklung. Da er keine IHK-Ausbilderprüfung (umgangssprachlich „Ausbilderschein“) hat, beruht seine Unterstützung auf Basis des in § 28 (Eignung von Ausbildenden und Ausbildern oder Ausbilderinnen) Abs. 3 Berufsbildungsgesetzes, kurz BBiG, Niedergeschriebenen (siehe https://www.gesetze-im-internet.de/bbig_2005/_28.html):

(3) Unter der Verantwortung des Ausbilders oder der Ausbilderin kann bei der Berufsausbildung mitwirken, wer selbst nicht Ausbilder oder Ausbilderin ist, aber abweichend von den besonderen Voraussetzungen des § 30 die für die Vermittlung von Ausbildungsinhalten erforderlichen beruflichen Fertigkeiten, Kenntnisse und Fähigkeiten besitzt und persönlich geeignet ist.

Aufgrund der über mehrere Jahrzehnte erworbenen beruflichen Fertigkeiten, Kenntnisse und Fähigkeiten u. a. in der Programmierung und Anwendungsentwicklung sowie der persönlichen Eignung erfüllt er diese Voraussetzungen und unterstützt die

Fachinformatiker*innen-Ausbildung seit ihrem Start 2011.

Daher ist es nicht zwingend notwendig, dass jede*r, der/die mit der Ausbildung zu tun hat, zwangsweise eine IHK-Ausbilderprüfung ablegen muss. Somit kann in einem Institut die Ausbildung, egal ob Fachinformatiker*innen oder andere Ausbildungsgänge, sehr breit gefächert, d. h. auf viele ausbildungsunterstützende Mitarbeiter*innen basierend, nach erwähntem § 28 Abs. 3 aufgeteilt werden, um die Auszubildenden bestmöglich und breit gefächert auf ihre Prüfungen und ihr späteres Berufsleben vorzubereiten.

AUSBLICK

Auch in kommenden Ausgaben der GWDG-Nachrichten soll in loser Folge immer mal wieder von interessanten Ausbildungsinhalten der Fachinformatiker*innen/Anwendungsentwicklung berichtet werden. In einem weiteren Artikel in einer der nächsten Ausgaben wird das Thema „Entwurfsmuster“ behandelt und wie dieser Lehrinhalt den FI/AE-Auszubildenden vermittelt wird. ■

Kurz & knapp

KI-Methoden und -Support für Wissenschaft und Industrie

Konferenz der deutschen KI-Servicezentren im September in Göttingen

(Presseinformation Nr. 118 der Universität Göttingen vom 15.08.2024)

(pug) Die Entwicklung im Bereich der Künstlichen Intelligenz verläuft derzeit mit hoher Dynamik. Für Unternehmen stellt der Einstieg in die Nutzung von KI für unternehmensspezifische Zwecke oftmals eine erhebliche Herausforderung dar. Seit 2022 fördert das Bundesministerium für Bildung und Forschung vier nationale KI-Servicezentren, die hierzu Beratung und Infrastruktur anbieten. Sie stärken die Zusammenarbeit zwischen Wissenschaft und Wirtschaft und versetzen Forschungseinrichtungen ebenso wie Unternehmen in die Lage, passende KI-Anwendungen zu nutzen. Die erste gemeinsame Konferenz der KI-Servicezentren findet am 18. und 19. September 2024 an der Universität Göttingen statt. Sie richtet sich an Forschende und Interessierte aus Wissenschaft und Industrie, insbesondere aus kleinen und mittleren Unternehmen mit Interesse und Bedarf an Rechenleistung und KI-Expertise.

Am ersten Tag geben Keynote-Vorträge Einblick in die Themenbereiche „KI und Gesellschaft“, „KI und Energie“ und „KI und Medizin“. Panel-Diskussionen mit Vertreterinnen und Vertretern aus Politik, Wirtschaft und Wissenschaft, wissenschaftliche Sessions, Hands-on-Workshops sowie Führungen durch das Göttinger Rechenzentrum runden das abwechslungsreiche Programm ab.

Das Vernetzungsevent am zweiten Tag lädt zu einem Austausch mit Multiplikatoren, Start-ups und den KI-Servicezentren ein. Nach einem Panel zum Thema „AI made in Germany –

Wie beschleunigen KI-Servicezentren den Transfer von KI-Anwendungen in Unternehmen?“ besteht die Möglichkeit, sich mit ausgewählten Organisationen und Unternehmen im Bereich KI zu vernetzen. Ebenso ist der direkte Kontakt mit den verschiedenen KI-Servicezentren gegeben, um mehr über deren Forschung, Bildungsangebote und vorhandene Ressourcen zu erfahren. Alle interessierten Unternehmen sind herzlich eingeladen, hieran teilzunehmen.

Gastgeber der KonKIS 2024 ist das KI-Servicezentrum für sensible und kritische Infrastrukturen KISSKI, das von elf Forschungseinrichtungen an den Standorten Göttingen, Hannover und Kassel getragen wird. Die Konferenz findet im Tagungs- und Veranstaltungshaus Alte Mensa, Wilhelmsplatz 3, statt. Die Anmeldung zu Konferenz und Vernetzungsevent ist noch bis zum 4. September 2024 geöffnet. Eine Anmeldung nur für das Vernetzungsevent ist möglich. Weitere Informationen und das genaue Tagungsprogramm sind unter <https://konkis.kisski.de> zu finden.

Hinweis an die Redaktionen:

Journalistinnen und Journalisten sind zur Veranstaltung herzlich eingeladen.

Kontakt:

Prof. Dr. Julian Kunkel
Georg-August-Universität Göttingen
Fakultät für Mathematik und Informatik
Institut für Informatik
Telefon (0551) 39-30144
E-Mail: julian.kunkel@gwdg.de
Internet: <https://www.uni-goettingen.de/de/619486.html>

Stellenangebot

Nr. 20240819

Die GWDG sucht zum nächstmöglichen Zeitpunkt zur Verstärkung der Arbeitsgruppe „eScience“ (AG E) zwei

Softwareentwickler*innen / Systemadministrator*innen (m/w/d)

mit einer regelmäßigen Wochenarbeitszeit von 39 Stunden. Die Vergütung erfolgt nach dem Tarifvertrag für den öffentlichen Dienst (Bund); die Eingruppierung ist je nach Qualifikation bis zur Entgeltgruppe TVöD E 13 vorgesehen. Die Stellen sind grundsätzlich auch für Teilzeitkräfte geeignet und zunächst auf 24 Monate befristet. Allerdings strebt die GWDG eine langfristige Zusammenarbeit an. Bei Interesse besteht die Möglichkeit zur Promotion.

Themengebiet

Im Rahmen verschiedener Forschungsprojekte entwickelt die GWDG gemeinsam mit internationalen Partnern aus der Wissenschaft IT-Dienstleistungen im Kontext von Persistenten Identifikatoren (PID). Diese dienen als Referenzen z. B. zu (Forschungs-)Daten oder Software und haben in den letzten Jahren immer mehr an Bedeutung im internationalen Wissensaustausch gewonnen und werden z. B. umfangreich in der Nationalen Forschungsdateninfrastruktur (NFDI) eingesetzt, an der die GWDG in diesem Kontext mitarbeitet. Zudem betreibt die GWDG seit Jahren PID-Dienste für eine Vielzahl von Kundengruppen aus der Forschung gemeinsam mit einem internationalen Konsortium.

Aufgabenbereiche

Sie werden in der eScience-Gruppe der GWDG mitarbeiten, die für die Durchführung von Forschungsprojekten und den Support von Wissenschaftler*innen in Bereichen wie Datenmanagement und Datenanalyse zuständig ist. Sie arbeiten gemeinsam mit einem Team von Entwickler*innen und Wissenschaftler*innen im Themengebiet der Persistenten Identifikatoren und übernehmen unterschiedliche Projektaufgaben. Ihre Tätigkeiten fallen dabei konkret in folgende Bereiche:

- System-Administration und Optimierung der PID-Infrastruktur
- Implementierung von automatisierten Abläufen (Konfigurationsmanagement sowie CI/CD)
- Implementierung von PID-spezifischen Diensten
- Weiterentwicklung von Standards im Austausch mit internationalen Partnern

Eine enge Zusammenarbeit mit relevanten Gremien und Projekten der Nationalen Forschungsdateninfrastruktur (NFDI) und der European Open Science Cloud (EOSC) ist ebenfalls Teil Ihres Aufgabenbereiches.

Anforderungen

Sie haben Freude daran, in einem agilen Team

mitzuarbeiten, moderne Software zu entwickeln und dadurch Forschende bei ihrer Arbeit zu unterstützen? Das sind die idealen Voraussetzungen für diese Stellen. Des Weiteren erwarten wir von Ihnen:

- Ein abgeschlossenes Hochschulstudium (Bachelor oder Master), vorzugsweise in einem Fach wie Informatik, Digital Humanities, Physik oder Mathematik, oder eine vergleichbare Qualifikation mit einschlägiger Berufserfahrung
- Gute Programmierkenntnisse in einer beliebigen Hochsprache, wobei Python, Java, und/oder JavaScript von Vorteil sind
- Erfahrung im Bereich System-Administration von UNIX-basierten Betriebssystemen
- Gutes analytisches Denkvermögen
- Englischkenntnisse in Wort und Schrift

Zudem wären folgende Erfahrungen wünschenswert (aber nicht essenziell):

- Erfahrungen im Bereich der datenintensiven Wissenschaften
- Erfahrungen in der Nutzung von Datenbanken

Unser Angebot

- Flexible Arbeitszeiten und die Möglichkeit zu mobilem Arbeiten
- Ein modernes, vielfältiges und außergewöhnliches Arbeitsumfeld mit großer Nähe zu
- Wissenschaft und Forschung an der Schnittstelle mehrerer innovativer Technologiesektoren
- Eine interessante, vielseitige Tätigkeit in einem großen Forschungsprojekt
- Mitarbeit in einem kompetenten und engagierten Team
- Unterstützung bei der Qualifizierung und Weiterentwicklung Ihrer Fähigkeiten sowohl im akademischen Umfeld als auch mit Hinblick auf eine Karriere in der Wirtschaft
- Sozialleistungen des öffentlichen Dienstes

Die GWDG strebt nach Geschlechtergerechtigkeit und Vielfalt und begrüßt daher Bewerbungen jedes Hintergrunds. Die GWDG ist bemüht, mehr schwerbehinderte Menschen zu beschäftigen. Bewerbungen Schwerbehinderter sind ausdrücklich erwünscht.

Haben wir Ihr Interesse geweckt? Dann bitten wir um eine Bewerbung **bis zum 16.09.2024** über unser Online-Formular unter <https://s.gwdg.de/tVE9pL>.

Fragen zu den ausgeschriebenen Stellen beantwortet Ihnen:

Herr Dr. Tibor Kálmán

Tel.: 0551 39-30266

E-Mail: tibor.kalman@gwdg.de oder

Herr Dr. Sven Bingert

Tel.: 0551 39-30278

E-Mail: sven.bingert@gwdg.de



Stellenangebot

Nr. 20240529

Die GWDG sucht zur Unterstützung der Arbeitsgruppe „Nutzerservice und Betriebsdienste“ (AG H)

Studentische Hilfskräfte (m/w/d)

mit einer Beschäftigungszeit ab 30 Stunden im Monat. Die Vergütung erfolgt entsprechend den Regelungen für Studentische/Wissenschaftliche Hilfskräfte. Die Stellen sind zunächst auf ein Jahr befristet.

Die Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) ist eine gemeinsame Einrichtung der Georg-August-Universität Göttingen, der Stiftung Öffentlichen Rechts und der Max-Planck-Gesellschaft. Sie erfüllt die Funktion eines Rechen- und IT-Kompetenzzentrums für die Max-Planck-Gesellschaft und des Hochschulrechenzentrums für die Universität Göttingen. Zusammen mit der Universität ist die GWDG eines von neun Rechenzentren im Verbund Nationales Hochleistungsrechnen (NHR) und nationales KI-Servicezentrum. Zudem ist die GWDG Servicebetreiber für die Nationale Forschungsdateninfrastruktur (NFDI). Die wissenschaftlichen Forschungsaufgaben der GWDG liegen im Bereich der Angewandten Informatik. Ferner fördert sie die Ausbildung von Fachkräften für Informationstechnologie.

Aufgabenbereiche

- Mitarbeit im First-Level-Support (Helpdesk)
- Systemüberwachung und Peripheriebetreuung
- Mitarbeit im Empfang und in der Leitstelle des Göttinger Rechenzentrums

Diese Aufgaben sind unter der Anleitung wissenschaftlicher Mitarbeiter zu bearbeiten.

Die Arbeitszeiten liegen in den Öffnungszeiten der GWDG (Montag bis Freitag von 7:00 – 21:00 Uhr sowie Samstag und Sonntag von 10:00 – 18:00 Uhr).

Anforderungen

- Schnelle Lernfähigkeit
- Gute Kommunikations- und Teamfähigkeit
- Gute Deutsch- und Englischkenntnisse in Wort und Schrift
- IT-Kenntnisse erwünscht

Unser Angebot

- Ein modernes, vielfältiges und inspirierendes Arbeitsumfeld mit großer Nähe zu Wissenschaft und Forschung an der Schnittstelle mehrerer innovativer Technologiesektoren
- Eine interessante, vielseitige Tätigkeit in einem großen, überregional agierenden IT-Kompetenzzentrum
- Qualifizierung und Weiterentwicklung Ihrer Fähigkeiten

Die GWDG strebt nach Geschlechtergerechtigkeit und Vielfalt und begrüßt daher Bewerbungen jedes Hintergrunds. Die GWDG ist bemüht, mehr schwerbehinderte Menschen zu beschäftigen. Bewerbungen Schwerbehinderter sind ausdrücklich erwünscht. Haben wir Ihr Interesse geweckt? Dann bitten wir um eine Bewerbung über unser Online-Formular unter <https://s.gwdg.de/rTIDXM>.

Fragen zu den ausgeschriebenen Stellen beantwortet Ihnen:

Herr Nikolaj Kopp

Tel.: 0551 39-30286

E-Mail: nikolaj.kopp@gwdg.de oder

Herr Sebastian Pohl

Tel.: 0551 39-30297

E-Mail: sebastian.pohl@gwdg.de



NEUE AUSZUBILDENDE SVENJA BRENDIEK, HANS BRAUN-LÜLLEMANN UND EMELY ELKE KNÖFEL

Am 1. August 2024 hat Frau Svenja Brendiek ihre Ausbildung zur Fachinformatikerin (IHK) in der Fachrichtung Anwendungsentwicklung bei der GWDC begonnen. Sie gehört der Arbeitsgruppe „Basisdienste und Organisation“ (AG O) an. Frau Brendiek hat bereits im Rahmen verschiedener Praktika Erfahrungen im Bereich Grafikdesign gesammelt. Für ihre Ausbildung bringt sie erste Kenntnisse in der Programmierung mit, die sie während ihrer beruflichen Neuorientierung erlernt hat. Frau Brendiek ist per E-Mail unter svenja.brendiek@gwdg.de zu erreichen.



Lewandrowski

Am 1. August 2024 haben Herr Hans Braun-Lüllemann und Frau Emily Elke Knöfel ihre Ausbildung zum/zur Fachinformatiker*in (IHK) in der Fachrichtung Systemintegration bei der GWDC begonnen. Beide gehören der Arbeitsgruppe „Nutzerservice und Betriebsdienste“ (AG H) an.



Herr Braun-Lüllemann besuchte bis zu seinem Ausbildungsbeginn das Gymnasium und sammelte bereits durch private IT-Projekte erste Erfahrungen im Bereich Informatik und Systemadministration. Herr Braun-Lüllemann ist per E-Mail unter hans.braun-luellemann@gwdg.de zu erreichen.

Frau Emely Elke Knöfel besuchte das Gymnasium und absolvierte bis zu ihrem Ausbildungsstart ein FSJ Digital und konnte so bereits erste Erfahrungen und Einblicke in die Systemadministration und in den Aufbau von Hardware und Netzwerkkomponenten erlangen. Frau Knöfel ist per E-Mail unter emelyelke.knoefel@gwdg.de zu erreichen.



Quentin

Wir freuen uns auf eine spannende gemeinsame Ausbildungszeit mit herausfordernden Ausbildungsprojekten.



25-JÄHRIGES DIENSTJUBILÄUM THORSTEN HINDERMANN

Am 22. Juni 2024 feierte Herr Thorsten Hindermann sein 25-jähriges Dienstjubiläum im öffentlichen Dienst, wovon er seit seiner Einstellung im Jahre 2003 der GWDC die Treue hält. Nach seiner Ausbildung zum Physik-Laboranten absolvierte Herr Hindermann neben seiner Tätigkeit am III. Physikalischen Institut in Göttingen eine Fachausbildung zum staatlich geprüften Techniker in der Elektrotechnik mit Schwerpunkt Datenverarbeitungstechnik, die er 1995 erfolgreich beendete. Es zeigte sich aber schon früh, dass seine Interessen eher der IT und dort insbesondere der Programmierung galten. So baute sich Herr Hindermann in den folgenden Jahren ein umfangreiches und breites Wissen über verschiedene Programmiersprachen und -plattformen, IT-Technologien und -Strukturen sowie verschiedenste Betriebssysteme auf. In seiner Tätigkeit als Systemtechniker bei der GWDC brachte er seine Kenntnisse in zahlreiche Projekte ein und bewahrt sich bis heute ein großes Interesse an neuen IT-Technologien und -Trends, mit denen er sich weiterhin mit großem Engagement und Erfolg beschäftigt. Darüber hinaus hat sich Herr Hindermann in den vielen Jahren durch seine Betriebsratszugehörigkeit, die Mitarbeit im Arbeitsschutzausschuss sowie die Planung und Durchführung von spannenden und technisch fortschrittlichen Ausbildungsprojekten ganz wesentlich um die GWDC verdient gemacht. Deshalb möchten wir ihm heute ganz besonders für die sehr gute und vertrauensvolle Zusammenarbeit in all den Jahren danken und freuen uns schon jetzt auf viele weitere erfolgreiche Aktivitäten und Projekte. Wir gratulieren Herrn Hindermann ganz herzlich zum 25-jährigen Dienstjubiläum und wünschen ihm alles Gute für die Zukunft.

Pohl

INFORMATIONEN:
support@gwdg.de
0551 39-30000

September bis
Dezember 2024

Academy



KURS	DOZENT*IN	TERMIN	ANMELDEN BIS	AE
INDESIGN GRUNDKURS – SCHWERPUNKT POSTER-GESTALTUNG	Töpfer	10.09. – 11.09.2024 9:30 – 16:00 Uhr	03.09.2024	8
WORKING WITH GRO.DATA	Dr. Király	10.09.2024 14:00 – 15:30 Uhr	03.09.2024	0
INDESIGN – AUFBAUKURS	Töpfer	25.09. – 26.09.2024 9:30 – 16:00 Uhr	18.09.2024	8
INTRODUCTION TO PERFORMANCE ENGINEERING	Dr. Ogaja	26.09.2024 9:30 – 16:00 Uhr	19.09.2024	4
SUPERCOMPUTING FOR EVERY SCIENTIST	Eulert, Dr. Lüdemann	02.10.2024 9:00 – 12:00 und 13:00 – 16:00 Uhr	25.09.2024	4
PRACTICAL: HIGH-PERFORMANCE COMPUTING SYSTEM ADMINISTRATION	Prof. Kunkel	07.10. – 11.10.2024 9:00 – 18:00 Uhr	30.09.2024	20
VIRTUELLE CLOUD-INFRASTRUKTUREN – KURS FÜR AUSZUBILDENDE	Kopp	08.10. – 10.10.2024 9:00 – 16:00 Uhr	01.10.2024	12
WORKING WITH GRO.DATA	Dr. Király	08.10.2024 14:00 – 15:30 Uhr	01.10.2024	0
PERFORMANCE ANALYSIS OF AI AND HPC WORKLOADS	Dr. Ogaja, Dr. Lüdemann	22.10.2024 9:00 – 12:00 und 13:00 – 17:00 Uhr	15.10.2024	4
STORAGE – KURS FÜR AUSZUBILDENDE	Quentin	23.10. – 24.10.2024 9:00 – 12:30 und 13:30 – 15:30 Uhr	16.10.2024	8
INTRODUCTION TO ALPHA-FOLD	Lux, Mühlhausen, Paleico	29.10.2024 9:00 – 13:00 Uhr	22.10.2024	2

KURS	DOZENT*IN	TERMIN	ANMELDEN BIS	AE
AFFINITY PUBLISHER – SCHNUPPERKURS FÜR EINSTEIGER*INNEN	Töpfer	29.10.2024 10:30 – 12:30 und 13:30 – 15:30 Uhr	22.10.2024	3
DEEP LEARNING BOOTCAMP: BUILDING AND DEPLOYING AI MODELS	Lewis	04.11. – 05.11.2024 14:30 – 16:30 Uhr	28.10.2024	3
JOINT NHR DATA MANAGE- MENT TRAINING	Nolte, Oeste	05.11. – 06.11.2024 08:00 – 17:00 Uhr	29.10.2024	0
ADMINISTRATION IM ACTIVE DIRECTORY	Quentin, Hast, Kopp	06.11.2024 9:00 – 12:30 und 13:30 – 15:30 Uhr	30.10.2024	4
MONITORING WITH TIG STACK IN THE GWDC	Merz	07.11.2024 9:00 – 10:00 Uhr	31.10.2024	1
PARALLEL PROGRAMMING WITH MPI	Prof. Haan	12.11. – 13.11.2024 9:15 – 16:00 Uhr	05.11.2024	8
AFFINITY DESIGNER – SCHNUPPERKURS FÜR EINSTEIGER*INNEN	Töpfer	12.11.2024 10:30 – 12:30 und 13:30 – 15:30 Uhr	05.11.2024	3
WORKING WITH GRO.DATA	Dr. Király	12.11.2024 14:00 – 15:30 Uhr	05.11.2024	0
SECURE HPC – PARALLEL COMPUTING WITH HIGHEST SECURITY	Tabougua	18.11.2024 10:00 – 11:30 Uhr	11.11.2024	1
GPU PROGRAMMING WITH CUDA – AN INTRODUCTION	Prof. Haan	19.11.2024 9:15 – 16:00 Uhr	12.11.2024	4
EINFÜHRUNG IN DIE STATIS- TISCHE DATENANALYSE MIT SPSS	Cordes	20.11. – 21.11.2024 9:00 – 12:00 und 13:00 – 15:30 Uhr	13.11.2024	8
GRUNDLAGEN DER BILDBEAR- BEITUNG MIT AFFINITY PHOTO	Töpfer	26.11. – 27.11.2024 9:30 – 16:00 Uhr	19.11.2024	8
SECURE HPC – PARALLEL COMPUTING WITH HIGHEST SECURITY	Tabougua	02.12.2024 10:00 – 11:30 Uhr	25.11.2024	1
ANGEWANDTE STATISTIK MIT SPSS FÜR NUTZER*INNEN MIT VORKENNTNISSEN	Cordes	04.12. – 05.12.2024 9:00 – 12:00 und 13:00 – 15:30 Uhr	27.11.2024	8
DEEP LEARNING BOOTCAMP: BUILDING AND DEPLOYING AI MODELS	Lewis	09.12. – 10.12.2024 14:30 – 16:30 Uhr	02.12.2024	3
WORKING WITH GRO.DATA	Dr. Király	10.12.2024 14:00 – 15:30 Uhr	03.12.2024	0

Teilnehmerkreis

Das Angebot der GWDG Academy richtet sich an die Beschäftigten aller Einrichtungen der Universität Göttingen, der Max-Planck-Gesellschaft sowie aus wissenschaftlichen Einrichtungen, die zum erweiterten Kreis der Nutzer*innen der GWDG gehören. Studierende am Göttingen Campus zählen ebenfalls hierzu. Für manche Kurse werden spezielle Kenntnisse vorausgesetzt, die in den jeweiligen Kursbeschreibungen genannt werden.

Anmeldung

Für die Anmeldung zu einem Kurs müssen Sie sich zunächst mit Ihrem Benutzernamen und Passwort in der GWDG Academy (<https://academy.gwdg.de>) einloggen. Wenn Sie zum Kreis der berechtigten Nutzer*innen der GWDG gehören, erhalten Sie anschließend automatisch Zugang zu unserem Kursprogramm. Sollten Sie noch keinen Account besitzen, können Sie sich unter <https://id.academiccloud.de> registrieren und müssen ggf. auf Anfrage für die Anmeldung zu unseren Kursen freigeschaltet werden. Bei Online-Kursen kann das Anmeldeverfahren abweichen. Genauere Informationen dazu finden Sie in der jeweiligen Kursbeschreibung. Einige Online-Angebote stehen Ihnen jederzeit und ohne Anmeldung zur Verfügung.

Absage

Absagen können bis zu sieben Tagen vor Kursbeginn erfolgen. Bei kurzfristigeren Absagen werden allerdings die für den Kurs angesetzten Arbeitseinheiten (AE) vom AE-Kontingent der jeweiligen Einrichtung abgezogen.

Kursorte

Die Kurse finden entweder in einem geeigneten Online-Format oder als Präsenzkurs statt. Nähere Informationen dazu finden Sie bei den jeweiligen Kursen. Auf Wunsch und bei ausreichendem Interesse führen wir auch Kurse vor Ort in einem Institut durch, sofern dort ein geeigneter Raum mit entsprechender Ausstattung zur Verfügung gestellt wird.

Kosten bzw. Gebühren

Die Academy-Kurse sind – wie die meisten anderen Leistungen der GWDG – in das interne Kosten- und Leistungsrechnungssystem der GWDG einbezogen. Die den Kursen zugrundeliegenden AE werden vom AE-Kontingent der jeweiligen Einrichtung abgezogen. Für alle Einrichtungen der Universität Göttingen und der Max-Planck-Gesellschaft sowie die meisten der wissenschaftlichen Einrichtungen, die zum erweiterten Kreis der Nutzer*innen der GWDG gehören, erfolgt keine Abrechnung in EUR. Dies gilt auch für die Studierenden am Göttingen Campus.

Kontakt und Information

Wenn Sie Fragen zum aktuellen Academy-Kursangebot, zur Kursplanung oder Wünsche nach weiteren Kursthemen haben, schicken Sie bitte eine E-Mail an support@gwdg.de. Falls bei einer ausreichend großen Gruppe Interesse besteht, könnten u. U. auch Kurse angeboten werden, die nicht im aktuellen Kursprogramm enthalten sind.



FTP-Server

Eine ergiebige Fundgrube!

Ihre Anforderung

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

Unser Angebot

Die GWDG betreibt seit 1992 einen der weltweit bekanntesten FTP-Server mit leistungsfähigen Ressourcen und schneller Netzanbindung. Er ist dabei Hauptmirror für viele Open-Source-Projekte.

Ihre Vorteile

- > Großer Datenbestand (65 TByte), weltweit verfügbar
- > Besonders gute Anbindung im GÖNET

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

Interessiert?

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

>> www.gwdg.de/ftp-server





Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen