A Medium-Scale Distributed System for Computer Science Research: Infrastructure for the Long Term

Henri Bal VU University, Amsterdam
Dick Epema Delft University of Technology
Cees de Laat University of Amsterdam
Rob van Nieuwpoort Netherlands eScience Center
John Romein ASTRON
Frank Seinstra Netherlands eScience Center
Cees Snoek University of Amsterdam
Harry Wijshoff University of Leiden

Abstract

The Distributed ASCI Supercomputer (DAS) is a Computer Science infrastructure designed by the Advanced School for Computing and Imaging (ASCI) for controlled experiments with parallel and distributed systems. We have set up five generations of DAS, each consisting of 4-6 clusters located at different Dutch universities and integrated into a single shared distributed system using an advanced wide-area network. DAS is unique in that it has been available for 18 years, but always was kept consistent with the current research agenda. Each generation was set up by a single organization and with one clear vision. The DAS systems therefore are homogeneous, consistent, and easy to maintain. The goal of this paper is to show the huge impact of such a persistent long-term investment in Computer Science infrastructure, including numerous major awards, top publications, over 100 Ph.D. theses, and highly fruitful collaborations with application domains.

Introduction

Physical research infrastructures are as important in Computer Science as in any other science that uses experiments. This is especially true for the area of parallel and distributed systems, where many researchers only have access to platforms that are designed and deployed for production work, such as supercomputers or clouds. Although such systems enable large-scale computational experiments, they are ill-suited for research that needs detailed control over the hardware or the systems software: controlled reproducible *distributed* experiments on multiple resources are difficult to perform with production systems.

Several well-known projects have tried to set up distributed infrastructures specifically for Computer Science research. Most of these projects focus on *large scale* experiments with thousands of nodes. One approach is to federate existing resources, as is done successfully in PlanetLab. However, since the resources are used by many applications at the same time, it is hard to do *reproducible* performance measurements with this setup. Another approach is to build a new large-scale dedicated distributed system. An outstanding example is the Grid'5000 system [1]. Unfortunately, with most large distributed infrastructures, the funding also is distributed over time and over different parties, resulting in incremental updates to different parts, at the cost of coherence in the long run.

The Dutch *Distributed ASCI Supercomputer* (DAS) project has taken a different strategy during the past 18 years. Instead of aiming at a large scale, it aims at *consistency with the current research agenda.* The research agendas have changed considerably over the past two decades, from cluster computing to wide-area metacomputing, grids, peer-to-peer networks, optical grids, clouds, green computing, and heterogeneous computing. With each shift in research focus, the infrastructure has to be re-aligned, which is difficult

DAS-1	**Wide-area computing (1997)**
	Features: Homogeneous hardware and software, dedicated ATM network
DAS-2	**Grid computing (2002)**
	Features: Globus middleware
DAS-3	**Optical Grids (2006)**
	Features: Photonically switched 10 Gb/s links between all sites
DAS-4	**Clouds, diversity, green IT (2010)**
	Features: Hardware virtualization, accelerators, energy measurements
DAS-5	**Harnessing diversity, data-explosion (2015)**
	Features: Wide variety of accelerators, larger memories and disks, Software Defined Networking

Table 1: Summary of the DAS research agenda and characteristic features for each generation

with either of the approaches described above. The strategy of DAS has always been to build relatively small-scale distributed systems that are replaced regularly to fit with the changing needs of researchers.

We have implemented our strategy by building five successive generations of a moderate-size distributed system, each with a clear vision and research agenda. The hardware for each generation was built from scratch, with a budget around 1.5M Euro, using grants from a highly competitive program of the Netherlands Science Foundation (NWO) with additional matching from the participants. Each DAS system consisted of 4-6 clusters (with an order of 200 compute nodes in total) located at the participating universities and integrated into a distributed system using a wide-area network.

The five generations have all used the same two principles. First, DAS is set up by a *single organization*, the ASCI research school (Advanced School for Computing and Imaging). ASCI is a formalized and accredited entity set up in 1995 by Dutch universities to stimulate research collaborations. ASCI uses a steering committee (the authors of this paper) to develop the vision based on the current research agenda, to write the grant proposal, and to set up the system in line with this vision.

Because of this central organization, each DAS system is set up with one clear vision, rather than just combining different resources in ad hoc ways. Especially, all systems have been designed to be as *homogeneous* as desirable for the current research. For example, they all run the same operating system and other systems software. This approach greatly increases the coherence of the whole system and drastically simplifies whole systems maintenance, requiring only 0.5 FTE support.

A second principle is that DAS is designed for *Computer Science* research, including interactive distributed experiments. DAS users have access to the *entire distributed system* and can allocate multiple clusters at the same time. The usage policies aim to optimize the system availability for such (interactive) experiments and *not* to maximize system utilization, as with production systems.

The five generations differ because they address rapidly evolving research agendas. Table 1 shows the generations together with their main research focus and characteristic features. None of these research themes is unique and many other infrastructures exist for each of them. What is unique, however, is the way Dutch scientists have managed to *persistently* build such coherent infrastructures for almost two decades. This has resulted in an almost permanent basis for experimental Computer Science research, always offering the same type of instrument yet tailored to the needs of current research. Numerous changes were made over the years, but each change was always adopted system-wide to maintain coherence. Examples include changing the operating system (during DAS-1), using grid middleware (Globus, for DAS-2), using a dedicated optical interconnect between the sites (starting with DAS-3), and the deployment of accelerators (DAS-4 and DAS-5).

This centrally-coordinated change has enabled us to do long-term research in many strategically important areas. In retrospect, this unique combination of persistence and coherence would have been hard to achieve if we had tried to build large-scale systems. With DAS, all clusters of each generation are replaced simultaneously at the end of their lifetime (after 4 or 5 years). Large-scale distributed systems are almost always replaced gradually and asynchronously, based on available funding for the different parts. Despite their moderate size, the DAS systems have been remarkably successful even in international competitions where scale matters. We have won the IEEE CCGrid SCALE challenge three times and the NIST TRECVID video retrieval competition five times. Also, the systems always had a user community

	Name	Description	Impact
DAS-1	LFC	User-level network protocol	400+ citations
	Albatross	Wide-area algorithms	Foundation for many projects below
	MagPIE	Wide-area collectives	500+ citations, influenced MPICH-G2
DAS-2	Awari	Solving the Awari game	Pickover's book (250 milestones in mathematics)
	Satin, Ibis	Distributed programming system	700+ cites, SCALE'2008, Euro-Par'14 award
	JavaGAT	Grid programming toolkit	$60,000+$ downloads; led to OGF SAGA standard
	KOALA	Co-allocating scheduler	CCGrid'2012 keynote
	RTPl	Wide area TCP protocol	Internet2 land speed record
	Tribler, Cyclon	Peer-to-peer protocols	3000+ cites, Best paper award P2P 2006
DAS-3	StarPlane	Reconfigurable optical network	Keynotes, architectural principles current networks
	VL-e	Virtual Lab for eScience	20 M Euro funding
	Robot dog	Object recognition on a grid	AAAI'07 Most Visionary Research Award
	INDL	Resource description framework	Used for GENI and FED4FIRE
DAS-4	Glasswing	MapReduce on many-cores	SC'2014 (nominated best student paper)
	Squirrel	Quick launching of VM images	SC'2013 + HPDC'2014 papers
	COMMIT/	Modelling GPU data transfers	CCGrid'2014 (nominated best paper)
	WebPIE	Distributed reasoning	500+ cites, SCALE'2010 award
	BTWorld	Large-scale time-based datasets	SCALE'2014 award
Applications	Multimedia	Video and image retrieval	3500+ cites, awards (5x TRECVID), best papers
	Astronomy	Radio signals; astrophysics	Nomination Gordon Bell Prize 2014
	Climate	Modeling sea level rise	Enlighten Your Research Global Award SC'2013

Table 2: Sample projects done with DAS and discussed in the paper

of 100-150 scientists, resulting in over 100 Ph.D. theses, numerous awards, and top publications.

The main goal of this paper is to show the huge impact of such a persistent investment in Computer Science infrastructure. We will do so by describing a selection of projects that have used DAS and that resulted in many top publications and awards, as summarized in Table 2. We organize the descriptions along the first four generations of DAS (DAS-5 was set up in May 2015). Table 3 contains an overview of these systems. After that, we discuss how (and why) we collaborate with many application domains. (The paper is based on a keynote lecture of the first author at Euro-Par'2014, on the occasion of the Euro-Par Achievement Award.)

DAS-1: Wide-area computing (1997)

The original idea of DAS by Andy Tanenbaum in 1997 was to design a completely homogeneous distributed system. DAS-1 consisted of four cluster computers located at four universities (Amsterdam, VU, Leiden, Delft) connected by a dedicated 6 Mb/s ATM network. All nodes used the same (Intel PentiumPro) CPUs, local network (Myrinet), and Operating System (initially BSDI Unix, later RedHat Linux). The only difference in configuration was the size of the VU cluster: due to extra investments by the VU in all DAS generations, we set up larger clusters at the VU, allowing clean comparisons between distributed and single-cluster algorithms.

An important advantage of DAS over production systems is that users can experiment with low-level systems software. A good example is our research on user-level network protocols, which directly access the Network Interface from user space, thus avoiding the operating system from the critical communication path. We implemented a new network protocol for Myrinet called LFC (Link-level Flow Control) [2], consisting of a user-level library and new firmware for the Network Interface Card. LFC showed that programmable network interfaces increase flexibility and reduce communication overhead.

DAS is used extensively for investigating how multiple geographically distributed resources can be combined to solve very computationally intensive problems. This *distributed supercomputing* research started by studying simple algorithms on DAS-1 and evolved over the years into real-world applications

	DAS-1	DAS-2	DAS-3	DAS-4	DAS-5
Year	1997	2002	2006	2010	2015
Clusters	4	5	5	6	6
Cores	200	400	792	1600	3252
CPU	200 MHz Pentium Pro	Dual 1 GHz Pentium 3	Dual 2.2-2.6 GHz AMD Opteron	Dual quad-core Xeon E5620	Dual eight-core Xeon E5-2630v3
Interconnect	Myrinet	Myrinet	Myrinet-10G	QDR Infiniband	FDR Infiniband
WAN	ATM	Internet	Light paths	Light paths	Light paths

Table 3: Overview of the DAS systems, including the CPUs, local interconnect, and wide-area network.

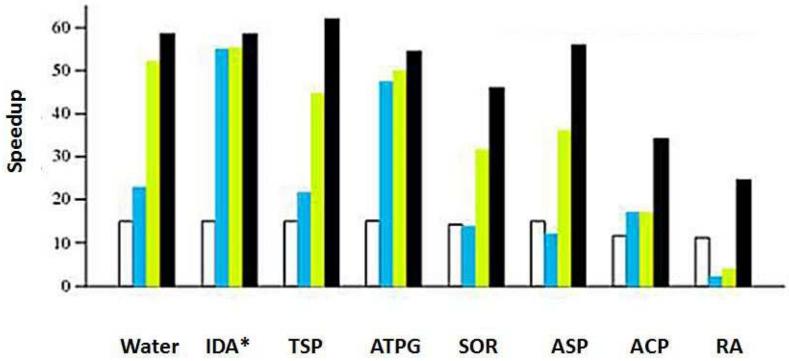

Figure 1: Performance of several parallel algorithms on 1 and 4 clusters. The leftmost and rightmost bars of each algorithm show the speedup (compared to 1 CPU) on a single 15-node resp. 60-node DAS cluster. The second bar gives the speedup of the original program spread over four 15-node clusters. The third bar gives the performance of a *wide-area optimized* program.

like climate modelling and astrophysics on DAS-4. To illustrate the early algorithmic work, Figure 1 shows the performance of several simple parallel algorithms on DAS-1. Some programs even run slower on four clusters with 15 nodes each than on one cluster with 15 nodes, because part of the communication now goes over wide-area links which are orders of magnitude slower than the local network. Most algorithms, however, can be optimized for wide-area systems, for example by doing latency-hiding, load balancing, message aggregation, etc., to reduce communication overhead of the wide-area links. These programs generally do run much faster on multiple clusters and often get performance close to that of a single large cluster with the same total number of nodes. Some fine-grained programs like Retrograde Analysis (RA) remain inefficient on wide-area systems, as expected.

The bandwidths of the wide-area networks have increased enormously over the generations, from 6 Mb/s for DAS-1 to 10 Gb/s and more for later systems. To allow early experiments with different wide-area speeds, we installed 8 local ATM links in the DAS-1 VU cluster, using the same hardware as in the real system. The latency and bandwidth of the ATM links were varied by delay loops. Except for the local ATM links, the experimentation system was identical to the real wide-area system, using the same binaries. This allowed us to analyze the sensitivity of parallel programs to wide-area latency and bandwidth [3].

These research projects are all typical for DAS: they need clean, laboratory-like experiments that are hard to perform on production systems, because they require co-allocation of identical resources or changes to the hardware or systems software. This early research showed that distributed supercomputing is feasible for a much wider class of applications than commonly believed, provided some (often simple) wide-area optimizations are applied. These insights are now applied in practice (see the Applications section). Many fundamental algorithmic insights obtained on DAS were used to design new programming systems for distributed supercomputing. MagPIe [4] is an implementation of MPI whose collective operations are optimized for hierarchical wide-area systems. Ideas from MagPIe have later been applied in MPICH-G2.

DAS-2: Grid computing (2002)

DAS-2 was based on the same principles as DAS-1, but it used the Globus middleware to enable grid experiments. Also, it used the normal (shared) university network infrastructure with 1 Gb/s Ethernet uplinks, instead of ATM to connect the sites. DAS-2 was used to study several grid programming environments and to conduct novel networking research.

The first major result obtained on DAS-2 was solving the game of Awari in 2003 (published in IEEE Computer). We computed the best possible move in all 889 Billion possible board configurations. This work was selected in Pickover's math book as one of 250 milestones in the history of mathematics.

Ibis [5] is a Java-centric programming system for high-performance applications on heterogeneous distributed systems. Its core communication layer is designed for dynamically changing systems. Ibis uses the JavaGAT (a predecessor of the SAGA standard of the Open Grid Forum) to transparently access different types of resources running a variety of grid middleware. Ibis also contains a component (SmartSockets) to solve connectivity issues due to firewalls. Satin is a Java-based programming system implemented with Ibis that can automatically run divide-and-conquer applications on distributed systems such as grids. Satin provides automatic cluster-aware load balancing, malleability, and fault-tolerance. Later (on DAS-4) we combined Satin with a new programming environment for many-cores, resulting in a system called Cashmere that runs on clusters containing different types of many-cores.

All performance experiments on these programming systems started on DAS, which gave a controlled environment. Once the issues were well understood, we moved to more realistic platforms by combining DAS with systems such as the GridLab testbed and Grid'5000, showing that our software also runs "outside the laboratory".

Another long-term research project is the KOALA multicluster scheduler. From the beginning, its key feature was the support for processor co-allocation for parallel applications, in terms of both mechanisms for interfacing with the local cluster schedulers and of scheduling policies. When co-allocating a parallel application, it can be either the user or the scheduler who decides how to split it up into *components* that are each scheduled on a single site. Later we have extended KOALA with support for the co-allocation of more application types, most notably Bags-of-Tasks and Workflows. Our current focus is on scheduling complete application frameworks, such as MapReduce.

In addition, DAS-2 was used extensively for networking research. DAS has always used both special testbed connectivity and the normal production network of the Dutch National Research and Education Network SURFnet. The developments in networking over the lifetime of DAS have been enormous. Interestingly, typical data transport protocols did not scale to these new high speeds. Many Internet protocols were defined in the 1980s when both computer memory and wide area bandwidth were extremely scarce. DAS-2 was used to experiment with different high speed transport protocols and to understand and optimize their throughput. We wrote a protocol test suite (RTPl) that enabled experimentation with controlled mixes of many normal low-bandwidth Internet flows and a few high throughput flows using extremely optimized new protocols, which may destructively influence each other. With this research we won the Internet2 Land Speed Records in 2002 and 2004.

Finally, DAS-started a long-term research line on peer-to-peer systems. Examples include a gossip-based peer sampling service [6], the Cyclon membership management protocol, and the Tribler social peer-to-peer system. Before we designed our BitTorrent-based Tribler P2P client [7] we performed extensive measurements of the world-wide BitTorrent P2P system to identify the main design challenges of P2P systems of decentralization, availability, and providing incentives. Performing these measurements required many IP addresses for contacting large numbers of peers. When designing and implementing Tribler, we emulated large numbers of Tribler peers on many DAS nodes to test new components under controlled but realistic circumstances.

DAS-3: Optical Grids (2006)

One of the big paradigm shifts of DAS-3 was the introduction of hybrid networking [8] where routed internet was augmented with optical photonic connectivity. Photonic devices allow dynamic color routing on a dark fiber infrastructure using wavelength selective switches. DAS-3 was one of the first systems that used such flexible infrastructure (called StarPlane), including several 10 Gb/s light paths between the sites provided by SURFnet. Also, we used a 10 Gbit/s optical path from Amsterdam to Paris to interconnect DAS-3 and Grid'5000.

StarPlane allowed us to study more data-intensive applications. For example, we implemented the DiVinE model checker on DAS-3, allowing it to use the memory of all clusters together, which is important since model checking requires a large memory due to state space explosion. The key optimization here was to do massive data-aggregation, i.e., combining many small messages (state exchanges) into large asynchronous data transfers. For this experiment, we had to combine multiple 10 Gb/s links to obtain the required wide-area bandwidth.

The hybrid networking and emerging heterogeneous computing infrastructures triggered yet another research problem: the need to have an integrated information system that is able to find the diverse resources. Typical current solutions work on one technology (e.g., BGP) but not across internet layers and domains and do not cover the data processing infrastructure. We proposed a semantic web based resource description framework (INDL) to define the topology and find resources of distributed infrastructures. This approach became successful and is now selected to handle similar functions in the USA NSF testbed GENI and European Future Internet FED4FIRE infrastructures.

One of the largest projects using DAS-3 was VL-e (Virtual Laboratory for eScience), which created an eScience environment and performed research on workflow, distributed programming, resource management, and other methodologies. VL-e obtained 20 M Euro funding from the Dutch government. About one third of the project consisted of researchers from ASCI, who exploited software like Ibis, JavaGAT, KOALA, INDL, VLAM-G (the VL-e workflow system) and others to create a Rapid Prototyping environment for eScience.

From DAS-3 on, the imaging researchers of the ASCI school also started to use DAS intensively. One of the most visible results was an application that could analyze the movies captured by the webcam in a Sony robot dog, using a world-wide grid. The initial implementation used TCP to distribute the video frames to clusters all around the world. Each cluster repeatedly processed a frame using a data-parallel MPI program. DAS-3 was part of this global grid and was also used for performance testing. The application won the 'most visionary research award' at AAAI 2007. The application was later redesigned using Java and Ibis, making it platform and middleware independent, fault-tolerant, and scalable. The Ibis version won the 2008 SCALE Challenge.

DAS-4: Clouds, diversity, green IT (2010)

DAS-4 was designed as a testbed for cloud computing, accelerators, and green IT. Its core (CPUs, LAN, OS) was still homogeneous, but various types of accelerators were added to the different sites, allowing performance comparisons between different GPU types within otherwise identical compute nodes. Also, DAS-4 contained power-monitors and its nodes could be set up with Cloud middleware.

DAS-4 is used for several projects on heterogeneous computing. Glasswing is a novel MapReduce framework on top of OpenCL that efficiently uses resources of heterogeneous cluster environments. It combines coarse-grained and fine-grained parallelism and aggressively overlaps computation, communication, memory transfers, and disk accesses. It used DAS-4 for a performance comparison against Hadoop on different types of accelerators, showing large performance improvements.

An interesting research problem triggered by a Climate Modelling application is how to optimize the data transfers between the host CPU and the GPU. The given application has many small kernels that use many transfers, and for each transfer a choice must be made whether to use explicit copying, memory mapping, or CUDA streams. In the COMMIT/ project, we developed a performance model that aids to make this decision without trying all possible combinations.

DAS-4 also was useful to study a variety of problems in Cloud computing and green IT. For example, Squirrel aims to efficiently start a large number of Virtual Machine images, which is a bottleneck when using Clouds for High Performance Computing. We also studied what it would take to virtualize networks and make them objects that compilers and advanced applications can program. This Internet Factories research recently resulted in a research collaboration with KLM and COMMIT/.

The GreenClouds project aims to decrease the energy consumption of HPC systems. For example, the project studied the energy profiles of Virtual Machines and it produced an energy budget calculator and resource manager that support Energy Efficient Ethernet (802.3az).

Another interesting line of research on DAS-4 is distributed reasoning. WebPIE [9] is a system (implemented on DAS with Ibis and Hadoop) that can do web-scale reasoning by computing the so-called materialization (closure) of huge RDF graphs. WebPIE won the SCALE challenge at CCGrid'2010 for solving a problem with 100 billion triples.

In 2014, the DAS consortium won the SCALE Challenge for the third time, now for a project (BT-World) that performed large-scale analysis of time-based datasets, in particular monitoring data collected from BitTorrent servers over a period of four years.

DAS: a stepping stone for applications

As even the performance of all clusters combined comes nowhere near that of top-500 supercomputers, DAS may seem totally unattractive for applications research. In fact, we have an explicit policy that DAS cannot be used for production runs during daytime. Therefore it is surprising that we have extremely good collaborations with many application areas, and each of the last three generations of DAS attracted participation and co-funding from a new domain (multimedia for DAS-3, astronomy for DAS-4, and eScience for DAS-5). The key insight is that DAS is attractive for applications as *stepping stone* for their research. Numerous applications first used DAS for prototyping and then migrated to production supercomputers. DAS has several advantages for applications:

- It allows easy and fast experiments with parallel algorithms, without complicated procedures or grant proposals. Domains like multimedia, astronomy, bioinformatics, semantic web, and distributed model checking have used DAS to develop and evaluate algorithmic alternatives, which were later ported to production platforms. As an example from astrophysics, the computational characteristics of the multi-physics simulations published in Nature [10] could not have been analyzed without the availability of DAS.

- DAS allows distributed experiments. For example, Ibis was used to efficiently run the Parallel Ocean Program on multiple clusters by optimizing its load balancing for wide-area distributed systems, similar to the earlier work on DAS-1. Climate modellers are now using this code on European production systems. This research won an Enlighten Your Research Global award at SC'2013, enabling further experiments using several top-10 supercomputers in the US, UK, and Germany connected by dedicated light paths.

- DAS allows experiments with modern hardware not yet available on production platforms. For example, the climate modellers and the astrophysicists first used the accelerators in DAS-4 to study the feasibility of this approach. Driven by these results, a Dutch supercomputer was extended with accelerators for production runs.

For Computer Science, these collaborations give real-world validations of new techniques and they can trigger new ideas for research. Also, some applications are from within Computer Science itself, such as multimedia, model checking, and semantic web. We thus support and encourage collaboration with application research on DAS, as long as it does not use DAS for production runs. Below, we discuss two examples in more detail, one from Computer Science and one from Astronomy.

Video and image retrieval

A challenge for researchers in video and image retrieval was the lack of programming tools for non-experts. We used DAS to develop user-transparent programming models that hide the difficulties of parallel implementation from their users, while supporting easy-to-use grid execution [11]. DAS enabled a successful line of research in image and video retrieval [12]. Award winning examples include the AAAI'2007 award and the detection of supernovae candidates in telescope image data (the DACH 2008 challenge).

The key challenge in visual retrieval is to understand what is happening where in the image by looking at pixels only. The standard approach involves processing large amounts of imagery to extract multiple color, shape, texture and motion features and to convert these to semantic labels, like “sheepdog”, “bowling alley”, and “teenager”, with the aid of supervised machine learning and a multitude of labeled examples. Examples of video and image retrieval innovations are new representations for video and images using color invariants, smart feature pooling, GPU-specific kernel computation, and harvesting training examples from the social-tagged web. Recently object localization in images, deep learning, and video event recognition have been added to the repertoire.

Due to the homogeneous DAS infrastructure and several algorithmic innovations, we have won each of the leading benchmarks in video and image retrieval one or more times in the past ten years. We have won 5 awards in the TRECVID video competition (for concept detection and interactive retrieval) and in competitions like the ImageCLEF photo annotation task (2009), the ImageNet classification with localization (2011), and the Pascal VOC classification with localization (2012).

Astronomy

DAS is a useful test bed to evaluate different architectures and algorithms for astronomy applications, in particular for analyzing radio astronomy signals. A good example is an extensive study of the impact of auto-tuning for an astronomy program (the dedispersion kernel, used for example in pulsar finding), using different accelerators (NVIDIA and AMD GPUs and the Intel Xeon Phi) and datasets from different telescopes. This research required a huge number of short runs to auto-tune several OpenCL parameters (like the optimal number of work-items or registers) for many different scenarios. The ASTRON cluster was extensively used to prototype applications that exploit new accelerator technologies. These applications (e.g., two GPU correlators, a beam former, an imager, and a pulsar pipeline) are now used in production by the largest radio telescope in the world (LOFAR).

DAS is also used for astrophysics simulations, in particular for the AMUSE (Distributed High-Performance Astrophysical Multipurpose Software Environment) project. A separate NWO grant was used to add a GPU cluster to the Leiden site in 2010. This work ultimately led to the experiment to simulate the Milky Way Galaxy on 18600 GPUs of the US ORNL Titan, which was nominated for the Gordon Bell prize in 2014.

Discussion

The Distributed ASCI Supercomputer has had a major impact on Dutch Computer Science for almost two decades. Whereas large-scale systems often have problems to guarantee long-term funding, DAS has always taken a different strategy and aims at *coherence* in combination with more modest system sizes. As a result, the interest in DAS has grown continuously over all generations. Each generation has also attracted new communities and co-funding partners.

Each DAS design has taken advanced wide-area networks into account by working together with the innovation program of SURFnet, which funded the wide-area networks of DAS. By studying distributed systems and wide-area networking hand-in-hand, several innovative results were obtained. Current developments on DAS-5 networking include 100 Gbit/s wide area connectivity and the introduction of Software Defined Networking (SDN), allowing a new layer of network virtualization.

The effectiveness of the investments in DAS is extremely high. Each system had a hardware budget of about 1.5M Euro. Without such a centralized investment, different groups would probably already spend such an amount on (fragmented) systems management for maintaining numerous local facilities and enabling ad-hoc sharing of these resources. The homogeneous design of DAS drastically simplifies systems maintenance.

Our overall approach and the limited budget also forced us to make compromises. The main limitation of DAS compared to Grid’5000 is that we do not provide user nodes with bare hardware on which they can run entire system software stacks, because it would complicate the design and system management. Instead, we provide nodes with preconfigured stacks, which can be modified if needed. In retrospect, a few projects therefore could not be done on DAS, in particular on new operating system designs and on low-level intrusion detection mechanisms. Another limitation is that DAS does not support large scale experiments, so we also use other, typically international, systems when needed.

The most important lesson learned from DAS is that both the central organization (ASCI) and the coherence of the system have been key factors in this success and in obtaining long-term funding from highly competitive programs. Despite their relatively modest size the DAS systems resulted in numerous awards and top-publications, over a hundred Ph.D. theses and highly fruitful collaborations with application domains.

Looking ahead, large-scale computing infrastructures like datacenters and clouds will continue to evolve and to raise research questions for a long time to come. To address these questions, infrastructures for computer science research such as DAS with successive new generations to meet the current research agenda will remain very important. For the foreseeable future of DAS, we can identify at least the three trends of heterogeneity, virtualization, and big data. First, heterogeneous computing on different types of accelerators is becoming increasingly important for supercomputing. Although DAS was initially designed as a completely homogeneous system, the central organization does allow us to add as much heterogeneity as desired for the research agenda, as currently demonstrated by DAS-5. Secondly, all components (computing, storage, networks, visualization, algorithms and libraries) of distributed systems are rapidly becoming virtualized, enabling easy grouping of these components into services from low bare metal to high level platforms. The algorithms for data processing and simulations need to evolve to ensure scaling, fault tolerance, and energy efficiency. A good example is the collection of data from Internet of Things into massive virtualized data stores. The virtualization and integration of wireless networks into these systems and platforms is the next challenge. Big data is yet another important trend. It requires both new information processing technologies (e.g., machine learning and semantic analysis) to address semantical issues and new high-performance technologies to handle large-volume and dynamic streams of data.

Acknowledgements

This paper was written by the researchers who designed and coordinated the most recent DAS systems. In addition, there are hundreds of people who have contributed to the DAS project by doing novel research on its infrastructure. We acknowledge in particular the ASCI board members who took the initiative for the first DAS system: Andy Tanenbaum, Bob Hertzberger, and Henk Sips. Kees Verstoep has coordinated the systems management of all DAS generations. We thank NWO, SURFnet, VL-e, MultimediaN, the Netherlands eScience Center, and the COMMIT/ public-private research community for their funding. We commemorate the late Lex Wolters, who has been steering committee member for several DAS systems.

References

[1] R. Bolze et al. Grid'5000: A large scale and highly reconfigurable experimental grid testbed. *International Journal of High Performance Computing Applications*, 20(4):481–494, 2006.

[2] R. A. F. Bhoedjang, T. Rühl, and H. E. Bal. User-Level Network Interface Protocols. *IEEE Computer*, 31(11):53–60, November 1998.

[3] A. Plaat, H.E. Bal, and R.F.H. Hofman. Sensitivity of Parallel Applications to Large Differences in Bandwidth and Latency in Two-Layer Interconnects. In *Proceedings of High Performance Computer Architecture (HPCA-5) (also published in Future Generation Computer Systems, 2001).*

[4] T. Kielmann, R.F.H. Hofman, H.E. Bal, A. Plaat, and R.A.F. Bhoedjang. MAGPIE: MPI's Collective Communication Operations for Clustered Wide Area Systems. In *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 131–140, Atlanta, GA, May 1999.

[5] R.V. van Nieuwpoort, J. Maassen, G. Wrzesinska, R.F.H. Hofman, C. Jacobs, T. Kielmann, and H.E. Bal. Ibis: a flexible and efficient Java based grid programming environment. *Concurrency and Computation: Practice and Experience*, 17(7-8):1079–1107, June 2005.

[6] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M.R. van Steen. Gossip-based peer sampling. *ACM Transactions on Computer Systems*, 25(3), August 2007.

[7] J.A Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D.H.J. Epema, M. Reinders, M.R. van Steen, and H.J Sips. Tribler: a social-based peer-to-peer systems. *Concurrency and Computation: Practice and Experience*, 20(2):127–138, 2008.

[8] T. DeFanti, C. de Laat, J. Mambretti, K. Neggers, and B. St. Arnaud. Translight: a global-scale lambdagrid for e-science. *Communications of the ACM*, 46:34–41, November 2003.

[9] J. Urbani, S. Kotoulas, E. Oren, and F. Van Harmelen. Scalable distributed reasoning using mapreduce. *The Semantic Web-ISWC 2009*, pages 634–649, 2009.

[10] S.F. Portegies Zwart and E.P.J. van den Heuvel. A Runaway Collision in a Young Star Cluster as the Origin of the Brightest Supernova. *Nature*, 450(7168):388–389, November 2007.

[11] F.J. Seinstra, J.-M. Geusebroek, D. Koelma, C.G.M. Snoek, M. Worring, and A.W.M. Smeulders. High-performance distributed image and video content analysis with parallel-horus. *IEEE Multimedia*, 14(4):64–75, 2007.

[12] K.E.A. van de Sande, T. Gevers, and C.G.M. Snoek. Evaluating Color Descriptors for Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

Henri Bal *is a full professor at VU University in Amsterdam. His research interests include programming environments for parallel, distributed, and smartphone systems. He received a PhD in Computer Science at VU University. He is a member of IEEE, ACM, and the Academia Europeana. Contact him at bal@cs.vu.nl.*

Dick Epema *is an associate professor at Delft University of Technology and a part-time full processor at Eindhoven University of Technology. His research interests include scheduling in large-scale distributed systems, and online social networks. He received a PhD in Mathematics at Leiden University. Contact him at d.h.j.epema@tudelft.nl.*

Cees de Laat *is a full professor at University of Amsterdam. His research interests include optical and switched networking, workflows, big data, e-infrastructures, and systems security & privacy. He is co-founder of the Global Lambda Integrated Facility and member of IEEE and ACM. Contact him at delaat@uva.nl.*

Rob van Nieuwpoort *is director eScience technology at the Netherlands eScience center. His research interests include efficient computing and scalable parallel programming models. He received a PhD in Computer Science at VU University. Contact him at R.vanNieuwpoort@esciencecenter.nl.*

John Romein *is senior researcher at ASTRON (Netherlands Institute for Radio Astronomy). He received his PhD degree in Computer Science at VU University. His primary research interest is accelerated computing. Contact him at romein@astron.nl.*

Frank Seinstra *is Director eScience Program at the Netherlands eScience Center, Amsterdam. His research interests include Advanced Distributed Cyberinfrastructure ("Jungle Computing") and its application in various research domains. He received a PhD in Computer Science at the University of Amsterdam. Contact him at f.seinstra@esciencecenter.nl.*

Cees Snoek *is an associate professor at the University of Amsterdam. His research interests focus on video and image retrieval. He received a PhD in Computer Science at the University of Amsterdam. He is a senior member of ACM and IEEE. Contact him at cgmsnoek@uva.nl.*

Harry Wijshoff *is a full professor at Leiden University. His research interests are optimizing compilers, irregular computations, and parallel algorithms. He received a PhD in Computer Science at Utrecht University. Contact him at h.a.g.wijshoff@liacs.leidenuniv.nl.*