

Inhalt



Editorial	2
Iterationsschleife	6
DASH - Distributed Data Structures in a Global Address Space	8
Ferienakademie 2018	18
Neues aus der KONWIHR Ecke	20
SPPEXA News	22
Notiz*Notiz*Notiz	23

Das Quartl erhalten Sie online unter <http://www5.in.tum.de/quartl/>



Das Quartl ist das offizielle Mitteilungsblatt des *Kompetenznetzwerks für Technisch-Wissenschaftliches Hoch- und Höchstleistungsrechnen in Bayern* (KONWIHR) und der *Bavarian Graduate School of Computational Engineering* (BGCE)

Editorial

Wie lange kann das mit unserem Begutachtungssystem noch gutgehen? Diese Frage haben Sie sich sicher auch schon mal gestellt. Egal ob Promotionen, Projektanträge, Tagungsbeiträge, Publikationen in Zeitschriften, Wettbewerbe oder Auszeichnungen – immer soll ausgewählt oder bewertet werden, und immer wird auf die gleiche zahlenmäßig sehr beschränkte Gruppe zurückgegriffen, die so genannten Peers.

Ich habe für mich mal ausgerechnet, wie viel Zeit ich denn im vergangenen Jahr durchschnittlich für jeden solchen Akt aufbringen konnte (wobei das „Können“ zunächst nur durch die 24 Stunden eines Tages sowie die ja noch anderen anstehenden Aufgaben bestimmt war, nicht etwa durch Dinge wie Schlafbedarf, Lust, Freizeit oder Reduktion der Gewissenhaftigkeit). Das Ergebnis ist niederschmetternd: Entweder ich habe den „evaluatorischen Blick“ und bin in der Lage, in kürzester Zeit den wissenschaftlichen Gehalt eines Schriftstücks festzustellen oder die Einschätzung einer Person vorzunehmen, oder aber mit Qualitätskontrolle hat das alles nicht mehr viel zu tun. Dieser Selbsterstickungseffekt des Systems hat verschiedene Gründe.

Erstens wird heute niemandem mehr eine Entscheidung „aus dem Bauch“ zugezogen – bei Prüfungen nicht und bei obigen Dingen erst recht nicht. Schnell ist die Rede von Beliebigkeit, Willkür oder gar Gutsherrenart, und laut ist der Ruf nach messbaren Kriterien. Es sieht halt so herrlich objektiv aus, wenn nach einem dreimonatigen Prozess bei der einen Kandidatin 41.5 Punkte und bei der anderen 42.5 Punkte stehen. Niemand hinterfragt das Zustandekommen der Punkte (allenfalls, um anschließend ein noch aufwändigeres und feingranulareres Verfahren vorzuschlagen), niemand muss sich selbst einen Vorwurf machen oder von anderen einen solchen anhören, alle scheinen über das objektive Ergebnis glücklich zu sein (OK, die Kandidatin mit 41.5 Punkten vielleicht nicht). Wir alle haben das inzwischen so stark verinnerlicht, dass wir uns selbst nichts mehr zutrauen – obwohl wir eigentlich schon ganz gut dafür ausgebildet und erfahren sind, und obwohl schnelle, intuitive Entscheidungen beispielsweise bei Ärzten oder Piloten ja so unüblich nicht sind. „Gibt es dafür einen Prozess?“, „Gibt es Kriterien?“, „Wie stelle ich denn die

Vergleichbarkeit sicher?“ – all das kriegt man immer dann zu hören, wenn man irgendwo mal schnell etwas Unaufwändiges einführen möchte. Alles versteckt sich hinter der vermeintlichen Objektivität – die eben nur noch eine vermeintliche ist, wenn ich für das Gutachten über einen 20-seitigen Projektantrag maximal eine Stunde Zeit zur Verfügung habe.

Zweitens hat sich dieser Begutachtungswahnsinn inzwischen schon auf mehreren Ebenen breit gemacht. Beantragt man z.B. ein größeres Verbundprojekt, so sind Mechanismen einer internen Begutachtung und qualitätsbezogenen Mittelvergabe schon fast die Regel. Der SFB scheint die Grenze dessen zu markieren, bei dem Gutachterinnen und Gutachter noch im Detail a priori festlegen können, was wie gefördert werden soll. Bei allem Größeren wird beim Zuschlag vor allem die große Linie bewertet, für die faktische Verteilung verlässt man sich anschließend auf entsprechend definierte Prozesse. An sich ein schöner Gedanke, nur erhöht das die Zahl der Involvierungen weiter. Eine solche Erfahrung machen wir z.B. auch immer wieder bei der IGSSE, unserer International Graduate School of Science and Engineering. Zwei Runden lang aus der Exzellenzinitiative gefördert, stehen inzwischen dauerhaft TUM-Mittel zur Verfügung. Alle Anträge werden von mehreren Sachverständigen extern begutachtet, was zunächst mal schon beim Aufspüren der Gutachterinnen und Gutachter zu riesigem Aufwand führt – manchmal muss man fünf fragen, um eine Zusage zu erhalten. Wobei das „aussagekräftige“ Gutachten dann nicht selten aus zehn Kreuzchen bei „outstanding“ besteht. Ich mache da niemandem einen Vorwurf – ich weiß ja selbst, wie das ist. Denn so manch einer bzw. eine fragt sich nicht ganz zu unrecht, wieso er/sie an der Universität XYZ eigentlich Zeit investieren soll, um das Qualitätsmanagement der TUM zu garantieren: „Glauben die bei euch, ich habe nichts zu tun?“ So habe ich dann bei der letzten Auswahlrunde der IGSSE als Reaktion auf eine wirklich nicht ganz glücklich formulierte Anfrage eine Email einer befreundeten Kollegin aus dem Schwäbischen erhalten der Art (frei übersetzt, ich geb's zu) „Sind die bei euch jetzt völlig durchgeknallt?“ Die Antwort: Die bei uns nicht, zumindest nicht mehr als die anderswo. Das System ist irgendwo durchgeknallt. Sind die Begutachteten wenigstens damit zufrieden? Weit gefehlt! Kaum eine Runde, nach der nicht irgendein(e)

Abgewiesene(r) rumrennt und über maximale Beliebigkeit und Intransparenz schimpft. Was dann wiederum die ärgert oder frustriert, die den Mechanismus am Laufen halten.

Drittens gibt es immer mehr Förderinstrumente – was ja zunächst schön ist. Aber alle werden kompetitiv aufgesetzt, die Mittel sollen immer in einem wettbewerblichen Verfahren vergeben werden, wobei man sich praktisch immer der Hilfe von externen Sachverständigen bedient – der Qualität wegen. Doch auch da gilt: Liebe Leute, die Zahl solcher Personen ist begrenzt!

Viertens wird das Instrument „Gutachten“ heute viel zu breit eingesetzt. Will eine Studierende ins Ausland, wird zweimal begutachtet – zunächst, ob sie ins Austauschprogramm der Heimat-Uni aufgenommen wird und daraus eine Förderung erhält. Und dann schaut sich die Gastgeber-Uni alles an, ob man sich der Auswahl wirklich anschließt – wofür natürlich wieder „Letters of Recommendation“ anzufertigen sind. Manchmal habe ich den Eindruck, dass jeder Klogang von Empfehlungsschreibern flankiert und abgesichert werden muss. Oder aktuell die Anfrage einer bayerischen Partner-Uni, ob ein ausgeguckter Wissenschaftler wirklich zum „apl. Prof.“ taugt. Herrje, der Kerl war in mehreren Verfahren in der Endrunde, hat gerade einen Ruf – kann da nicht einfach ein Dekan oder ein Fakultätsrat sagen „Wir machen das jetzt!“? Eine zusätzliche Schiefelage entsteht dadurch, dass die Gutmütigen immer mehr abkriegen, weil viele andere es sich leicht machen, z.B. mit einem Argument wie „Das ist nicht mein Fachgebiet!“ die Begutachtung ablehnen. (In diesem Zusammenhang eine Warnung: Seit Jahren führe ich Buch über jede einzelne Absage, die ich mir einfange, und die Webseite „Wer nach eigener Aussage von was keine Ahnung hat“ ist in Vorbereitung, das Material hierfür wächst unaufhörlich. Das dürfte für etliche Personen hochnotpeinlich werden – denn natürlich führen sie zu Themen, von denen sie angeblich keine Ahnung haben, beispielsweise Forschungsprojekte durch ... Aber dies nur am Rande.)

Gegenmaßnahmen? Meistens sind die eher grotesker Art. So kam neulich für das Design des Verfahrens bei der Evaluierung unserer Tenure-Track-Professuren allen Ernstes folgender Vorschlag: Da man ja wisse, dass die

meisten dieser Gutachten mit heißer Nadel gestrickt sind, müsse man die Verlässlichkeit durch Statistik erzielen – man möge doch jeweils zehn Gutachten einholen (davon dann natürlich sechs international, fünf von Frauen, und aus Objektivitätsgründen mindestens drei von Experten, die vom zu Begutachtenden noch nie etwas gehört haben, sprich fachlich völlig unvorbelastet sind, etc. etc.). So kann man einen GAU aktiv herbeiführen.

Doch es gibt auch konstruktive Vorschläge. An verschiedenen Stellen wird in Projekten oder Arbeitsgruppen nach alternativen Selektionsprinzipien gefahndet, wobei da durchaus auch König Zufall konsultiert wird. Warum nicht bei einer Ausschreibung die (doch oft ziemlich offensichtlichen) Überflieger einfach durchwinken, die chancenlosen Fälle (die wir etwa aus der berühmten Kategorie C bei Berufungsverfahren kennen) einfach ausmustern, und unter dem Rest dann das Los entscheiden lassen? Das mag absurd, ungerecht, unwissenschaftlich oder gottlos anmuten, aber immer mehr sind der Ansicht, dass es nicht unbedingt schlechter als die aktuelle Praxis ist. Und einfacher ist es auf jeden Fall, was uns mehr Zeit für unsere eigentlichen Aufgaben lässt. Das wäre doch was. Bei der IGSSE überlegen wir uns das auch.

Ach ja, ich habe übrigens noch eine zweite Email von besagter Kollegin aus dem Schwäbischen erhalten, verbunden mit der nachdrücklichen Bitte auf Aufnahme ins Quartl (der ich gerne nachgekommen bin): Nachdem sie das mit dem Schnell-Gutachten für die IGSSE netterweise gemacht hatte, flogen ihr diverse Devotionalien zu – eine Tasse, ein Kuli, ein Block ... Einerseits einfach eine nette Geste, andererseits die Fortsetzung des Absurden. So hat halt alles zwei Seiten, mindestens.

Einen hab ich noch: Folgende Anfrage eines Teilnehmers an einer von uns organisierten Tagung erreichte uns neulich: „I am planning to attend XYZ and would like to arrange accommodation at the Hotel ABC. Is it possible to get a room with a sofa?“ Mei, was so alles gewünscht wird ...

Die gesamte Quartl-Redaktion wünscht Ihnen, liebe Leserinnen und Leser, Frohe Ostern, eine entspannte und erfolgreiche Eiersuche sowie wenige Anfragen, ob Sie nicht schnell ein Gutachten schreiben könnten ...

Hans-Joachim Bungartz

Iterationsschleife

N=26

27. Februar 2018

Der Abgang ist entscheidend. Aus dem Abgang – so heißt es ^a - lasse sich auf die Qualität schließen. Bei mancher Leistung ist also der Abgang, wenn nicht allein entscheidend so doch ausschlaggebend für die Bewertung des gesamten präsentierten Werks. Selbst der Schauspieler ist klar, dass sich vieles zwischen Auftritt und Abgang abspielt, dass aber der zeitlich und dramatisch richtig gewählte Abgang entscheidend für den letzten Eindruck der Kritikerin sein kann.

Ein starker Abgang kann alles retten oder zumindest den Betrachter mit dem Gefühl zurücklassen, zu früh geurteilt zu haben und nun von einem richtig guten Abgang eines Besseren belehrt zu werden. Zeichnet sich der Abgang deutlich ab, so steigt die Wertschätzung mitunter dramatisch an. Was und wer lange Zeit als mittelmäßig galt, kann nun durch die Andeutung oder auch die Androhung des Abgangs seine Sympathiewerte steigern. So mancher italienische Ministerpräsident etwa hat seine Amtszeit verlängert, indem er dem Parlament bewusst die Frage stellte: wollt Ihr meinen Abgang? Die Politik nennt das: die Vertrauensfrage stellen. So etwas kommt mitunter sehr oft vor und zeigt, dass der angedrohte Abgang manchmal der einzige sein kann, den man mit Sicherheit vermeidet^b. Gefährlich ist es, wenn man aus der Sympathie für den sich ankündigenden Abgang auf eine gleiche Sympathie für das Bleiben zu schliessen versucht. Wer als Abgänger beliebt ist, muss es nicht als Dauergast sein.

Was den Abgang so schwer macht, ist wohl das Gefühl der Sicherheit die in der Übung, dem Amt, der Bühne oder der Öffentlichkeit liegt. Wer geht schon gerne, wenn der Fluss noch trägt, wenn das Publikum noch atemlos lauscht oder wenn noch alle Probleme sich als lösbar erweisen? Wie sieht man, ob die Wertschätzung für das Amt an sich längst die Wertschätzung für den Amtsträger abgelöst hat? Denn, wie ist dieser Unterschied zu erkennen? Und wo sind die Freunde, die sehr höflich und direkt wagen zu fragen „ich schätze Deine Arbeit sehr, aber wäre es nicht Zeit für den Abgang“?

^a[https://de.wikipedia.org/wiki/Abgang_\(Nahrungsmittel\)](https://de.wikipedia.org/wiki/Abgang_(Nahrungsmittel))

^b<http://www.spiegel.de/politik/ausland/berlusconis-vertrauensfrage-showdown-nummer-51-fuer-den-cavalierere-a-791707.html>

Eine professorale Untersuchung des Phänomens hilft wenig weiter, denn gerade professorale Abgänge gehören zu den schwierigsten auf dieser Welt. So meinte etwa einmal ein 77-jähriger Kollege, sein Co-Direktor habe noch nicht verstanden, dass er zu alt sei, während eben dieser Co-Direktor meinte, sein 78-jähriger Kompagnon habe die Zeichen der Zeit noch nicht erkannt.

Der Splitter im Auge des anderen mag sichtbar sein. Aber wo ist der Balken, den man im eigenen Auge übersieht?

M. Resch

DASH – Distributed Data Structures in a Global Address Space

In 1980 an Intel 8088/87 CPU could perform a floating point operation in around 10 microseconds. In the same amount of time a whopping 20 bytes of data could be fetched from memory [1]. The computer systems of that period were largely limited by their ability to perform arithmetic operations — How profoundly things have changed since then! Today, the performance most applications can hope to extract from a system is often not limited by the raw CPU computing power in terms of floating point operations per second (FLOPs), but by the time it takes to feed the data into the processor’s compute units. Data thus has a place at the center of attention for HPC developers and the DASH project aims to provide a programming environment that makes working with distributed data structures easier.

Introduction As the CPU manufacturing process evolves into using smaller feature sizes, data transfers account for an ever increasing fraction of the time and energy budget of a CPU and this situation is expected to get significantly worse in the future [2]. At the same time, HPC systems are getting bigger in terms of core and node counts and data locality must be taken into account both horizontally (between shared memory domains) and vertically (within a shared memory domain) [3]. Data thus needs to have a place at the center of attention for HPC application developers. This is where the DASH project aims to make contributions by providing distributed data structures and parallel algorithms similar in spirit to what is provided by the C++ STL (standard template library). The goal of DASH is to make working with data easier and more productive for HPC developers.

The overall structure of the DASH project is shown in Fig. 1. DASH makes use of existing one-sided communication substrates and is based upon a runtime system called DART. This runtime system provides central abstractions such as global memory allocation and addressing and offers one-sided communication operations (puts and gets). The execution model adopted by DART follows SPMD (single program, multiple data) semantics, where the

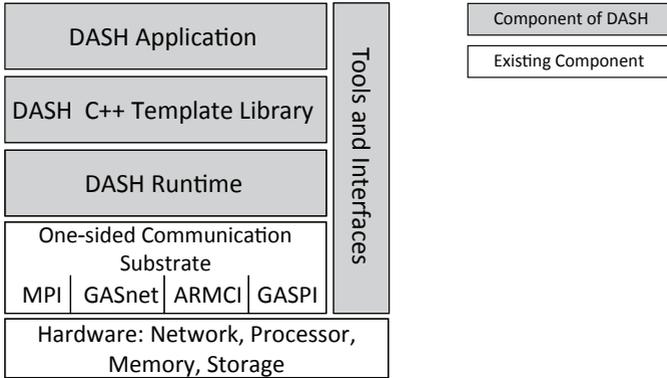


Figure 1: The architecture of the DASH PGAS framework.

individual participants are called *units*. Units can be hierarchically grouped into *teams* to address the growing size of systems and their increasingly multi-leveled organization. To enable seamless integration with the large number of existing MPI applications, DART is based on the MPI-3 RMA (remote memory access) interface [5].

The DASH project started in 2013 with the first funding phase of SPPEXA and continued into the second phase with a consortium of four German partners. The project is led by LMU Munich, where also the bulk of the C++ library development is situated. The partners in HLRS Stuttgart, IHR Stuttgart, and TU Dresden contribute expertise in runtime system development, application engagement and tools integration, respectively.

Data Structures with Global-view and Local-view Semantics

DASH offers data structures that mimic the interface and behavior of STL containers. The most basic data structure available in DASH is a fixed-size one-dimensional array. `dash::Array<int> arr(1000)` declares such an array of one thousand integers, where the data to store the individual elements is contributed by all units that execute the program. The array `arr` is said to have *global-view* access semantics, since each process has a unified

(global) view of the container — `arr.size()` returns the same (global) size and `arr[42]` refers to the same element on each unit. Global-view access is very convenient when working with data structures with dynamically changing and irregular data structures but it comes with an overhead in terms of access cost, since for each access a locality check and a network transfer might be needed to retrieve or store remote data.

To avoid such access overheads entirely when working with known local data, DASH also supports a *local-view* access mode, using the `.local` accessor object. For example `arr.local.size()` returns the number of data elements available locally, `arr.local[0]` returns the first element stored locally, and so on. Using this form of local-view access has large performance benefits and allows for a straightforward realization of the *owner computes* parallel computation model.

Besides the basic one-dimensional fixed-size array, DASH also supports multidimensional arrays (`dash::NArray`) where support for slicing in arbitrary dimensions is included. Dynamic (growing/shrinking) data structures are under development, including dynamic lists and hashmaps.

Data Distribution Patterns In large-scale HPC systems, parallelism implies distribution, i.e., several compute nodes are interconnected by some form of high-speed interconnect network. The way in which the data that is being operated on is distributed among these nodes can have an important influence on program performance. Since DASH offers distributed data structures than can span multiple interconnected compute nodes, it also has to provide flexible ways in which to specify data distributions.

In DASH this is achieved by specifying a so-called *pattern*. The pattern determines how data elements are distributed among a set of units and how the iteration sequence is determined. In one dimension, DASH offers the usual choice of a blocked, cyclic, and block-cyclic distribution. In multiple dimensions these specifications can be combined in each dimension. Additionally, a tiled distribution is supported where contiguous blocks of the iteration space are specified.

Fig. 2 shows several examples for DASH data distribution patterns in two dimensions. The colors correspond to the processes (units), the iteration sequence is additionally visualized for unit 0 using connected dots. DASH also supports both row-major as well as column-major storage order.

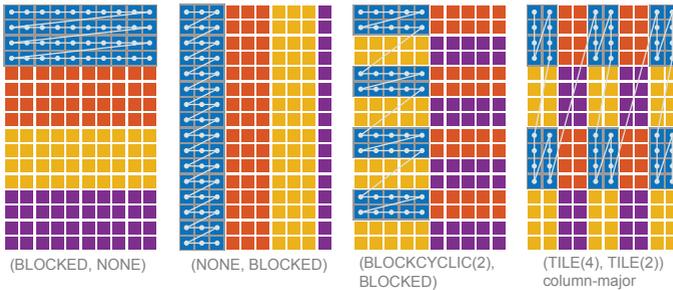


Figure 2: Several examples for 2D DASH data distribution patterns.

Productivity through Standard Algorithms Distributed data structures are convenient, but ultimately developers are interested in performing computations on the data in an efficient and productive way. In addition to element-wise and bulk data access feeding into existing code, DASH also offers inter-operability with sequential STL algorithms and provides a set of parallel algorithms modeled after their STL counterpart.

As an example, standard algorithms such as `std::sort` or `std::fill` can be used in conjunction with the global-view and the local-view mode of a DASH container.

`std::sort(arr.local.begin(), arr.local.end())` sorts the local portion of the distributed array and

`std::min_element(arr.begin(), arr.end())` finds the smallest element in the whole array. In the latter example, the minimum is found without exploiting the available parallelism, since the STL algorithm cannot be aware of the data distribution and available compute units. When the algorithm provided by DASH `dash::min_element(arr.begin(), arr.end())` is used instead, all units collaboratively find the global minimum by first finding their local minima and then collectively determining the global minimum.

The usage of DASH algorithms can enhance programmer productivity significantly. Instead of the classic imperative programming style commonly used in C/C++ or Fortran MPI codes, the usage of algorithms provides a more declarative style that is both more compact and invariant under changes of the underlying data distribution.

Fig. 3 shows a basic complete DASH program using a 2D array (matrix) data structure. The data type (`int`) and the dimension (`2`) are compile-time template parameters, the extents in each dimension are set at runtime. In the example a 10 x 8 matrix is allocated and distributed over all units (since no team is specified explicitly). No specific data distribution pattern is requested, so the default distribution by block of rows over all units is used. When run with four units, each unit gets $\lceil 10/4 \rceil$ matrix rows, except for the last unit, which receives only one row.

```
1 #include <iostream>
2 #include <libdash.h>
3
4 int main(int argc, char *argv[]) {
5     dash::init(&argc, &argv);
6
7     // 2D integer matrix with 10 rows, 8 cols
8     // default distribution is blocked by rows
9     dash::NArray<int, 2> mat(10, 8);
10
11     for (int i=0; i<mat.local.extent(0); ++i) {
12         for (int j=0; j<mat.local.extent(1); ++j) {
13             mat.local(i, j) = 10*dash::myid()+i+j;
14         }
15     }
16
17     dash::barrier();
18
19     auto max = dash::max_element(mat.begin(), mat.end());
20
21     if (dash::myid() == 0) {
22         print2d(mat);
23         cout << "Max is " << (int)(*max) << endl;
24     }
25
26     dash::finalize();
27 }
```

Figure 3: A basic example DASH application, making use of a distributed twodi-dimensional array and a parallel algorithm.

Lines 11 to 15 in Fig. 3 show data access using the local matrix view by using the proxy object `mat.local`. All accesses are performed using local indices (i.e., `mat.local(1,2)` refers to the element stored locally at position (1,2)) and no communication operation is performed. The barrier in line 17 ensures that all units have initialized their local part of the data structure before the `max_element()` algorithm is used to find the maximum value of the whole matrix. This is done by specifying the global range that encompasses all matrix element (`mat.begin()` to `mat.end()`). In the library implementation of `max_element()`, each unit determines the locally stored part of the global range and performs the search for the maximum there. Afterwards a reduction operation is performed to find the global maximum. The return value of `max_element()` is a global reference for the location of the global maximum. In lines 21 to 24, unit 0 first prints the whole matrix (the code for `print2d()` is not shown) and then outputs the maximum by dereferencing the global reference `max`.

Fig. 4 shows the output produced by this application and how to compile and run the program. Since DASH is implemented on top of MPI, the usual platform-specific mechanisms for compiling and running MPI programs are used. The output shown is from a run with four units (MPI processes), hence the first set of three rows are initialized to 0...9, the second set of three rows to 10...19, and so on.

Memory Spaces and Locality Information To address the increasing complexity of supercomputer systems in terms of their memory organization and hardware topology, work is currently under way in DASH to offer constructs for productive programming constructs dealing with novel hardware features such as non-volatile and high-bandwidth memory. These additional storage options will be represented as dedicated memory spaces and the automatic management and promotion of parts of a data structure to these separate memory space will be available in DASH. Additionally, a locality information system is under development which supports an application-centric query and exploitation of the available hardware topology on a specific machine.

Compile and Run:

```
$> mpicc -L ... -ldash -o example example.cc  
$> mpirun -n 4 ./example
```

Output:

```
0 1 2 3 4 5 6 7  
1 2 3 4 5 6 7 8  
2 3 4 5 6 7 8 9  
10 11 12 13 14 15 16 17  
11 12 13 14 15 16 17 18  
12 13 14 15 16 17 18 19  
20 21 22 23 24 25 26 27  
21 22 23 24 25 26 27 28  
22 23 24 25 26 27 28 29  
30 31 32 33 34 35 36 37  
Max is 37
```

Figure 4: Commands to compile and run the DASH application and the output produced by the program.

Conclusion DASH is a C++ template library that offers distributed data structures with flexible data partitioning schemes and a set of parallel algorithms. Stand-alone applications can be written using these features but DASH also allows for integration into existing MPI codes. In this scenario, individual data structures can be ported to DASH, incrementally moving from existing two-sided communication operations to the one-sided operations available in DASH. DASH is available as open source software under a BSD clause and is maintained on GitHub (<https://github.com/dash-project/dash>). Additional information on the project, including tutorial material, can be found on the projects webpage at <http://www.dash-project.org/>. More information can also be found in a recent overview paper [4].

Karl Furlinger, LMU München

Acknowledgements The author would like to thank the members of the DASH team and gratefully acknowledges funding by the German Research Foundation (DFG) through the German Priority Programme 1648, Software for Exascale Computing (SPPEXA).

References

- [1] McCalpin, John D. “A survey of memory bandwidth and machine balance in current high performance computers.” *IEEE TCCA Newsletter* 19 (1995): 25.
- [2] Ang, James A., et al. “Abstract machine models and proxy architectures for exascale computing.” *Hardware-Software Co-Design for High Performance Computing (Co-HPC)*, 2014. IEEE, 2014.
- [3] Didem Unat, et al., “Trends in Data Locality Abstractions for HPC Systems,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 3007-3020, Oct. 1 2017.
- [4] Karl Furlinger, Tobias Fuchs, and Roger Kowalewski. “DASH: a C++ PGAS library for distributed data structures and parallel algorithms.” *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2016 IEEE 18th International Conference on. IEEE, 2016.
- [5] Zhou, Huan, et al. “DART-MPI: an MPI-based implementation of a PGAS runtime system.” *Proceedings of the 8th International Conference on Partitioned Global Address Space Programming Models*. ACM, 2014.

Ferienakademie 2018

Wie seit über 30 Jahren wird es auch heuer wieder ein buntes Kursprogramm bei der Ferienakademie im Sarntal (Südtirol) geben, das Dozenten der drei ausrichtenden Universitäten FAU Erlangen-Nürnberg, Stuttgart und TUM in standort- und oftmals auch fächerübergreifenden Kursen anbieten (Details siehe Bild).

Nach einer langen Bewerbungsphase, die bis 02. Mai 2018 andauern wird, werden dann die zugelassenen Kandidaten der drei ausrichtenden Universitäten durch eine Auswahljury der Ferienakademie ermittelt. Für diese Teilnehmer geht es dann in den letzten beiden Wochen des September ins Sarntal.



Neugierig?

Die Themen und das Zielklientel der Kurse sind im Ferienakademieposter (vgl. Abb. 1) aufgeführt. Wir freuen uns schon auf spannende Diskussionen vor Ort im Herbst.

Tobias Neckel

Ferienakademie



23.09. – 5.10. 2018
Sarntal (Südtirol)

18
20

Programm

1	Der Sarntaler: eine digitale Währung für die Ferienakademie	D. Schröder, Erlangen H. Seidl, München	Informatik, Wirtschaftsinformatik, Mathematik (Bachelor ab 1. oder 2. Studienjahr)
2	SarntalX: Agile Aircraft Design	B. Brügge, München S. Wagner, Stuttgart	Informatik, Software Engineering, Elektrotechnik, Luft- und Raumfahrtstechnik, Maschinenwesen, Industrial Design, Human Factors Engineering (Bachelor ab 2. Studienjahr oder Master)
3	Physik und Elektronik im Alltag	G. Denninger, Stuttgart R. Gross, München V. Krstic, Erlangen (GD)	Physik, Elektro- und Informationstechnik (Bachelor im 1. oder 2. Studienjahr)
4	Multiscale Problems in Mechanics: Models – Simulation – Application	E. Rank, München H. Steeb, Stuttgart	Ingenieurwissenschaften, Mathematik, Informatik (Master)
5	Simulation of Fluids and Wave Phenomena – High Order, High Performance, High Productivity	M. Bader, München C.-D. Munz, Stuttgart D. Fey, Erlangen (GD) H. Köster, Erlangen (GD)	Ingenieurwissenschaften, Mathematik, Informatik, Physik (Bachelor ab 3. Studienjahr oder Master)
6	Accelerating Physics Simulations with Deep Learning	M. Engel, Erlangen N. Thuerey, München M. Meth, Stuttgart (GD)	Informatik, Mathematik, Physik, Verfahrenstechnik, Chemie, Ingenieurwissenschaften (Bachelor ab 3. Studienjahr oder Master)
7	Energiewende: Faktencheck mit (Big) Data Analytics	K. Diepold, München J. Karl, Erlangen	Ingenieurwissenschaften, Naturwissenschaften, Wirtschaftswissenschaften, Mathematik, Informatik (alle Fachsemester)
8	Simulation Technology: From Models to Software	B. Flemisch, Stuttgart B. Wohlmut, München G. Leugering, Erlangen (GD)	Mathematik, Informatik, Physik, Ingenieurwissenschaften, Simulation Technology (Bachelor ab 2. Studienjahr oder Master)
9	Redundancy and Irrelevance in Source and Channel Coding	B. Edler, Erlangen O. Kramer, München A. Kaup, Erlangen (GD)	Elektrotechnik, Informations- und Kommunikationstechnik, Informatik, Mathematik, Physik (Bachelor ab 3. Studienjahr oder Master)
10	Computational Medical Imaging	T. Lasser, München A. Maier, Erlangen	Medizintechnik, Informatik, Elektrotechnik, Mathematik, Physik (Bachelor ab 3. Studienjahr oder Master)
Organisation: T. Neckel, München, neckel@in.tum.de A. Spruck, Erlangen, andreas.spruck@fau.de F. Berkmann, Stuttgart, ferienakademie@it.uni-stuttgart.de		Universitätsbeauftragte: E. Rank, München A. Kaup, Erlangen J. Schultze, Stuttgart	
		Direktor: H.-J. Bungartz, Institut für Informatik, TU München	

Bewerbungsschluss 2. Mai 2018



**FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG**



**Technische
Universität
München**



Universität Stuttgart

www.ferienakademie.de

Abbildung 1: Poster der Ferienakademie 2018

Neues aus der KONWIHR Ecke

Heimlich, still und leise hat sich in den letzten Jahren in unserem Kompetenznetzwerk viel verändert und getan. Mitbekommen haben es leider die wenigsten, da es ja selbst unsere – freundlich formuliert – „angestaubte“ Webseite nicht gemerkt hat. Und hier geloben wir nicht nur Besserung, sondern sind gerade dabei, unseren Webauftritt etwas wiederaufzufrischen. In Zukunft sollen alle Interessenten sich wieder adäquat über die vielfältigen und erfolgreichen KONWIHR Projekte informieren können.

Doch bis es soweit ist, möchte ich hier kurz über die wichtigsten personellen Änderungen und über den Ausgang der letzten Antragsrunde berichten. Zunächst ist natürlich die lautlose Stabübergabe in der Rolle des Sprechers zu nennen. Prof. Bode ist nach vielen Jahren als Sprecher ausgeschieden und hat den KONWIHR Stab an Prof. Bungartz übergeben. An dieser Stelle möchte sich das Kompetenznetzwerk herzlichst bei Prof. Bode für seine engagierte und erfolgreiche Tätigkeit bedanken – für ihn war und ist die Förderung der Optimierung und Parallelisierung von Anwenderprogrammen immer untrennbar mit der Beschaffung neuer Rechner verbunden. Im Kielwasser mit dem Ausscheiden von Prof. Bode hat dann auch Herr Meyer die Geschäftsstellenaufgaben mit zeitlichem Versatz abgegeben. Auch ihm herzlichen Dank für die sehr gute und hilfreiche Tätigkeit die immer einen reibungslosen Antrags- und Begutachtungsprozess sicherstellte. Seine Aufgaben hat seit 1.1.2018 Julian Hornich an der FAU inne, und als erste Herausforderung sitzt er gerade am Update der Webseiten.....

Doch auch jenseits dieser strategischen Weichenstellungen gehen kontinuierlich neue KONWIHR- Projekte an den Start. So konnten im Rahmen der letzten Antragsrunde (September 2017) folgende 5 Projekte einen positiven Förderbescheid erhalten:

- Dynamics of Complex Fluids (Prof. Dr. Jens Harting, FAU)
- Fraktale Monsterkurven (Prof. Dr. Jörg Arndt, TH Nürnberg)
- Computing continuum fields from particle data (Prof. Thorsten Pöschel, FAU)
- Parallelization and Optimization of Numerical Methods in the Polaris(MD) Molecular Dynamics Software (Prof. Dr. Martin Zacharias, TUM)
- High-performance implementation of the Geometric Electromagnetic Particle-In-Cell Framework (Prof. Dr. Eric Sonnendruecker, TUM)

KOWNIHR lebt also noch, und zur aktuellen Förderrunde sind bereits auch etliche Anträge eingegangen. Daher: Stay tuned – zum nächsten Quartl wird's auch eine neue spannende KONWIHR Webseite geben.....

Gerhard Wellein

SPPEXA News

SPPEXA is now a Gold Sponsor of Science Node. You are asking yourselves – what is Science Node? Science Node is a platform of international connection and a redistribution point – a point of communication, intersection, and convergence.



The mission of Science Node is to inform the global research community about research breakthroughs, the resources that make such breakthroughs possible, and new advances that will make possible the breakthroughs of tomorrow. Our motive of using Science Node is to enhance our public relations distribution, especially in the US – as the Quartl audience already knows SPPEXA quite well.

The sponsorship allows us to publish 8 titles about SPPEXA over two years. We have already published a first general article introducing SPPEXA as a whole, see <https://sciencenode.org/feature/the-race-to-exascale.php>. The next articles will focus on concrete projects. The articles will be interview-based with the Science Node authors writing about the projects. If any SPPEXA project is keen to feature the next article, please approach us.

Benjamin Uekermann

*** Notiz * Notiz * Notiz ***

150 Jahre Technische Universität München Informationen und Hintergründe dazu gibt es unter <https://www.150.tum.de/>.



Im Rahmen dieser Feierlichkeiten wird es ein umfangreiches Programm geben - hier nur eine kleine Auswahl:

- 12.04.2018 Festveranstaltung TUM 150 Jahre im Herkulesaal der Residenz
- 03.05.2018 Mitarbeiterfest in Garching
- 04.05.2018 GALILEO:
Eröffnung der neuen Lebensmitte des Campus Garching

Diese und noch viele weitere Veranstaltungen sind unter https://www.150.tum.de/file/2018/02/tum_programm-2018-a5-web-1.pdf zu finden. Ein Blick in die Broschüre lohnt sich also!

Sparse Grids and Applications 2018

Der Sparse Grids and Applications Workshop findet vom 23-27.Juli 2018 am Institut for Advanced Studies in München statt. Informationen und Hintergründe unter <https://www5.in.tum.de/SGA2018/>.

Quartl* - Impressum

Herausgeber:

Prof. Dr. A. Bode, Prof. Dr. H.-J. Bungartz, Prof. Dr. U. Rüdè

Redaktion:

S. Herrmann, B. Rùth, Dr. S. Zimmer

Technische Universität München, Fakultät für Informatik

Boltzmannstr. 3, 85748 Garching b. München

Tel./Fax: ++49-89-289 18611 / 18607

e-mail: herrmasa@in.tum.de, **www:** <http://www5.in.tum.de/quartl>

Redaktionsschluss für die nächste Ausgabe: **01.06.2018**

* **Quartel**: früheres bayerisches Flüssigkeitsmaß,

→ das **Quart**: $1/4$ Kanne = 0.27 l

(Brockhaus Enzyklopädie 1972)