DOI: 10.14529/jsfi240101

Grid Computing Evolution in Scientific Applications

Maria A. Grigorieva[1], ***Alexei A. Klimentov***[2]

The advent of interconnected machines laid the foundation for utilizing distributed computing resources. Coinciding with rapid advancements in computing technologies and significant hardware innovations at the turn of the century, the field of science also experienced exponential growth. As supercomputers remained limited in availability and usage, distributed computing leveraged the potential of idle and dedicated resources, bridging the gap between scientists and computationally intensive projects. This paper provides a review of the evolutionary journey of grid computing in scientific applications, starting from the advancements in network connection technologies and concept of metacomputing and progressing to the current developments integrating cloud technologies with large-scale grids. The paper aims to outline the key milestones, advancements, and challenges encountered throughout this evolution, highlighting the potential of grid computing in enabling scientific breakthroughs and addressing future research directions. The most popular middleware systems are considered, as well as a description of scientific grid systems that existed in the past and are still in operation today is given. At the end of the article, we examined two of the most significant scientific discoveries that became possible largely thanks to grid technologies.

Keywords: grid, distributed computing, history, science, metacomputing, grid middleware, scientific grid, cloud computing.

Introduction

The advancement of distributed computing technologies was primarily driven by scientific endeavors. Since the early 21th century, the technologies faced significant challenges posed by international scientific projects in data-intensive fields such as particle physics, astronomy, computational chemistry, bioinformatics, and Earth and climate sciences. Subsequently, these technologies evolved and met these challenges by connecting existing heterogeneous computing resources distributed globally into unified environments through high-throughtput networks and specialized middleware systems. This integration ensured secure, dependable, efficient, and transparent operations. As a result, massive amounts of data, reaching the scale of multipetabytes and even exabytes, became distributed and accessible to broad international communities. The advancement of big science propelled technological and software innovations, eventually leading to the creation of grid computing infrastructures.

This article aims to provide an overview of the key stages in the development of distributed computing systems within scientific domains. The paper is structured as follows:

- *Section 1* outlines the evolutionary journey of grid computing in scientific disciplines, highlighting significant milestones while also discussing advancements in network technologies.
- *Section 2* explores the historical background of interconnecting machines for computational purposes.
- *Section 3* gives an overview of metacomputing projects, emphasizing the stages of metacomputing and delving into the insights gained from these projects.
- Moving to *Section 4*, the focus is centers on crucial grid computing initiatives, including the I-WAY experiment, pioneering literature on grid computing, grid forums, and the shift towards Web 2.0 integration.

[1]Lomonosov Moscow State University, Moscow, Russian Federation
[2]Brookhaven National Laboratory, Upton, US

- *Section 5* offers insights into grid middleware systems like HTCondor, Globus Toolkit, NIMROD, ARC, UNICORE, BOINC, which became the foundation for the efficient functioning of distributed systems.
- *Section 6* elaborates on various significant and captivating scientific grids such as TeraGrid, GridPP, WLCG, IGWN, among others.
- Over time, technologies have evolved to such a level that they can dictate their requirements to science. Hence, *Section 7* delves into exploring the integration of large grid systems with cloud technologies, using the WLCG as a case study example.
- The concluding *Section 8* illuminates two groundbreaking scientific discoveries that have been enabled through the innovative applications of grid computing technologies.

The topic of grid computing has a very long history, and it is so multifaceted that it does not allow for all its aspects to be covered within the framework of one article. Our objective is to present a broad perspective on the evolution of the technology, its various developmental stages, and to illustrate, using past and present grid systems as examples, the significance of this technology in scientific endeavors. It has established itself as a dependable bedrock for research, facilitating the continuous success of numerous major projects and fostering optimism for forthcoming discoveries.

Table 1 shows a representative sample of the most significant events in the evolution of grid computing, highlighted in this paper.

1. Distributed Computing and Network Throughput

Since a grid refers to computers connected by networks, the evolution of grid technologies is tightly connected with the significant increase of the network throughput.

In the 1960s, networks were primarily used for communication between government agencies and universities. These early networks were slow, with limited capacity and functionality. Thus, Ethernet was first developed in the early 1970s and initially it operated at a speed of *2.94 Mbps (megabit per second)*. However, over the years, Ethernet has evolved significantly and has seen several speed upgrades. In 1980, Ethernet was upgraded to support *10 Mbps*, which became known as 10Base-T. This was followed by the introduction of Fast Ethernet in 1995, which supported speeds of up to *100 Mbps*, also known as 100Base-T. Gigabit Ethernet was introduced in 1998, which allowed for even faster data-transfer rates of up to *1 Gbps (Gigabit per second)*. This was followed by 10 Gigabit Ethernet in 2002, which allowed for speeds of up to 10 Gbps, and more recently, by *40 Gigabit* and *100 Gigabit Ethernet*. So, the speed of Ethernet has evolved significantly from 2.94 Mbps in the 1970s to multi-gigabit speeds today. And of course, network speed contributed a lot in the development of distributed technologies [66].

To gain a deeper understanding of the history of distributed computing, it is essential to differentiate between the two types of networks: LAN (Local Area Network) and WAN (Wide Area Network). The main difference between LAN and WAN is their geographic scope, with LAN being limited to a small area and WAN covering a much larger area. This also affects the transmission speed and cost, as WANs often require high-speed and expensive communication links to connect various locations. Before the 2000s, WAN speeds had lagged behind LAN. While LAN technologies evolved rapidly with advancements such as Ethernet, WANs had to catch up with the advancements in fiber optics, routing protocols, and networking equipment to improve data transfer speeds over long distances. However, as networking technology has evolved WAN data rates have increased [16].

Table 1. Timeline of Grid Computing Evolution in Scientific Applications

Years	Technology/Event	Description
Early 1970s	ARPANET, NCP, FTP, TCP/IP, Ethernet	Development of protocols and networking technology that laid the groundwork for a connected world, enabling seamless communication and information exchange among computers.
Late 1980s–early 1990s	Gigabit Testbed and Metacomputing Projects, Condor, Web	Testing of high-speed network technology, developing projects that utilize distributed HPC for solving complex problems through parallel coordination. Emergence of the high-throughput computing system that helped in managing and scheduling computational jobs across distributed resources. Web technologies have been employed to create interfaces for accessing grid resources.
Mid–1990s	NIMROD, Globus Toolkit, UNICORE, I-WAY, Web Metacomputing	Development of grid middleware systems and projects showcased the feasibility, scalability, and advantages of grid computing, shaping its evolution.
Late 1990s	Grid book, Grid Forums, Web 2.0 and Web Services	Literature about grid computing, organization that worked on developing grid technologies, standards, and best practices to advance grid computing globally. Emergence of Web services have greatly influenced grid computing evolution by improving the accessibility, interoperability, and scalability of distributed computing resources.
2000–2010	Condor-G, ARC, Globus Toolkit 2-4, gLite, BOINC, NIMROD-G, Globus Online	Development and upgrading of notable grid middleware systems: additional features for managing and scaling tasks in grid environments, gradual transition to service-oriented platforms.
2011–now	Clouds over Grids, HTCondor, Unicore 7-9, Globus Toolkit decommissioning	Integration of cloud computing technologies with grids to improve scalability, flexibility, and resource utilization. Upgrading major grid middleware systems to support grid and cloud environments, transition from the older framework to more modern and streamlined approaches to distributed computing.

In the 1960s WANs connecting different organizations or regions were rare and typically used leased lines or dial-up connections. The data transmission rates were very low (about *56 kbps*). In 1970–80s the use of network protocols such as TCP/IP and X.25 for WANs started to emerge. Then, in the 1980s, the wide-scale deployment of dial-up modems and Integrated

Services Digital Network (ISDN) for WANs began, which improved the network speed compared to leased lines [74]. In the 1990s, when LANs started to use Fast and Gigabit Ethernet, WAN technologies like Asynchronous Transfer Mode (ATM) and Frame Relay replaced X.25. For example, ATMs offered speeds of up to *622 Mbps*. In the beginning of 2000s WANs moved to MPLS (Multiprotocol Label Switching) and VPN technologies. Fiber optic cables, which have much higher bandwidth than copper cables, also became more commonly used. MPLS could support multiple protocols and traffic types and offered speeds of *10 Gbps* and beyond. Over time, with the advancement of technologies like fiber optics, MPLS, and improvements in networking equipment, WAN speeds have also significantly improved and caught up with LAN speeds, now offering multi-Gbps and even 10 Gbps+ connections, narrowing the gap between LANs and WANs in terms of bandwidth capabilities.

Table 2 shows the evolution of LANs and WANs throughput over time.

Table 2. Evolution of LAN and WAN Throughput

Years	LAN	WAN
1980s	1–10 Mbps	a few Kbps to a few Mbps
1990s	100 Mbps (Fast Ethernet)	1.544–2.048 Mbps
Late 1990s to 2000s	1 Gbps (Gigabit Ethernet)	multiple to tens Mbps
2010s	10–100 Gbps	100 Mbps – 1 Gbps
Present	400 Gigabit Ethernet	10 Gbps and beyond

We are discussing the evolution of network technologies at the beginning of this paper as they have made a significant contribution to the development of distributed computing. Essentially, the development of grid technologies in international mega-science projects is closely connected with the evolution of WAN connections.

2. Early Ages of Distributed Computing

The idea of leveraging unutilized storage and CPU resources emerged when computers were first linked by networks. Initially, there were two directions for the evolution of remote computing. The first grew from the necessity of data exchange between military departments of the USA, and the second – from the introduction of the first mainframe machines.

The predecessor to the modern Internet was the ARPANET [59]. It appeared in 1967 as a project of the United States Department of Defense's Advanced Research Projects Agency (DARPA). Its purpose was to create a network that could connect computers and share information between them, even in the event of a nuclear attack, which implies, above all, reliability and fault-tolerance in data transmission. The network throughput in the ARPANET was initially 50 Kbps. The first successful message sent on the Internet was in 1969, between two computers at the University of California, Los Angeles (UCLA) and the Stanford Research Institute (SRI). These machines were linked by a direct physical connection via a leased telephone line. Actually, by 1969, the first computers of different types were connected on a network, but it was not useful until the introduction of the NCP (Network Control Protocol) in 1970 – a primary protocol responsible for establishing connections, detecting errors, packetizing data, and routing data across the network[3]. It enabled the first two host-host protocols, remote login (Telnet) and File Transfer Protocol (FTP). The ARPANET was declared operational in 1971. NCP was replaced by the Transmission Control Protocol/Internet Protocol (TCP/IP) in 1972, which is still used today as the standard protocol for the Internet. These technologies became the basis for developments over the next 20 years.

Another direction of the evolution of remote computing was related to the mainframe computers. They were first introduced in the 1960s and they were expensive, required specialized skills to operate, making them inaccessible to most individuals and businesses. To overcome this challenge, a system of remote computing was used that allowed users to access these mainframe computers from remote locations. That was another usage of the NCP protocol that allowed a user on one machine to login to another device via the Telnet protocol and upload a file. FTP allowed users to connect to a mainframe computer through a network connection and transfer files back and forth between their local computer and the mainframe. FTP quickly became a popular tool for remote computing and file sharing, and it remains in use today as a standard protocol for transferring files over the Internet.

Until the early 1970s, the dial-up connections using modems allowed users to connect to remote computers over telephone lines. This was a slow and unreliable method of communication. The emergence of Ethernet, FTP and TCP/IP protocols in the early 1970s laid the foundation for the evolution of remote computing by enabling remote access to data and resources and facilitating communication between different computer systems over a network. At the same time researchers began to understand that distributed computing would be difficult. As any message may be lost, corrupted, or delayed, the robust algorithms must be used in order to build a coherent system.

2.1. Linked Machines for Computations

In the early 1980s there were attempts to develop production systems that were supposed to ensure consistency, availability, and performance in distributed systems. And besides data exchange, there were early attempts at utilizing linked machines for computations. For example, the HTCondor (initially called Condor) project was developed in 1984 by Professor M. Livny at the University of Wisconsin, USA. It was a software package to manage workloads in the distributed computing environment. It allowed large numbers of computing tasks to be distributed across multiple machines, so that they could be executed in parallel. This advancement enabled the efficient processing of vast quantities of data at a significantly accelerated pace compared to using a single machine [58].

The idea behind the utilization of linked machines for computations was to aggregate idle computing resources that would otherwise be unused and use them to perform computationally intensive tasks. Condor used a variety of protocols for data processing on multiple machines, including FTP for file transfer, TCP/IP for reliable communication between devices, and other protocols such as UDP (User Datagram Protocol) for faster data transmission.

2.2. Emergence of World Wide Web

In the 1980s, software systems were developed with limited consideration for their interoperability with other systems. However, as the number and diversity of computer systems

[3]https://www.rfc-editor.org/info/rfc33

increased, the necessity of data exchange between them became more apparent. These systems utilized varying data formats and communication protocols, posing challenges in data sharing and information exchange. Consequently, this lack of interoperability impeded the development of new applications that could seamlessly operate across different systems. Hence, the demand for a standardized approach to connect diverse software systems became evident, leading to the emergence of "middleware". Although the term "middleware" was not introduced until the mid-1990s, it denotes the need for a standardized solution to facilitate software system integration.

Things completely changed with the emergence of the World Wide Web (or Web)[4]. It is considered that Web was developed in 1989 at CERN by Tim Berners-Lee. However, actually, there was a whole scientific group involved in this development. The Web was described as a method for sharing information among geographically dispersed scientists: the CERN community at that time included more than 17000 scientists from over 100 countries. In Web infrastructure all hypertext documents were linked into an information system accessible from any node on the network [19].

The initial set of protocols and standards that were used to create the first Web system were limited, but still formed the foundation of the modern Web: HTML (Hypertext Markup Language), HTTP (Hypertext Transfer Protocol), URL (Uniform Resource Locator), TCP/IP, DNS (Domain Name System), FTP. The development of the Web created new demands for high-throughput computing that eventually revolutionized the way people access, share, and use information and resources.

3. Metacomputing

Another direction of the evolution of distributed computing in science was the idea of sharing supercomputing resources to address the growing demand for computing power in scientific and engineering fields. In the 1990s the scientific interests began to recognize that distributed supercomputing might achieve higher performance than individual supercomputers or clusters can provide [80]. So, various geographically distributed supercomputers can be connected by WANs in order to solve grand challenge problems in reasonable time. Such systems could be used, for example, to carry out combined climate and ocean modeling as demonstrated by using different types of the unique linked computing resources. This concept was called "metacomputing" and can be considered as the first prototype of grid computing. Basically, metacomputer can be defined as a network of heterogeneous, computational resources linked by software in such a way that they can be used as easily as a personal computer.

Supercomputers and high-performance computing (HPC) systems were often expensive, difficult to maintain, and not easily scalable to new applications. This made it difficult for individual researchers and small organizations to access the computing power they needed to conduct advanced studies and solve complex problems. Moreover, grand challenge problems run weeks and months even on supercomputers and clusters. To solve these issues, a wide variety of powerful batch execution systems such as LoadLeveler (descendant of Condor, 1994), LSF (Load Sharing Facility, 1992) [84], Maui (1999) [54], Portable Batch System (PBS, 1991) [51] were developed in the United States and spread throughout academia and business.

For instance, LoadLeveler was a system designed to effectively manage both serial and parallel jobs across a cluster of servers. This cluster, referred to as a LoadLeveler cluster, comprised

[4]https://www.home.cern/science/computing/birth-web/short-history-web

diverse machines or servers, such as desktop workstations utilized for batch jobs during idle periods, dedicated servers, and parallel machines [77].

The allocation of jobs to machines in the cluster was done by a scheduler. This allocation depended on the availability of resources within the cluster and various rules that could be set by the LoadLeveler administrator. When a user submitted a job using a job command file, the LoadLeveler scheduler tried to locate resources within the cluster to meet the job's requirements. Meanwhile, LoadLeveler's responsibility was to maximize the efficiency of the cluster. It achieved this by maximizing resource utilization while minimizing the turnaround time for jobs experienced by users.

LoadLeveler was used on several high-performance computing systems, including the IBM Blue Gene/L supercomputer, the Cray XT5h at Oak Ridge National Laboratory, and the IBM SP at the Max Planck Institute in Munich.

Despite significant progress in the development of batch execution systems, it can be assumed that the establishment of gigabit networks played a crucial role in the development of metacomputing at a global scale.

3.1. Gigabit Testbed Initiative

The Gigabit Testbed Initiative (USA, 1987–1995) [1] was a major effort by approximately forty organizations representing universities, telecommunication carriers, industry and national laboratories, and computer companies to create a set of very high-speed network testbeds and to explore their application to scientific research. This initiative started in 1990 and was funded by the National Science Foundation (NSF) and the Defense Advanced Research Projects Agency (DARPA).

At the time the Gigabit Testbed Project began, WAN speeds were in the 50 Kbps to 1.5 Mbps range and LAN aggregate speeds were typically 10 Mbps or less. So, we could argue that the creation of gigabit networks was a milestone which enabled the discovery of new possibilities for networked data transmission, and triggered the refinement of grid technologies.

"The coupling of computer network researchers, who have largely come from the field of computer science, with the carrier telecommunications community provided another important dimension of integration. The development of computer communications networks and carrier operated networks have historically proceeded along two separate paths with relatively little cross-fertilization. The testbeds allowed the two communities to work together, allowing each to better appreciate the problems and solutions of the other" (quote from The Gigabit Testbed Initiative Final Report, December 1996 [2]).

The five testbeds were geographically located around the US. Each testbed had a different set of research collaborators and a different overall research focus and objectives. At the same time, there were also common areas of research among the testbeds, allowing different solutions for a given problem to be explored. The Gigabit Testbed Initiative, by creating a new model for network research, has had a major impact on both education and industry.

3.2. Stages of Metacomputing Evolution

Metacomputing evolved in several stages: creating software to make users jobs executed on different machines easier (LAN); distribution of applications for seamless users interaction (LAN); and transition from LAN to WAN metacomputing systems.

3.2.1. Developing software to simplify the execution of users jobs on different machines

The first stage was in creating and harnessing the software to make the users jobs to be executed on different machines easier. *"For any one project, a typical user might use a desktop workstation, a remote supercomputer, a mainframe supporting the mass storage archive, and a specialized graphics computer. Some users have worked in this environment for the past decade, using adhoc, custom solutions, providing specific capabilities at best, in most cases moving data and porting applications by hand from machine to machine. The goal of building a metacomputer is elimination of the drudgery involved in carrying out a project on such a diverse collection of computer systems"* [78]. So, the first stage involved interconnecting the resources with high-performance networks, implementing a distributed file system, coordinating user access across the various computational elements, and making the environment seamless.

The metacomputer at the National Center for Supercomputing Applications (NCSA, USA) was one of the examples of the LAN metacomputer of the first stage. In 1987 NCSA embarked on a research project aimed at investigating various emerging computational technologies that held the potential to greatly influence the field of scientific computing. This initiative, named the Rivers (Research on Interactive Visual Environments) Project, focused on the development of hardware and software systems. These systems aim to transition high-end 3D visualization from a batch process to an interactive one, as well as enable visualization-based interactive steering of supercomputing simulations within a high-performance distributed environment [50].

Figure 1 demonstrates schematic architecture for the NCSA LAN Metacomputer. Upon

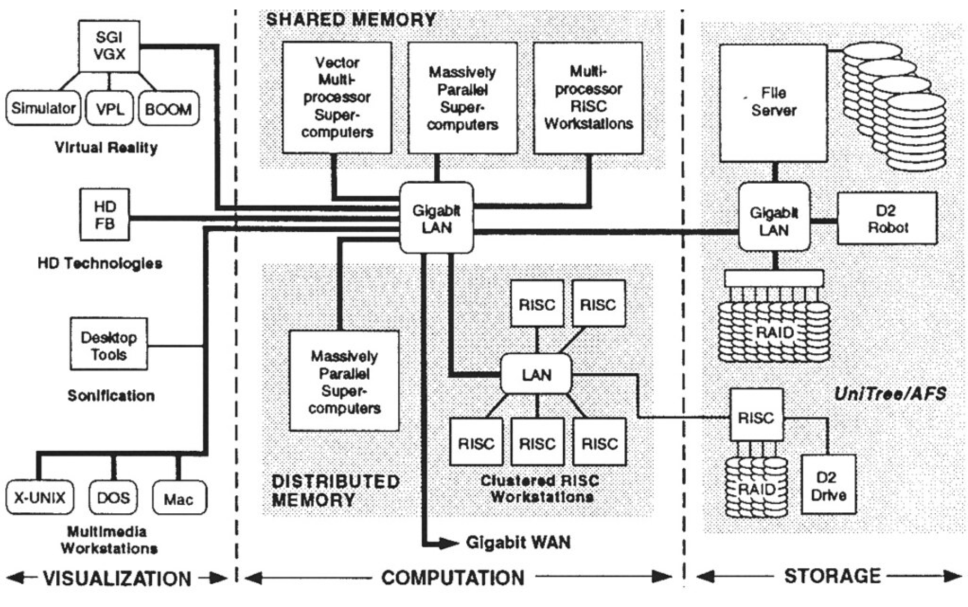

Figure 1. NCSA LAN Metacomputer [50]

closer examination, it is quite feasible to extend this diagram to modern grid and cloud computing setups, featuring distinct compartments for computation, storage, and a web interface for resource access. The pipelined structure of the visualization process required careful hardware system design. A balanced visualization system had to consider sustained bandwidth between distributed hosts, mass storage systems, host memory, and graphics accelerators. Shared memory architectures were also considered attractive for visualization. Powerful general-purpose processors were deemed necessary along with rendering engines due to computation-intensive calculations in the early stages of visualization. The distributed hardware system for visualization under development within Rivers and NCSA included dual, three-tier schemes for computing and network strategies. High-level machines were connected by a high-speed network (with a peak bandwidth of 100 Mbps), while a middle-tier network (50–80 Mbps) handled messages and conventional network traffic. Graphics accelerators were utilized for rendering operations. The Rivers group developed a three-dimensional, interactive animation tool capable of real-time animation of complex polygon data sets. It was utilized to visualize data from diverse fields such as solid mechanics, atmospheric sciences, and astrophysics.

3.2.2. Distribution of applications for seamless users interaction

During the following stage, efforts were made to distribute applications, enabling users to work seamlessly, in addition to pooling resources by networks. The special software that allowed this to be done in a general way, as opposed to some custom solutions, was emerging since that time. This software is commonly referred to as "middleware".

Several initiatives have resulted in valuable services for developers of metacomputing applications. To name a few:

- Legion (1992) – a software system that created system components based on a distributed object-oriented model [28]. It provided a high degree of flexibility in managing distributed resources;
- Message Passing Interface (MPI, 1994) – a standard protocol used for message-passing, particularly in distributed memory systems for parallel computing [31].

3.2.3. Transition to WAN metacomputing systems

The first two phases of the evolution of metacomputing predominately sought to harness LAN technology, given the apparent disparity in network capacity between LAN and WAN options. The capabilities of LAN-based metacomputers surpassed their WAN counterparts by roughly a year. The third stage of metacomputing entailed the establishment of a transparent national network, achieved through the implementation of WAN infrastructure and standardization of administrative, file systems, security, accounting, data transferring, and other levels to enable cooperation between multiple LANs.

For example, the High Performance Computing Center Stuttgart (HLRS) partnered with the Pittsburgh Supercomputing Center (PSC) and Sandia National Laboratories (SNL) were linked to establish a transatlantic wide area application test-bed in 1997 as part of the G7 initiative. The project aimed to couple HPC resources to create a powerful computing cluster with a theoretical peak performance of about 1 TFLOPS. The project utilized a high-speed network and a newly developed message-passing software, PACX-MPI[5], for distributing a single application across multiple MPPs[6]. This setup was enhanced by integrating the visualization software COVISE[7] for efficient data extraction directly from the application [73].

The project highlighted the need for faster networks for global work, the value of message-passing for distributed computing, challenges with standards in message-passing, limitations in

[5]PACX-MPI – extension of the Message Passing Interface (MPI) standard, which is widely used in distributed and parallel computing to allow communication between processes running on different computing nodes

[6]MPP – Massively Parallel Processing

[7]COVISE – visualization tool for analyzing scientific and engineering data in various fields such as computational fluid dynamics, structural mechanics, and geosciences

closely coupled applications like Navier-Stokes, and the lack of standards for resource management in metacomputing.

Another WAN metacomputing project aimed to create a global metacomputer by connecting supercomputers from Japan, the USA, Germany, and the UK, over a network spanning 10,000 miles. The optimization was achieved by using dedicated PVCs (Permanent Virtual Circuit) or improved routes. The project demonstrated the usability of this global metacomputer during Supercomputing 99 (SC99) conference in Portland with various demanding applications, showcasing the capability to link different types of machines like Cray-T3E and Hitachi SR8000, and successfully handle data-intensive tasks, such as processing output from experimental facilities [70].

3.3. Web Metacomputing

The rise of the World Wide Web brought a surge in Internet hosts. By the turn of the millennium, around 360 million computers were connected. This network of processors forms a potent parallel supercomputer, but many machines are underused, mainly used for basic tasks like email, file editing, and web browsing, resulting in idle time. Hence, leveraging this vast computing resource for cryptography, mathematics, and computational science problems is valuable.

To fill this gap, several alternative metacomputing projects had emerged to leverage the power of the Internet, one of which was Web metacomputing. This concept was related to the coupling of conventional machines. Web metacomputers were called distributed computing systems designed to run on top of the Web. These systems used web-based interfaces to allow users to access computing resources from different machines and locations.

Many of the issues that were addressed for LAN metacomputing systems, such as programmability, scheduling, and security, were considered in isolation and in well-controlled environments. But utilizing the Web as a parallel metacomputer introduced new challenges: there was no shared file system, no user who had accounts on all the potentially available machines, no common architecture, and the execution environment was dynamic.

Several projects were implemented to exploit the potential of the Web for metacomputing: ParaWeb (1995–1996) [23], Popcorn project (1996–1998) [69], Charlotte project (1998–2000) [14], SuperWeb project (1998–2000) [10].

For example, the SuperWeb project was a prototype of a distributed computing infrastructure that integrated hosts, brokers, and clients. Hosts registered a portion of their available computing resources (such as CPU time, memory, disk space, and bandwidth) with resource brokers. Client computations were subsequently assigned to registered resources by the broker. Additionally, an economic model was examined for trading computing resources within this project.

3.4. Lessons Learned

Metacomputing projects have provided valuable insights across networking, programming models, applications, and resource handling, shaping future directions for this field.

- Networking remains a crucial aspect, highlighting the need for faster global networks to support metacomputing on a larger scale efficiently. The limitations in bandwidth, especially for transatlantic or worldwide operations, underscore the importance of addressing network challenges for seamless metacomputing workflows.

- Dealing with different data representations, processor speeds, and communication speeds across various systems.
- Addressing load imbalance caused by differing processor speeds at the application level due to the lack of standardized ways to determine processor speed.
- Handling resource management and scheduling challenges, particularly in coordinating the availability of resources across multiple systems and networks.
- Recognizing the importance of a common file system for seamless access to data.
- Enhancing interoperability and setting message-passing standards are crucial for optimizing metacomputing applications.

Overall, these projects highlighted the complexities involved in integrating and coordinating resources in a metacomputing environment, emphasizing the need for standardized approaches and advanced resource management techniques.

By the beginning of 2000s, none of the Web metacomputing projects were widely adopted. One of the main reasons contributing to this issue was the lack of widespread standardization and interoperability between the different web metacomputing projects. Another crucial factor was the need for robust security mechanisms to ensure the privacy and confidentiality of sensitive data being processed on remote servers, which often proved to be a significant concern for users. As a result, many Web metacomputing projects were left largely unutilized, with researchers and practitioners eventually pivoting to other more promising alternatives.

However local and WAN metacomputing experiments demonstrated the possibility to create a new class of computing resources based on the coupling of the unique supercomputers. These metacomputing systems were the prototype of distributed systems, and such computing structure introduced some specific hurdles, including different data representations on each system, variable CPU speeds that led to load imbalancing, different communication speeds for internal messages and messages between systems, lack of a common file system and resource management [71].

4. Exploring Grid Computing Initiatives

4.1. Information Wide Area Year Experiment

The next step towards the evolution of distributed computing in scientific applications was the I-WAY (Information Wide Area Year) project, launched in 1995. It was a network connecting supercomputers, databases and advanced visualization devices, like virtual reality (VR) at 17 different sites within North America, interconnected through 10 high-bandwidth ATM networks of varying bandwidth (typically 45–155Mbps) and protocols, using different routing and switching technologies. The idea was not to build a network but to integrate existing high bandwidth networks. I-WAY was used by over 60 application groups for experiments in high-performance computing [35].

The logical structure of the I-WAY project is shown in Fig. 2.

It demonstrates that the significant portion of the I-WAY's physical networking infrastructure leveraged pre-existing smaller ATM research networks. These individual networks were interconnected through collaboration with various prominent network service providers.

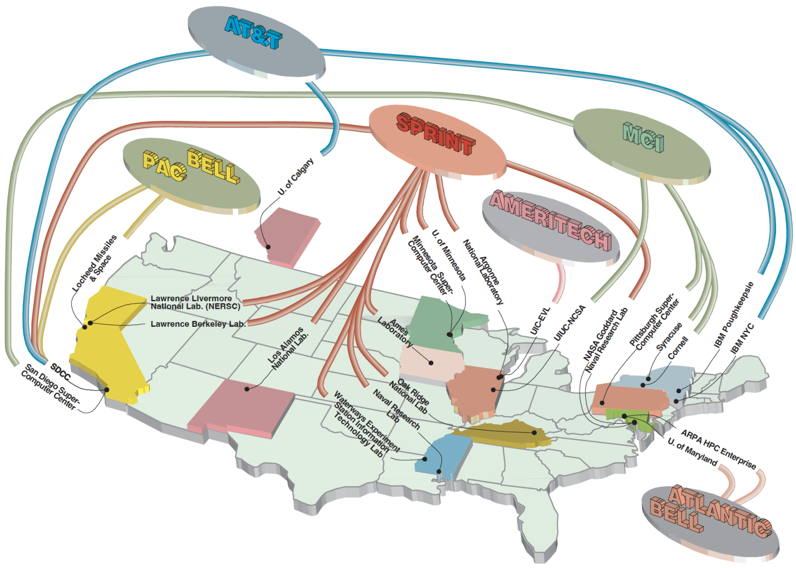

Figure 2. I-WAY Logical Structure [35]

I-WAY project identified several issues vital for designing future distributed computing environments:

- networking infrastructure should be relatively stable and persistent, because any disruption in the networking infrastructure can lead to loss of data, decreased performance, and reduced usability of the system;
- number of resources (people, machines, data) interconnected by the network should be large enough to create a critical mass of users;
- users should be involved with the computer scientists in development of experimental middleware. This can provide valuable feedback on the usability of the system, and also help identify any issues that may arise during implementation.

A major challenge of the I-WAY was providing a uniform environment across the geographically distributed and dispersed resources. To meet this challenge, researchers developed a middleware, called I-SOFT. This system was designed to run on dedicated I-WAY machines deployed at each participating site, and provided uniform authentication, resource reservation, process creation, and communication functions across I-WAY resources. It had a modular structure, with different components responsible for different functions. Some of the major components included a Resource Manager, which provided a uniform interface for managing resources, a Data Manager for a high-speed data transfer service, and an Execution Manager, which allowed users to submit and monitor jobs on remote systems.

The foundations of modern grid systems were laid in the I-WAY project. I-SOFT infrastructure was a precursor of the Globus Toolkit middleware framework, which became a de-facto standard of grid middleware in large-scale scientific applications for about two decades [43].

After the approach of connecting heterogeneous and distributed resources for collective use was tested in metacomputing systems, the development of distributed computing technology accelerated significantly. There was a need to provide a collection of solutions to problems that frequently come up when trying to build collaborative distributed applications: security, monitoring and resource discovery, access to computing and processing power, moving and managing data, deployment environments.

4.2. The Grid Book

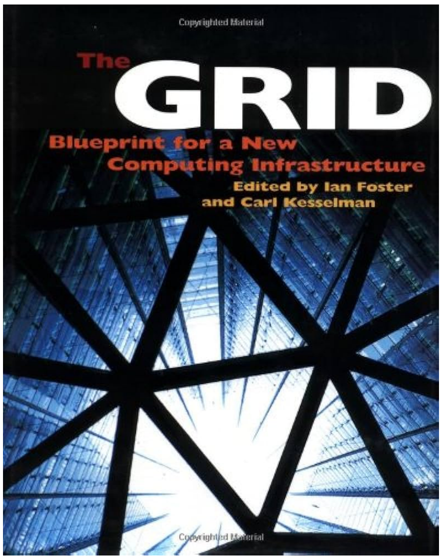

Figure 3. The Grid: Blueprint for a New Computing Infrastructure First Edition

Going back to the late 90's and early 2000s, the period of formation and development of distributed computing, the book "The Grid: Blueprint for a New Computing Infrastructure" by Kesselman, Foster (Fig. 3) was published, which had a catalyzing effect on the development of grid computing and the concept of grid was introduced in this monograph [42]. Kesselman and Foster were the winners of the British Computer Society's Lovelace Medal for their Grid work. They described the grid in the following words:

"The Grid is an emerging infrastructure that will fundamentally change the way we think – and use – computing. The word Grid is used by analogy with the electric power Grid, which provides pervasive access to electricity and, like the computer and a small number of other advances has had a dramatic impact on human capabilities and society. Many believe that by allowing all components of our information technology infrastructure – computational capabilities, databases, sensors, and people – to be shared flexibly as true collaborative tools, the Grid will have a similar transforming effect, allowing new classes of application to emerge".

Later on, Ian Foster, at one CERN Computing Seminar that was held in 2001, compared the Grid and the Web, called the Grid "The Web on Steroids"[8]. The Web is not yet a grid, with its open, general-purpose protocols that support access to distributed resources, it provides uniform access to HTML documents, but not the coordinated use of those resources to deliver negotiated qualities of service. On the other hand, grid ensures flexible, high-performant access to all resources, allowing on-demand creation of powerful virtual computing systems. So, whereas the Web is mainly focused on communication, grid computing enables resource sharing and collaborative resource interplay toward common business goals.

This book also provided an early reference to the term "grid middlewar" in the context of grid computing. The authors define grid middleware as *"software that bridges the gap between applications and lower-level network services, facilitating the development of distributed applications that span multiple administrative domains".*

4.3. Grid Forums

A special organization called the Global Grid Forum (GGF) was created in 1999, which included academic institutions, as well as computer system manufacturers and software providers [13].

The GGF created and published a large number of documents on grid computing architecture, security, data management, and resource management, which were used as a reference for many grid projects worldwide. The GGF organized its meetings three times

[8]http://cseminar.web.cern.ch/2001/0117/slides.pdf

a year, often in conjunction with other conferences and events, primarily in North America, Europe, and Asia.

Since that time, scientific communities and industry started to look seriously at grid computing as a solution to resource federation problems. In 2002, the GGF and IBM introduced the Open Grid Services Architecture (OGSA). With this development, grid services were defined as a specialized type of web services, enabling standard internet protocol-based interaction with grid resources [79]. Beginning from 2002 large European, American, and international projects were established based on grid computing and many commercial and industrial grids were launched.

In 2004, the Enterprise Grid Alliance (EGA) consortium emerged as a significant competitor to GGF and attracted substantial industry players like Fujitsu Siemens Computers, Hewlett-Packard, Intel, NEC, Oracle, Sun Microsystems, and EMC.

To enhance collaboration and consolidate the grid community, GGF and EGA finally merged in 2006. This merger resulted in the formation of the Open Grid Forum (OGF)[9], aimed at advancing grid and distributed computing environments.

4.4. Transition to Web 2.0

Another important milestone in the development of grid at the end of the millennium was the transition to Web 2.0 technologies and Web services in 1999. The term "Web 2.0" was introduced to refer to a new kind of web applications that use innovative architectures and toolkits to create very responsive and user-focused applications. This marked the progression from basic, static web pages to more dynamic ones with user-generated content. Additionally, the emergence of social media as a significant mode of internet communication was a hallmark of this second phase of internet development.

Web 2.0 was characterized by three key technologies: rich internet applications, web-oriented architecture, and the social web. There were two primary interface types that gained widespread acceptance: the "web as platform" interface or API (Application Programming Interface), which allowed companies to provide developers with access to data stores via APIs, and AII (Application Interaction Interface), which leveraged APIs and were classified as web applications.

The Web 2.0 and Grids Workshop at OGF19[10] held in 2007 showcased how Web 2.0 and grid technologies intersect in the realm of e-Science. While both offer solutions, Web 2.0's standout advantage lies in fostering collaboration and content creation. A key takeaway was the ease of use of Web 2.0, allowing seamless integration with existing resources, encouraging wider participation, and reaping associated benefits. Moreover, the grid's role was highlighted in providing a robust foundation for Web 2.0 APIs, emphasizing the importance of the API ecosystem. The session underlined the commonalities between workflows and mashups, hinting at the need for further exploration.

Web 2.0 and API technology have contributed significantly to the evolution of grid computing by enabling the creation of more dynamic and interactive grid-based applications and services. Web 2.0 technologies such as AJAX and RESTful APIs have made it easier to develop and deploy grid applications that can interact with a range of web-based services and data sources, making grids more widely accessible and practical for a range of applications and uses.

[9] https://gridcf.org/
[10] http://www.semanticgrid.org/OGF/ogf19/

5. Grid Middleware

During the early 1990s, the task of enabling programs to communicate with one another across multiple machines presented a significant challenge, particularly when different hardware systems, operating systems, and programming languages were involved. Typically, programmers were required to construct an entire protocol framework from scratch using sockets, or alternatively, the programs were unable to establish a connection.

The concept of middleware emerged in response to the growing need for distributed computing systems and the challenges of integrating heterogeneous and complex applications [20].

In the early 90s, scientific literature began to see an increasing number of publications dedicated to middleware, discussing its essence, applications, research, and improvement. Thus, in publications of that time, middleware was defined as an infrastructure within which the development of distributed applications is possible, a middleware service – the APIs and protocols it supports and sits between the platform and the application. Middleware components are generic across applications and industries, run on multiple platforms, are distributed, and support standard interfaces and protocols. Middleware that is transparent with respect to a standard API is more easily accepted by the market, and applications using an existing API can use the new service without modification. Providing a set of standardized, reusable functionalities that can be leveraged to build more complex and sophisticated systems was the essence of the service concept.

Below are multiple application scenarios for middleware systems within extensive scientific projects:

- secure data sharing and transfer between organizations;
- high-performance data movement and processing;
- authentication and authorization of users and their data access;
- high-throughput computing, job scheduling and resource allocation for computing-intensive research;
- access to distributed resources for scientific workflows and simulations;
- integration of various middleware systems for a seamless, user-friendly interface;
- data staging, synchronization, and management for distributed applications;
- in-memory data processing, querying, and distributed caching and integration with big data frameworks for near-real-time analytics.

As the demand for interoperability between different platforms and technologies grew, open standards-based middleware solutions emerged. One such solution was CORBA (Common Object Request Broker Architecture, 1991), which provided a way for different software systems to communicate using a standardized interface. For example, the U.K. Distributed Aircraft Maintenance Environment (DAME) project that was set up in 2004 and applied grid technologies to the challenging problem of computer-based fault diagnosis, utilized CORBA as a middleware to support intercomponent communication among different software modules. In particular, DAME was working to diagnose faults in Rolls Royce aircraft engines, based on sensor data recorded at the rate of one gigabyte per engine per transatlantic flight. Additionally, Java RMI[11] was used for communication between distributed Java objects,

[11] Java RMI – Remote Method Invocation – Java-specific technology relies on Java interfaces and allows Java objects to invoke methods on objects located on remote machines

and SOAP[12] – to enable remote collaboration and data sharing among its various stakeholders. Web Services enabled integration and interoperability of various components. The combination of these middleware technologies enabled DAME [55] to provide a flexible and scalable distributed environment for aircraft maintenance that improved efficiency and reduced downtime.

CORBA was a powerful middleware architecture that filled an important niche in distributed systems and object-oriented programming, but it was not well-suited to the needs of grid computing, and was outmatched by simpler, more focused solutions that were designed specifically for this field [61]. The reasons behind the waning popularity of CORBA are extensively examined in the publication "The Rise and Fall of CORBA: Valuable Lessons from its Mistakes" [52].

Other popular middleware solutions that emerged in the late 1990s and early 2000s were focused on various use-cases, some were designed to meet the needs of High-Throughput Computing (HTC), other – for High-Performance Computing (HPC), and there were even middleware developed specially for the in-memory and volunteer distributed computing.

Below we discuss several examples of general-purpose middleware systems. However, there are a lot of more specialized middleware, developed for more specific purposes: for data management, job scheduling, resource discovery, etc.

5.1. HTCondor

HTCondor

Condor is a middleware system that focuses primarily on high-throughput computing (HTC), particularly for running large-scale batch jobs on a network of distributed computing resources.

The first version of Condor was released in 1988, and it quickly gained popularity among researchers and scientists who needed to run large-scale batch jobs on a network of distributed computing resources. *"Condor – A Hunter of Idle Workstations"* – that is the title of the program paper, describing the system design of the first version of Condor and its main principles [58]. The motivation of Condor was to maximize the utilization of workstations interconnected by high capacity networks with as little interference as possible between jobs it schedules and the activities of the people who own these workstations.

Figure 4 demonstrates the initial structure of Condor's scheduler. Each machine had a local scheduler and background job queue for user job submissions. A central workstation housed a central coordinator, local scheduler, and background job queue. The central coordinator polled stations every two minutes to allocate remote cycles and manage pending background jobs. Local schedulers monitored station capacity and preempted background jobs for local user activity. Idle workstation capacity was allocated by the central coordinator to local schedulers with pending jobs, with local schedulers making decisions on job prioritization if multiple background jobs were waiting. Condor offered a comprehensive set of features including job management, scheduling policies, priority schemes, resource monitoring, and resource management. What set Condor apart from other so-called Resource Management Systems (RMS), was its unique architecture and mechanisms, enabling outstanding performance in environments where traditional RMS systems struggled – such as sustained high-throughput computing and opportunistic computing. In a high-throughput computing environment, the primary objective is to efficiently utilize all available network resources to provide computational power over extended periods with fault

[12] SOAP – Simple Object Access Protocol – protocol-independent, XML-based messaging protocol that is used for exchanging structured information in the implementation of web services

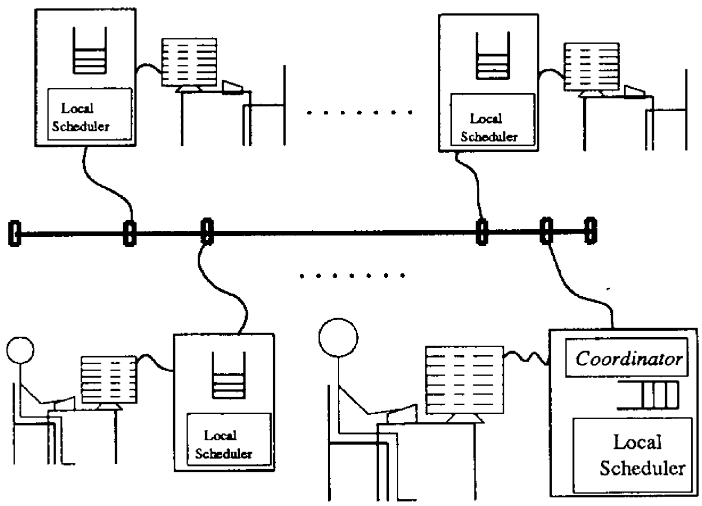

Figure 4. The Condor Scheduling Structure [58]

tolerance. Opportunistic computing, on the other hand, focuses on resource utilization whenever they are available, without requiring 100% resource availability. These two goals naturally complement each other. And Condor's remarkable flexibility lied in its ability to seamlessly integrate High Throughput Computing (HTC) with volunteer computing, effectively harnessing idle resources. For example, consider the capability Condor offered to run jobs solely on desktop workstations during idle periods when both the keyboard and CPU remained idle. In the event of a user keystroke while a job was ongoing on a workstation, Condor adeptly shifted the job to another workstation, ensuring a smooth continuation from the point where it was interrupted. Additionally, Condor's impressive mechanics enabled preemptive-resume scheduling for dedicated compute clusters.

This vital functionality empowered Condor to handle priority-based scheduling on clusters with utmost efficiency. Whenever a node in a dedicated cluster remained idle without a task, Condor could opportunistically allocate it for computing purposes.

Over the years, Condor had undergone several iterations and enhancements to keep up with changing computing requirements. In the early 2000s, with the growing interest in distributed computing and the need for such computing systems to support more diverse applications, the developers of Condor implemented a new version called Condor-G (where G stands for Grid) [45]. They added many features that made it easier to use Condor with grid standards such as the integration with Globus Toolkit services (Grid Resource Allocation and Management (GRAM), Globus Toolkits Global Access to Secondary Storage (GASS), Globus Toolkits Grid Security Infrastructure (GSI)), which allowed it to access remote computing infrastructures and provided users with an integrated and transparent interface to use grid resources.

Condor-G demonstrated its strengths when four mathematicians from Argonne National Laboratory, University of Iowa, and Northwestern University used it and several other technologies to solve a problem known as NUG30 challenge. The paper "Distributed Computing in Practice: The Condor Experience" describes this challenge as follows: *"...a solution to NUG30 was discovered utilizing Condor-G in a computational run of less than one week. During this week, over 95,000 CPU hours were used to solve the over 540 billion linear assignment problems necessary to crack NUG30. Condor-G allowed the mathematicians to harness over 2500 CPUs at ten different sites ... spanning eight different institutions"* [80]. Overall, the winning team was able to achieve a 15x speedup compared to the NERSC baseline solution and a 6x speedup compared to the fastest solution submitted in the challenge prior to theirs.

Condor-G actively participated in various prominent projects, including the Grid Physics Network (GriPhyN), the International Virtual Data Grid Laboratory (iVDGL), the Particle Physics Data Grid (PPDG), the NSF Middleware Initiative (NMI), the TeraGrid, and the NASA Information Power Grid (IPG). Furthermore, as a founding member of the National Computational Science Alliance (NCSA) and a close collaborator of the Globus project, Condor-G played a pivotal role in advancing computational science and grid technologies.

In late 2012 Condor-G was upgraded and modernized to HTCondor (High-Throughput Computing Condor)[13]. HTCondor builds on the features and capabilities of Condor, adding new functionalities, performance improvements, and better support for current computing environments and architectures. Since the transfer, HTCondor has become a widely adopted solution for managing large-scale scientific workloads in the United States and around the world.

5.2. NIMROD – HPC in Plasma Physics

NIMROD, developed in 1995, was a tool for performing parameterized simulations over networks of loosely coupled workstations [29]. Using NIMROD the user could interactively generate a parameterized experiment. NIMROD then controlled the distribution of jobs to machines and the collection of results. In this context, this middleware was highly effective for research studies with users who have varying degrees of parallel programming skills. It was more suitable for individual researchers and small research groups who require access to parallel computing resources, but not for larger organizations and collaborations with complex workflows, data management needs, and secure access requirements.

In 2005 NIMROD was modernized to NIMROD/G (for Grid) [24] by a team at the Los Alamos Laboratory in the US. The project utilized funding from the Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) program. It became a highly specialized Grid middleware that had been designed specifically to meet the needs of the plasma physics simulation: it included functionality for modeling magnetohydrodynamics (MHD) and other relevant physical phenomena, a number of custom plugins and modules that were specifically tailored to the needs of plasma physicists and it used a custom job submission and management system that was optimized for large-scale, long-running simulations [49]. It provided a framework for running parallel processing jobs across multiple heterogeneous computing resources, as well as managing data transfer between those resources. While it might not be as broadly applicable to other scientific domains as some other middleware systems, it was highly effective for the specific use cases for which it was developed.

While NIMROD/G is no longer actively maintained, it was used in a variety of scientific and engineering projects over the years: DIII-D tokamak experiment at General Atomics in San Diego, California, USA; National Spherical Torus Experiment (NSTX) at Princeton Plasma Physics Laboratory in New Jersey, USA; Korea Superconducting Tokamak Advanced Research (KSTAR) experiment at the National Fusion Research Institute in Daejeon, South Korea; ITER project – an international collaboration to build a fusion energy prototype reactor in Cadarache, France.

[13] https://htcondor.org/

5.3. Globus Toolkit

Globus Toolkit is a middleware system specifically designed for distributed data-intensive scientific research applications, which require the transfer and management of large volumes of data across geographically distributed resources. In 1996 the Globus Alliance was formed to conduct R&D for the technology, standards and systems that form the grid. Alliance members eventually produced open-source software Globus Toolkit (GT) that evolved out of trying to solve real problems in real projects. It was a community-based, open-source set of services and software libraries that support grids [41].

The first version of the Globus Toolkit (GT1) emerged in the late 1998, during a time when the concept of grid computing was just beginning to emerge. The main functionality of GT1 was to provide a software infrastructure that could enable scientists and researchers to share distributed computing resources such as computing power and data storage over the Internet. GT1 was a relatively simple toolkit, with a limited set of functionality, focused mainly on providing low-level mechanisms for distributed computing, such as security and authentication mechanisms, resource discovery, and job submission and management.

The Globus Toolkit version 2 (GT2) emerged in 2002, and was a significant upgrade to GT1. GT2 was designed to make it easier to build and manage grid systems by providing a standardized set of middleware services and tools for building grid applications. GT2 introduced many new features and improvements over its predecessor GT1, including better support for authentication and authorization, support for multiple security mechanisms, and improved data transfer performance. It was widely adopted by many grid communities and projects, including the European DataGrid (EDG) and the GriPhyN project, among others.

In general, the four main GT protocols were:

- *Grid Security Infrastructure (GSI)* – authentication, authorization, policy, delegation. GSI had become a standard for grid security. It was based on the free SSLeay package and used X.509. It enabled "Single sign-on" (SSO) on the grid, user identity was guaranteed with a single certificate that helped to avoid frequent logins into various resources with different passwords;
- *Grid Resource Allocation Management (GRAM)* – for the remote allocation, reservation, monitoring, control of compute resources;
- *Grid Resource Information Service Protocol (GRIS)* – provided access to structure and state information, and
- *Grid File Transfer Protocol (GridFTP)* – high-performance data access and transport protocol.

The main challenge of the GT2 was its applicability for business processes: it was too much like a distributed batch system. New technologies required new concepts of grid middleware. Instead of functioning as a distributed batch system, it was supposed to be a network of services, small programs with standardized interfaces, geographically distributed. So, to meet the new requirements, a GT3 version was implemented in 2004, which was based on grid services.

In 2006, GT4 [39] was introduced with the addition of Web Services, XML-based mechanisms for describing, discovering, and invoking network services. GT4 provided a set of infrastructure services that implemented interfaces for managing computational, storage, and other resources. In many Globus deployments, such as TeraGrid, Open Science Grid, Cancer Bioinformatics Grid (caBIG), EGEE, LHC computing Grid, UK National Grid Service, China Grid, China National

Grid and NAREGI, these services were deployed to support a range of different application communities.

Effective use of high-speed networks for research required breaking the usability barriers that impede network use by non-expert users. Thus, the Center for Enabling Distributed Petascale Science (CEDPS) project launched the Globus Online project in 2009 [40], enabling the reliable high-performance research networking for the masses. Globus Online was a cloud-based service that provided data management and transfer capabilities to researchers and organizations who require data-intensive computing. If Globus Toolkit provided resource federation as a service, Globus Online was the cloud-hosted file transfer service, or software-as-a-service (SaaS) provider, taking the responsibility for managing the end-to-end data transfer processes, performed via GridFTP.

In 2018, after two decades of being a de-facto standard for grid applications, the Globus Toolkit, announced the end of support of the open-source project. The decommissioning of the Globus Toolkit affected a number of grid computing projects that relied on its technology. In response to the end-of-support of the Globus Toolkit the Grid Community Toolkit (GCT) was created[14]. It is an open-source fork by the Grid Community Forum (GCF) of the venerable Globus Toolkit created by the Globus Alliance. The GCT is derived from the Globus Toolkit, but is not the Globus Toolkit. Further, the GridCF is not a part of the Globus Alliance. Through this initiative, the following objectives are sought:

- Ensure the preservation of vital features, such as the Grid Security Infrastructure (GSI) and an open-source GridFTP implementation, by maintaining their quality and security.
- Foster collaboration among various grid organizations to distribute the responsibility of supporting these critical functionalities.
- Create an inclusive and open platform where external contributors can actively participate and enhance the software.

5.4. UNICORE

UNICORE (Uniform Interface to Computing Resources, 1997) [36] is an open-source middleware that provides a job submission and management system for grid computing. Originally this project was initiated in the HPC domain. It is the project of the German Ministry for Research and Education (BMBF). Initial aim was in creating a network of supercomputing centres and providing a uniform interface while using existing technologies. UNICORE middleware was developed to meet the specific requirements of complex distributed data-intensive applications that require high-performance computing environments. The implementation was based on Java and Java applets that ensure high portability. UNICORE was initially designed as a monolithic system, where various modules such as authentication, authorization, job submission, and data transfer are all tightly coupled. UNICORE provided a more restrictive job submission model where users must first register their application on the server.

UNICORE passed through many stages in the evolution:

- Initial focus on applications across multiple disciplines in Europe in 2001, driven by research projects sponsored by the European Commission. UNICORE saw deployments in

[14] https://gridcf.org/gct-docs/latest/index.html

several European high-performance computing centers, the creation of a High-Level API, and the introduction of interactive access to HPC systems through UNICORE.

- Transition to open-source in 2004 when the UNICORE software was made available under the BSD License via a SourceForge project.
- Shift towards Web Services-based architecture in 2007, followed by a focus on Data Oriented Services in 2011, emphasizing high-speed file transfers and enhancing support for data management and data-intensive applications within UNICORE.
- Major updates with UNICORE 7 in 2014, UNICORE 8 in 2020, introducing significant improvements.
- Ongoing development culminating in UNICORE 9 in 2022, representing a fully REST-based system.

Compared to the Globus Toolkit, UNICORE provides a more integrated and comprehensive interface to grid resources including an advanced security system that can support the access control policies and data privacy requirements of complex distributed applications. UNICORE middleware was designed for European grid infrastructures but is now used around the world. It was integrated with such highly-recognized scientific grid projects like, D-Grid and Human Brain Project.

One of the flagship UNICORE projects is Jupyter-JSC[15], so-called "Supercomputing in Your Browser" (Jlich Supercomputing Centre (JSC)). It combines interactive supercomputing with UNICORE, offering researchers and analysts an efficient platform for data research, analysis, and visualization. Jupyter's open-source, web-based design allows for diverse workflows and programming methods in a single interface. JupyterHub enables multi-user functionality, making it suitable for supercomputing centers. UNICORE ensures data access by integrating tools like XUUDB[16], UNICORE/X[17], TSI[18], allowing users to spawn Jupyter applications with their HPC accounts. At the JSC, Jupyter-JSC provides direct web access to start and connect Jupyter or JupyterLab. UNICORE manages jobs and provides reliable access and information output.

5.5. ARC

Advanced Resource Connector (ARC)[19], in the past called NorduGrid middleware, is a middleware system established in 2002 to provide a management framework for distributed computing resources in the Nordic countries [57]. Scientific and academic computing in the Nordic countries exhibited a distinct characteristic with a multitude of small and medium-sized facilities, each differing in nature and ownership.

Considering this, the development of ARC had prioritized the need for a portable, compact, and manageable middleware solution with a focus on interoperability between different software and hardware systems that caters to both server and client sides. Notably, the first stable client version of ARC package occupied only 14 megabytes and was compatible with most Linux distributions, allowing installation at any accessible location by nonprivileged users. And for computing service, only three main processes are needed: file transfer service, Grid Manager and Local Information Service [3].

[15] https://jupyter.jsc.fz-juelich.de/

[16] UNICORE user database, manages access to resources

[17] Server, the main component, interacts with XUUDB for authentication purposes

[18] Target System Interface – interface of the execution environment, launches tasks on the target system

[19] https://www.nordugrid.org/arc/arc6/index.html

ARC is designed to manage the execution of large numbers of independent tasks (HTC workloads), such as those commonly found in scientific computing and data-intensive research applications. However, it can be used to manage HPC workloads as well. Since 2002, ARC has continuously supported production-level systems and demonstrated its exceptional performance in demanding High Energy Physics (HEP) computing tasks. In fact, it is the pioneering middleware that has successfully provided services for such a massive global data processing endeavor.

Being a part of HEP distributed environment in ATLAS experiment at the LHC, ARC introduced several new features and possibilities in the traditional ATLAS computing model. Firstly, ARC allowed for pilot jobs that come with pre-cached input files, reducing the need for data transfer during job execution. Secondly, ARC enabled automatic migration of jobs between different sites, improving job allocation and resource utilization. Thirdly, it allowed for the integration of remote sites that do not have direct storage connectivity, thus expanding the pool of available computing resources. Next, ARC provided automatic brokering for jobs with specific resource requirements, facilitating efficient job allocation. Additionally, ARC offered an automatic data transfer model that allowed computing sites to participate in ATLAS' global task management system without centralized brokering or data transfer services. Finally, ARC included a powerful API with Python and Java bindings, making it easy to build new services for job control and data transfer [38].

The middleware leveraged well-established open source solutions such as OpenLDAP, OpenSSL, SASL, and the Globus Toolkit 2 (GT2) libraries. ARC introduced pioneering solutions that were crucial for a reliable and high-performing middleware, including the Grid Manager, ARC GridFTP server, information model and providers, User Interface and broker, an extended Resource Specification Language (xRSL), and a comprehensive monitoring system. These innovations provided by ARC enhanced the functionality and effectiveness of the middleware.

5.6. BOINC – Volunteer Computing

BOINC is a platform for Volunteer Computing (VC) that was developed in 2004. It is a client-server desktop grid middleware that has grown to power very large computational projects. BOINC clients request computing jobs to a central server and run them alongside other regular applications [11].

"Volunteer computing" is the use of consumer digital devices for high-throughput scientific computing. It can provide large computing capacity at low cost, but presents challenges due to device heterogeneity, unreliability, and churn. VC is a type of computing architecture that is best suited to High-Throughput Computing (HTC) workloads, where the goal is to complete a high volume of jobs rather than focusing on low job turnaround time. However, VC is less well-suited for workloads that require high memory or storage requirements, or where the ratio of network communication to computing is very high.

Compared to other forms of HTC, VC presents unique challenges that must be addressed by the platform. For example, VC computers are anonymous, untrusted, inaccessible, and uncontrollable. They may behave unpredictably and cannot be punished or stopped. Additionally, VC computers are heterogeneous in all hardware and software dimensions, which requires either multiple application versions or the use of virtualization and makes it challenging to estimate job runtimes. Another challenge of VC is creating the resource pool, which involves recruiting and retaining volunteers. This requires incentive features such as teams, computing credit accounting, and screensaver graphics. Finally, the scale of VC is much larger, with potentially millions of computers and millions of jobs per day. Therefore, the server software must be efficient and scalable. BOINC platform takes into account all these challenges.

SETI@Home (1999 – 2020) was one of the largest projects that used BOINC as a middleware [12]. This project truly popularized distributed computing and showed that it could work. SETI@Home was an effort by the Search for Extraterrestrial Intelligence (SETI) at the University of California at Berkeley. The project was started in 1999 to analyze the radio telescope signals. There were over three million users who volunteered their idle computing resources in the search for extraterrestrial intelligence. Anyone who has an Internet connection and some spare CPUs can participate by running a free program that analyzes radio telescope data. The project was retired on March 31, 2020, after over 20 years of operation. The data collected by SETI@Home is now being archived for future analysis by astronomers.

Below, there are some other examples of grid projects that used BOINC middleware:

- **SimGrid (1999 – present time)** [26] – a distributed computing project that provides tools and libraries for distributed computing research, such as grid and cloud computing, high-performance computing, and peer-to-peer networks. The project provides a platform for researchers to model, simulate, and analyze various strategies for large-scale distributed systems. SimGrid runs on various computing systems.
- **Folding@home (2000 – present time)** [17] – a distributed computing project that simulates protein folding, misfolding, and related diseases. The project is run by a consortium of universities that has built one of the world's largest and most powerful supercomputers, with the help of over 4 million volunteers who donate their unused computing power for research. The project consists of tens of thousands of CPUs and GPUs from around the world.
- **Climateprediction.net (2003 – present time)**[20] – a distributed project that uses global climate models to predict the possible impacts of climate change on the earth. The project uses hundreds of thousands of volunteers' computers to run simulations of the atmosphere, ocean, and land surface. The project has been used to produce some of the most extensive surveys of the effects of climate change to date, and it has resulted in numerous published research papers.
- **Rosetta@home (2005 – present time)**[21] – project focused on protein structure prediction and designs of new proteins to aid computational drug discovery. The project uses over 600,000 CPUs and GPUs from around the world.
- **Einstein@home (2005 – present time)** [8] – searching for gravitational waves from black holes, pulsars and other objects in the universe. The project uses data from the Laser Interferometer Gravitational-Wave Observatory (LIGO) detectors. It has distributed over 30 million work units to volunteer computers and currently uses over 200,000 devices to process data.
- **MilkyWay@home (2006 – present time)** [33] – simulations of the formation and evolution of galaxies, including the Milky Way. The project uses volunteer computing

[20] https://climateprediction.net/

[21] https://boinc.bakerlab.org/

power for parameter studies, such as studying the effects of dark matter. MilkyWay@Home runs on over 100,000 CPUs and GPUs contributed by volunteers around the world.
- **GPUGRID.net (2007 –present time)**[22] – a distributed computing project focused on molecular dynamics simulations. It performs pharmacological simulations, including simulations of protein-ligand interactions and drug design. The project runs on more than 5,000 GPUs from volunteers worldwide.
- **Asteroids@home (2008 – present time)** [82] – searching for unknown asteroids and comets by processing astronomical data using image-recognition algorithms. The project aims to assist professional astronomers in discovering asteroids that could pose a threat to our planet. Currently, the project runs on about 8,000 devices contributed by volunteers worldwide.

5.7. Specialized Middleware Systems in High-Energy Physics Projects

Grid middleware systems, like Globus Toolkit, Condor, UNICORE, ARC, gLite, play a vital role in enabling efficient data management, job execution, resource utilization, and collaboration in grid applications. The specific middleware systems utilized may vary depending on the requirements and objectives of the grid projects. In various projects, different combinations of middleware systems are used based on the specific needs and goals of each project.

For example, there were several different middleware packages available in the D-Grid project: Globus Toolkit, UNICORE, LCG/gLite, GridSphere and the Grid Application Toolkit (GAT). LIGO Data Grid utilized a combination of Globus Toolkit and Condor-G services. The widest variety of middleware systems, including custom middleware services, is in the WLCG – the largest grid infrastructure in the world. There are several scientific experiments connected to the WLCG: ATLAS, ALICE, CMS and LHCb – these are the largest international high-energy physics projects that brought notable contributions in the development of elementary physics. In the realm of experimentation, middleware technologies are utilized across various experiments, with some similarities and differences in their deployment as per specific experiment requirements.

The distinctive data handling and processing demands of each of the experiments conducted at the LHC have given rise to the development of separate distributed frameworks for data management, job submission, and execution by each respective experiment team:
- **PanDA (Production and Distributed Analysis)** is a workflow management system developed by the ATLAS experiment at CERN. It is designed to manage large-scale data processing and analysis tasks, and can be used with a variety of computing resources including grid, cloud, and HPC systems [60].
- **Rucio Distributed Data Management system** – a primary distributed data management system in ATLAS since 2014. Rucio is responsible for the allocation and placement of files across distributed storage systems, data transfer monitoring and optimization, and authentication and authorization of user access to data. Rucio also provides ATLAS with a centralized data catalog, where the metadata of all datasets and files are stored and indexed, making it easy for users to access the data they need. Furthermore, Rucio supports data preservation and long-term archiving, ensuring that the data produced by ATLAS remains accessible and usable for future scientific studies [47].

[22] https://gpugrid.net/

- **DIRAC (Distributed Infrastructure with Remote Agent Control)** is a middleware system developed by the LHCb experiment at CERN. It is designed to manage distributed data processing and analysis tasks, and can be used with a variety of computing resources including grid, cloud, and HPC systems. It includes features such as job queuing, job monitoring, and data replication [25].
- **ALIEN (ALICE Environment)** is a distributed computing system developed by the ALICE experiment to manage large-scale data processing and analysis tasks, and can be used with a variety of computing resources including grid, cloud, and HPC systems. Some of its key features include support for user authentication and authorization, job prioritization, and data management [75].
- **Grid Control System (GlideinWMS)** is the primary system used for job submission and management on the grid in the Compact Muon Solenoid (CMS) experiment. It allows users to submit and monitor jobs to various computing resources around the world, including those provided by the WLCG. GlideinWMS is often used for large-scale Monte Carlo simulations, where virtual particles are generated and their interactions with the detector are simulated to predict the outcome of physical processes [76].

6. Instances of Scientific Grids

Numerous scientific grids have emerged since the beginning of 2000s. Each initiative has its own specifics and the scale of computing resources. In this section the largest and the most interesting grids are described.

TeraGrid (2001 – 2011) [27] was a collaboration between multiple supercomputing centers in the United States. In the beginning of 2000s it was the most ambitious high-performance grid project in the United States (Fig. 5[23]). It offered over a petaflop of total compute capabilities and many different services and gateways to thousands of US scientists. The genesis of the TeraGrid project was the Distributed Terascale Facility (DTF). The DTF was conceived in October 2001 as a distributed high-performance computing grid that could link supercomputers as well as some specialized resources for data management and visualization. As of July 2006 the range of scientific disciplines in TeraGrid was from molecular bioscience, physics, astronomical sciences and chemistry to the neuroscience, communication systems and environmental biology. Until the beginning of the Large Hadron Collider (LHC) project, TeraGrid was the worlds largest, most comprehensive distributed cyberinfrastructure for open scientific research. The project was finished in 2011.

GridPP (2001 – still ongoing)[24] is a community of particle physicists and computer scientists based in the United Kingdom and at CERN. Initially it was formed to cater for the substantial computing demands of the Large Hadron Collider experiments, representing the UK in the Worldwide LHC Computing Grid (WLCG). As the GridPP community has grown and evolved, however, many user communities from a wide range of disciplines have taken advantage of the computing resources offered by GridPP.

The institutes that form the GridPP Collaboration are organized as follows: The UK Tier-1 center, which is hosted at the Rutherford Appleton Laboratory (RAL), 18 UK university-based

[23]https://www.nsf.gov/news/mmg/media/images/teragrid_h.jpg
[24]https://www.gridpp.ac.uk/

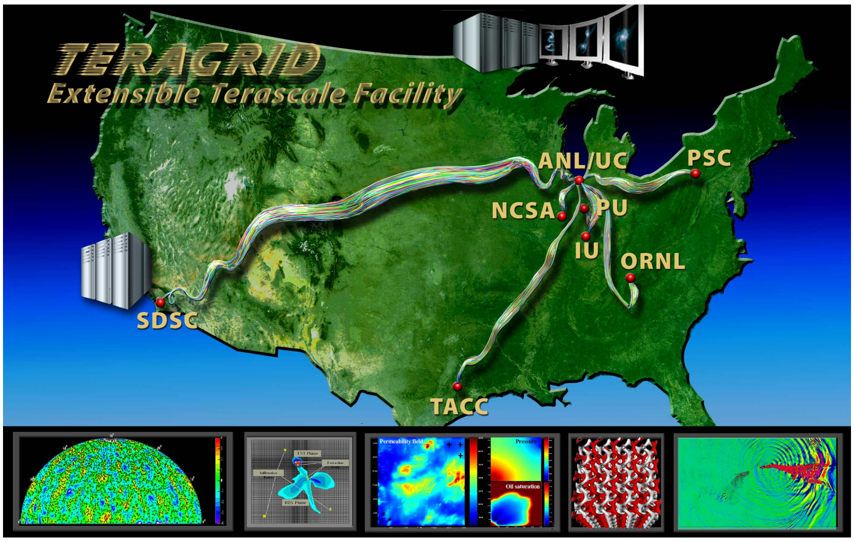

Figure 5. TeraGrid Members

Figure 6. GridPP Collaborating Institutes

Tier-2 sites, which are grouped into four Regional Tier-2s (LondonGrid, NorthGrid, ScotGrid, SouthGrid)[25].

As of the beginning of 2012, these GridPP sites collectively offer approximately 30,000 logical CPUs and around 29 PB of disk-based storage. The Deployment Board (DB) governs both the Tier-1 center and the regional Tier-2s[26].

GridPP supports research in physics (LSST, Fusion, Supersymmetry, LUCID, MAGIC) and particle physics (LHC, H1, BaBar, D0, SNO+), and other sciences (BioMed, EPIC, PRaVDA, Proteomics and Phylogenomics at QMUL, WISDOM).

[25]https://www.gridpp.ac.uk/about/collaborating-institutes/
[26]https://www.gridpp.ac.uk/infrastructure/

International Gravitational Wave Observatory (IGWN) Computing Grid (2002 – still ongoing) [9] is a distributed, high-throughput computing infrastructure based on HTCondor middleware. It facilitates the processing of scientific workflows across a geographically diverse set of resources. These resources include dedicated computing resources provided by IGWN member groups, as well as opportunistic resources from external partners. The grid is connected through a support infrastructure maintained in collaboration with the Open Science Grid (OSG). It also offers utilities for remote access to data and services, enhancing the capabilities of researchers and enabling collaboration within the IGWN community[27].

The LIGO project uses interferometric detectors to search for gravitational waves, ripples in space-time caused by the acceleration of massive objects. LIGO has been operational since 2002, with the first round of observations running from 2005-2007. The second round of observations began in 2015, and has resulted in several major gravitational wave detections, including the first observations of binary black hole mergers and binary neutron star mergers.

Actually, there are three detectors in the infrastructure of gravitational-wave astronomy: LIGO (United States), Virgo (Italy and France) and Kamioka Gravitational Wave Detector (KARGA, Japan). To successfully detect gravitational waves, a robust computational framework is necessary to not only support the detectors themselves but also the data analysis pipelines that are employed to locate and define gravitational wave signals. Virgo, LIGO and KAGRA collaborations joined efforts to move from partially interoperable owned computing resources to a wholly shared common computing infrastructure IGWN by adopting common tools and interfaces.

The Gravitational Wave community is ushering in a new era of computing with several transformative features. These include achieving full interoperability between the Virgo, LIGO, and KAGRA observatories, establishing a common and sustainable computing environment, standardizing a uniform runtime environment for offline pipelines, embracing scalability and heterogeneous resource utilization, and adopting mainstream, widely-used tools. These advancements pave the way for enhanced collaboration, increased computational efficiency, and optimized resource utilization, ultimately propelling gravitational wave research towards new frontiers of discovery.

Worldwide LHC Computing Grid (WLCG, 2002 – still ongoing) [22]: high energy physicists designing the LHC were the first to realize that they needed grids to federate computing systems at hundreds of sites to analyze the many petabytes of data to be produced by LHC experiments. Thus, they launched the EU-DataGrid (2001) project in Europe, the Particle Physics Data Grid (PPDG, 2002) and Grid Physics Network projects in the US – these two efforts led to the creation of the Open Science Grid (OSG, 2003) in the US, EGEE (Enabling Grids for E-sciencE, 2004) and then EGI (European Grid Infrastructure, 2008) in Europe. The international LHC Computing Grid (LCG, 2006), which was later renamed to the Worldwide LCG Computing Grid (WLCG). This was done to better reflect the global scale and collaborative nature of the project.

WLCG is the largest grid infrastructure. At an early stage of the development of the LHC computing model (2000s), it was decided to combine the existing and newly constructed computing centers (more than 200) into a distributed data processing center and to do this in such a way that physicists from universities and scientific organizations of the participating countries

[27]https://computing.docs.ligo.org/guide/dhtc/

would have equal opportunities to analyze information. As a result of the work of physicists, scientists, and IT engineers, WLCG was created (Fig. 7)[28].

Figure 7. WLCG sites

Today the WLCG is the largest academic distributed computing network in the world, consisting of about 300 computing centers in 42 countries. More than 10K scientists use these centers to analyze LHC data in search of new physics phenomena. In the WLCG, up to 3M physics jobs run daily, the total storage space exceeds 1 EB, the data processing results are archived, distributed between data processing and analysis centers, and go directly to the physicists workplace. Such a system can be compared with a huge computing complex the nodes of which are connected by high-speed Internet. Data transfer rates between centers are up to 20 GB/s (average value during a day and 35 GB/s at peak). Now the WLCG consists of four tiers: 0, 1, 2, and 3.

- *Tier 0:* CERN Data Center, responsible for data archiving, reconstruction, and reprocessing during LHC down-times.
- *Tier 1:* 13 large centers with CPU and storage capacity, providing round-the-clock support. They store data, perform reprocessing, distribute data to Tier 2s, and store simulated data.
- *Tier 2:* Universities and scientific institutes that store data and provide computing power for analysis tasks. There are around 170 Tier 2 sites globally.
- *Tier 3:* Computing facilities in universities and research labs, retrieving data from Tier 2 for processing and analysis.

WLCG has mesh structure where data centers are interconnected, allowing for efficient distribution of workloads and computing resources. This mesh structure provides high redundancy and robustness. In case of a data center failure, workloads can be automatically rerouted to other nodes in the network, ensuring continuous availability of services without major disruption.

Russian Data Intensive Grid (RDIG, 2003 – still ongoing) [21] is a national-level initiative in Russia that aims to facilitate and support large-scale data-intensive research projects. Initally it was established at the initiative of the Kurchatov Institute National Research Centre and the Joint Institute for Nuclear Research in Russia to participate in the distributed data processing of experiments at the LHC. Later, its objective evolved beyond high-energy physics to encompass a broader range of scientific disciplines, including astrophysics, bioinformatics, and climate modeling, among others.

[28]https://wlcg.web.cern.ch/using-wlcg/monitoring-visualisation

RDIG connects the largest Russian scientific centers, such as IHEP (Institute for High Energy Physics), IMBP RAS (Institute of Mathematical Problems in Biology), ITEP (Institute of Theoretical and Experimental Physics), JINR (Joint Institute for Nuclear Research, Dubna), KIAM RAS (Keldysh Institute of Applied Mathematics), PNPI (Petersburg Nuclear Physics Institute), NRC KI (Kurchatov Institute, Russia's National Research Center), SINP MSU (Skobeltsyn Institute of Nuclear Physics at Moscow State University), SPBU (Saint Petersburg State University), NovSU (Novgorod State University).

Now RDIG is an important component of the IT infrastructure being developed in Russia, which ensures the functioning of mega-science projects, including providing storage for the national genetic information database and biorepository centers - key elements of the genetic research infrastructure.

International Lattice Data Grid (ILDG, 2003 – still ongoing) [56] is a global collaboration of lattice quantum chromodynamics (QCD) groups. It is designed to provide a unified framework for storing and sharing lattice QCD data sets in a secure, efficient, and standardized manner. ILDG initially emerged as a federation of interoperable yet independently operated infrastructures and services, often referred to as a "Grid of Grids" It consists of several regions (Australia, Japan, Continental Europe, UK, US). Each regional grid has the autonomy to implement and manage its services in a manner compatible with ILDG standards. For instance, the regional grids in Japan and Europe use rather different solutions for data storage: a global file system (GFARM) in case of Japan Lattice Data Grid (JLDG) [26], and several (distributed) storage elements with SRM interfaces being part of the Worldwide LHC Computing Grid in case of European Latfor DataGrid (LDG). One crucial responsibility of the regional grids is to acquire the required storage resources for their data.

ILDG's primary role is to facilitate community-wide user registration, promoting collaboration among its members. Additionally, ILDG maintains two working groups tasked with developing ILDG-wide metadata schemata and APIs for the regional grid services. Through these standardized specifications, data sharing and metadata searching become seamless and accessible across the entire ILDG community.

Several large LQCD collaborations in Japan have been working on QCD simulations using super-computers[29].

European LDG operates at the following institutions: DESY (Hamburg and Zeuthen, Germany), IN2P3 (Lyon, France), INFN (Bologna, Italy), JSC (Jülich, Germany), RUG (Groningen, Netherlands), SARA (Amsterdam, Netherlands)[30].

UK Lattice Field Theory region currently comprises 35 academics from 10 different UK institutions (Swansea, Glasgow, Plymouth, Liverpool, Southampton, Edinburgh, Imperial, Cambridge)[31].

US Region operates at the resources of Jefferson Lab LQCD (JLAB), Fermilab Lattice QCD Facility, BNL clusters for USQCD.

Naregi National Research Grid Initiative, NAREGI (2003 – 2012) [68] is one of the major Japanese national IT projects to develop and use state-of-the-art, high-performance general-purpose supercomputers promoted by the Ministry of Education, Culture, Sports, Science and Technology. The Computational Nanoscience Center at the Institute for Molecular Science is actively utilizing grid computing technologies to conduct cutting-edge research in the

[29] https://www.jldg.org/system.html
[30] https://hpc.desy.de/ldg
[31] https://generic.wordpress.soton.ac.uk/uklft/

field of nano-science and nanotechnology simulation applications. This research aims to facilitate the discovery and development of novel materials and advanced nano-devices for future generations.

Enabling Grids for E-sciencE (EGEE, 2004 – 2010) [46]. The main purpose of EGEE was to create a sustainable pan-European grid infrastructure to support scientific research and collaboration across different disciplines. EGEE involved participation from numerous countries across Europe and beyond. Some of the countries that participated in the project included Belgium, France, Germany, Italy, Netherlands, Spain, Switzerland, United Kingdom, Greece, Portugal, Russia, Taiwan, and many more. The project comprised a network of data centers that collectively formed a powerful computing grid. The exact number of data centers evolved over time as new institutions joined the project. EGEE shared its grid infrastructure with LCG, and the grid operations activity within EGEE had a substantial overlap with LCG operations.

EGEE encompassed a global infrastructure consisting of approximately 200,000 CPU cores, hosted collaboratively by over 300 centers worldwide. Throughout the duration of the project, around 13 million jobs were executed on the EGEE grid every month.

EGEE was completed in 2010 with the establishment of its successor project, the European Grid Infrastructure (EGI). This transition was driven by the need for a more sustainable and operational model for grid infrastructure.

Distributed European Infrastructure for Supercomputing Applications (DEISA, 2004 – 2011) [48] was a project funded by the European Commission to create a European-wide distributed supercomputing infrastructure for research in science and engineering. The project aimed to provide researchers with access to high-performance computing resources and expertise that they otherwise would not have had access to. The project involved 11 European supercomputing centers in seven countries, which were connected by a high-speed network. Prominent scientists throughout Europe are taking advantage of the combined power of supercomputers and corresponding global data management infrastructures in an effective and convenient manner. The emphasis is particularly on grand-challenge applications related to key scientific fields including material sciences, climate research, astrophysics, life sciences, and fusion-oriented energy research.

D-Grid (2004 – 2015) [44] was the German Grid Initiative (DGI) aimed at providing computer infrastructure for education and research in the field of e-Science. The initiative focused on utilizing grid computing technology. It commenced on September 1, 2005, with six community projects, an integration project (DGI), and various partner projects. For example, AstroGrid-D – a joint research project managed by the Astrophysical Institute Potsdam (AIP) focused on integrating German research facilities in astronomy into a unified nationwide infrastructure; C3-Grid – a project aiming to link distributed data archives in climate research; GDI-Grid – a project integrating geospatial data and grid technologies to enhance spatial data infrastructures; HEP-Grid – a project optimizing data analysis in High Energy Physics, Nuclear Physics, and Astroparticle Physics.

Open Science Grid (OSG, 2005 – still ongoing) [72] is a project funded by the National Science Foundation (NSF) that offers researchers distribued high throughput computing. Numerous institutions contribute their computer resources to the OSG, allowing users to run jobs during periods of idle activity. The scale of the OSG's capacity is impressive. By 2021, the OSG provided a staggering 1.1 million core hours, which is equivalent to utilizing 42 thousand cores in a single day [6]. Currently, OSG includes 249 site and 153 institutes distributed all over the world (see Fig. 8)[32].

Figure 8. OSG sites

Each pool within the OSG is structured and managed to cater to specific research communities, such as campuses or multi-institutional collaborations. These pools utilize technologies and services offered by the core OSG Team. One notable pool is the Open Science Pool, which serves the entire US-associated open science community. Consequently, the Consortium encompasses all researchers, resources, individuals, and institutions that either benefit from or contribute to any of the OSG's Fabric of Services (further detailed below).

Currently there are more than 100 scientific projects carried out within the OSG pool of resources. The wast majority of the CPU hours are occupied with physics, chemical science and bioinformatics projects.

Earth System Grid Federation (ESGF, 2006 – still ongoing) [30] is a collaborative set of international interlinked data centers that work together to support climate and Earth system science through data sharing and computational frameworks. ESGF has a global network of distributed data centers, each of which provides infrastructure and resources for scientific data management, access, and sharing. As of early 2022, there are over 40 participating data centers in the ESGF spread across more than 20 countries[33].

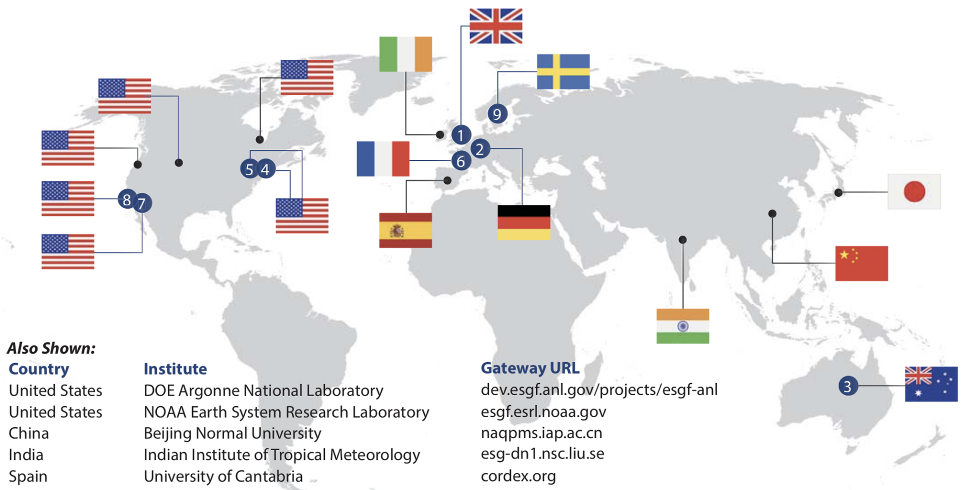

Figure 9. Major federated ESGF worldwide sites [5]

[32]https://map.opensciencegrid.org/map
[33]https://esgf-index1.ceda.ac.uk/projects/esgf-ceda/

Among them, CEDA (Centre for Environmental Data Analysis), based in the UK; DKRZ (Deutsches Klimarechenzentrum), located in Germany; IPSL (Institute Pierre Simon Laplace), a French research institute; LIU (Linkping University), based in Sweden; LLNL (Lawrence Livermore National Laboratory) and ORNL (Oak Ridge National Laboratory) metagrids – US-based metagrids offering data storage and computing resources for scientific research and data analysis in various fields including climate and energy; NCI (National Computational Infrastructure), a data center in Australia [5].

MammoGrid (2009 – 2013) [83] was a collaboration between several European research institutions and universities. The aim of the project was to develop a Europe-wide database of mammograms. MammoGrid was a distributed computing system that aims to improve the accuracy and efficiency of mammography screening. The system used machine learning algorithms to analyze mammograms and provide recommendations to radiologists. MammoGrid used to process large volumes of mammograms at a high speed, enabling healthcare providers to quickly and accurately identify potential breast cancer cases. This project represented an important milestone in the development of eHealth services and distributed computing infrastructure for medical imaging analysis.

European Grid Infrastructure (EGI, 2010 – still ongoing)[34] is a federation of national computing and storage resources across Europe. Its purpose is to foster advanced research,

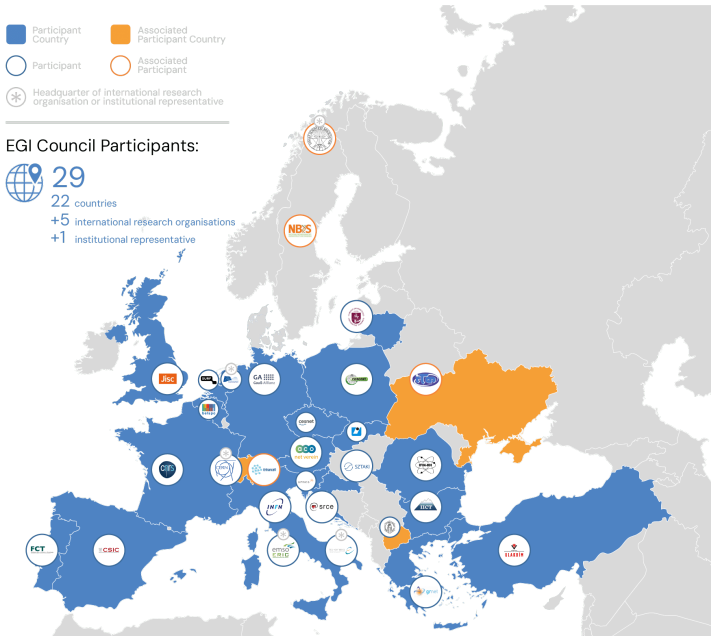

Figure 10. EGI Members

innovation, and knowledge transfer within the continent. EGI builds upon the substantial investments made by national governments and the European Commission over a span of more than a decade.

Currently EGI has 29 participants, 22 represented countries and 300+ represented organisations (see Fig. 10)[35].

EGI brings together a range of resources: a federated cloud infrastructure and HTC Platform (EGI HTC). With over 94,000 users, EGI is a significant player in the field, delivering 80 million

[34] https://www.egi.eu/
[35] https://www.egi.eu/egi-federation/

cloud CPU hours and 7 billion HTC CPU hours in 2023. The platform boasts 580 PB of online storage capacity. The EGI Federation draws support from nearly 300 data centers predominantly in European countries aligned with the EGI Council. It further benefits from resources in Canada, the USA, Latin America, North Africa, and the Asia-Pacific region.

EGI supports scientists in various research disciplines, including high-energy physics, astrophysics, computational chemistry, life sciences, earth sciences, fusion, and many others.

Partnership for Advanced Computing in Europe (PRACE, 2010 – still ongoing)[36] is a collaborative project that was established in 2010 to provide researchers with access to some of the most powerful supercomputers in Europe to carry out large-scale simulations and data analysis tasks in a wide range of scientific domains, including biochemistry, climate modeling, astrophysics, and high-energy physics. The collaboration consists of 26 members (Fig. 11), including most of the major EU nations[37]. PRACE operates a distributed infrastructure that

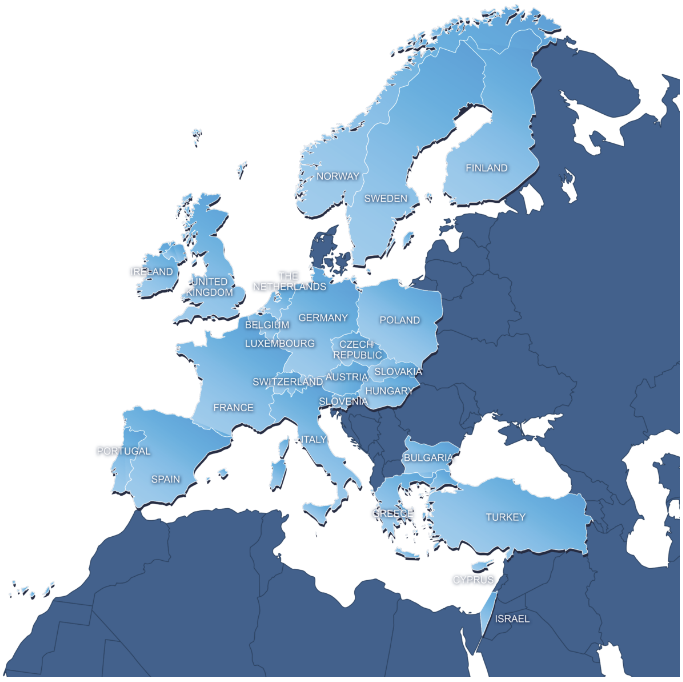

Figure 11. PRACE member countries

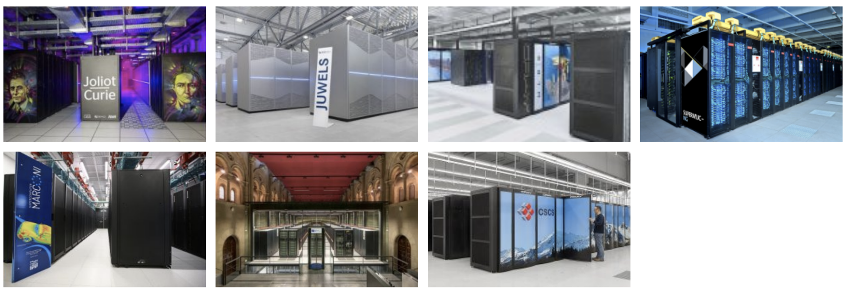

Figure 12. PRACE Tier-0 Centers

consists of a number of Tier-0 supercomputing centers (Fig. 12), located in different countries, that host some of the fastest and most powerful supercomputers in the world, including Joliot-

[36] www.prace-ri.eu

[37] https://prace-ri.eu/about/members/

Curiee in France, JUWELS, HAWK and SuperMUC-NG in Genmany, MARCONI in Italy, MareNostrum 4 in Spain, Piz Daint in Switzerland[38].

The PRACE infrastructure also includes Tier-1 centers that provide additional computing resources and services to support researchers' work.

7. Clouds over Grids

Among all grid applications, the WLCG is the largest and the most long-living grid project. When the WLCG was created, technologies were driven by science and adapted to the needs of Big Science. However, with the rapid technological growth in the last decade, the situation has changed dramatically, and now science is following technology. The same happened with the WLCG, which gradually began to adapt to the use of cloud resources.

Until recently, the computing model at the WLCG was built on the assumption that experiments are the "owners" of computing resources. To be more precise, the computing resources were operated by laboratories and university partners or deployed for a specific program, owned and operated by the host laboratory.

However, HEP experiments revised the initial strategy of utilizing only on-premises resources due to several reasons [53]:

- *Stochastic and bursty nature of computing activities and heterogeneous resources:* The computing activities of the HEP experiments can be categorized into two types: central production activities and analysis activities. Central production is planned and centrally managed, while analysis activities are more chaotic, involving submissions from multiple individuals and being less predictable. In both cases, computing activity occurs in bursts, influenced by the accelerator schedule and advancements in software and detector modernization. Observing a typical data processing for the on-premises resources of HEP experiments demonstrates the stochastic nature of compute demand and the cost implications of provisioning for peak capacity rather than steady-state usage. However, having peak capacity is necessary to perform significant amounts of computing within shorter timeframes, such as for new analyses with discovery potential or conference deadlines.
- *Multiple programs reaching their peak:* By 2025, several HEP programs, including the muon and neutrino programs, and the LHC, will be at the apex of their offline analysis. This stage requires significant computing power and resources.
- *Evolution of the experimental program:* The ongoing upgrades and new experiments in high-energy physics generate a higher demand for computing resources, which surpasses the expected performance gains from advancements in computing techniques and technologies. For example, the High-Luminosity LHC[39] and the Deep Underground Neutrino Experiment (DUNE)[40] are two upcoming programs in HEP that will generate massive amounts of data. The increased precision, event complexity, and luminosity of these programs alone will push computing needs nearly two orders of magnitude higher than current capabilities, generating exabytes of data.
- *The landscape of modern computing resources and the needs for them are dramatically different from the situation 20–30 years ago, when HEP applications were one of the main "consumers" of computing power in the global world.* For example, the LHC at CERN

[38]https://prace-ri.eu/infrastructure-support/prace-hpc-infrastructure/
[39]https://www.home.cern/science/accelerators/high-luminosity-lhc
[40]https://www.dunescience.org/

generates about 50 petabytes per year at WLCG, while there is a large pool of computing resources outside of the HEP: commercial resources (Google, Amazon, Microsoft) and supercomputer centers that are hundreds of times more powerful than the WLCG consortium. Currently, a diverse range of projects are responsible for vast data generation. For instance, social media platforms like Facebook create around 250 petabytes of new data per year. Google's search index alone tracks a minimum of 62 petabytes of data per year. Over 260 petabytes of video content are uploaded to YouTube annually. [32].

Therefore, HEP experiments decided to move towards the usage of cloud technologies that imply on-demand allocation of the resource from a large pool of computing and storage resources, which can be accessed via standard protocols via an abstract interface.

Clouds can be built on top of many existing protocols (WDSL, SOAP), Web 2.0. Technologies (REST, RSS, AJAX) and implemented over existing grids leveraging more than a decade of community efforts in standardization, security, resource management, and virtualization support.

The adoption of cloud computing over grid systems at CERN was a gradual and systematic process, with different commercial cloud solutions being applied to various workflows. Now, let us provide a concise summary of the process involved in establishing cloud infrastructure within WLCG.

OpenStack at CERN Since 2013, CERN has deployed an OpenStack-based private cloud to manage resources across its main data centre in Meyrin, Switzerland, and a remote extension in Budapest, Hungary. The purpose of the use of OpenStack at CERN was to provide resources to scientists and researchers who are conducting experiments related to high-energy physics and particle acceleration. In just its first two years of implementation, the OpenStack cloud tremendously expanded, supporting over 12,000 virtual machines that were spread across 5,500 compute nodes. The usage of this cloud environment was highly dynamic, witnessing an average daily creation and deletion of approximately 3,000 virtual machines. As of 2020, the total number of cores in use has surpassed 300,000, highlighting the remarkable growth and scalability achieved [18].

OpenStack helps CERN to flexibly and efficiently allocate computing resources, reducing management overhead and increasing productivity for the researchers. Additionally, OpenStack enables CERN to better manage cost and capacity planning as it supports the scaling of computing resources which is critical for big science.

OpenStack plays a crucial role in managing and provisioning the majority of CERN's computer center infrastructure, accounting for over 90%. This encompasses various essential functions such as physics processing, storage, databases, and the infrastructure supporting laboratory administration. To ensure robust resource management, accounting accuracy, and effective lifecycle tracking, the remaining hardware within the computer center is currently being integrated into Ironic.

Since 2011, CERN has actively developed and refined OpenStack, with over 1,000 commits. They primarily focused on Magnum, Nova, and Keystone projects. CERN shared their experiences at various events, including over 30 talks at OpenStack summits and regional gatherings like open Infrastructure days. Notably, CERN organized an OpenStack day in 2019 showcasing real-world applications in science, and in 2020, they hosted the Ironic mid-cycle meetup, fostering collaboration within the OpenStack community [37].

European HelixNebula

ATLAS experiment pioneered the usage of cloud computing as a flagship project during the first stages of the European HelixNebula initiative between 2012 and 2014. Helix Nebula is a consortium formed by public research institutions (CERN, EMBL[41] and ESA[42]) and several commercial providers of cloud services (T-Systems, Atos, CloudSigma, The Server Labs, Interoute, etc.) with the aim to make the private resources accessible by research institutions in Europe in a transparent way. During the first proof of concept, this project contributed over 40,000 CPU-days of Monte Carlo production throughput to the ATLAS experiment with marginal manpower required. CERN's experience, together with that of ESA and EMBL, is providing a great insight into the cloud computing industry and highlighted several challenges that are being tackled in order to ease the export of the scientific workloads to the cloud environments [65].

Atos and Microsoft Azure

In March 2015, CERN embarked on their first cloud activity project in collaboration with Atos, a globally renowned company providing information technology, consulting, system integration, and remote management services [34]. Atos, founded on July 1, 2011, is the number one company in Europe in the field of Operational Services and has served as a key IT partner of the International Olympic Committee (IOC) for over two decades. This project involved the provisioning of approximately 3,000 single-core virtual machines (VMs) for the ATLAS collaboration to process simulation jobs over a six-week period. The project demonstrated the ability to incorporate cloud infrastructure-as-a-service (IaaS) within the WLCG workflows to execute central processing unit (CPU) intensive tasks accessing data and software libraries remotely via technologies such as CVMFS and XRootD.

In parallel, Microsoft Azure performed an evaluation by conducting simultaneous deployments in three different data centers, two in the European Union and one in the United States, with collaborations from the ATLAS, CMS, and LHCb submitting simulation jobs.

HEPCloud The Fermilab Scientific Computing Division launched the HEPCloud project in June of 2015 with the objective of creating a comprehensive resource facility with a common interface for a variety of resources including local clusters, grids, high-performance computing, and both community and commercial clouds. The pilot project to assess the feasibility and capability of HEPCloud was initiated in 2016 and executed workflows from the CMS and NOvA experiments. In January of the same year, the project showcased the ability to increase global CMS resources by 58,000 cores from 150,000 cores (a 25% increase rate) to prepare for the Recontres de Moriond, one of the most significant international High Energy Physics (HEP) events [67].

The cost of commercial cloud resources was found to be comparable to on-site resources. This project suggested that steady-state computing costs can become more cost-effective as the cloud computing industry scales, benefiting from potential economies of scale. The flexibility in resource payment provides improved planning and cost efficiency. However, workflows with

[41] European Molecular Biology Laboratory

[42] European Space Agency

intensive data processing might be more suitable for local resources. Therefore, it was shown that adopting a hybrid approach like the HEPCloud facility, which integrates both on-premises and off-premises resources into a unified virtual facility, offers the optimal flexibility to meet the dynamic requirements of the scientific community.

Deutsche Borse Cloud Exchange (DBCE)

Deutsche Borse Cloud Exchange (DBCE) were the vendors for the second cloud procurement in autumn 2016, providing 1,000 4-core VMs running simulation jobs for all LHC collaborations. Although established in May 2013, DBCE was discontinued in 2016 and operated as an international marketplace for the buying and selling of cloud resources. Unlike other commercial cloud providers, DBCE served as a broker to connect customers with IaaS providers in the DBCE marketplace where compute capacity was traded like any other commodity exchange. During this time, DBCE settled a contract agreement among CERN and five different cloud providers, namely ClouData, Cloud&Heat, DARZ, Innovo, and Ultimum [34].

DEUTSCHE BÖRSE CLOUD EXCHANGE

T-Systems and Open Telekom Cloud

T-Systems[43] was allocated a contract for a Pre-Commercial Procurement (PCP) initiated by CERN to design, develop and pilot the Helix Nebula Science Cloud. This joint venture between industry, space, and science aimed to establish an ecosystem with open cloud services that seamlessly integrate science into a business environment. T-Systems and CERN conducted a three-month pilot to assess the capabilities of the Open Telekom Cloud, built on the OpenStack open-source architecture, for data and resource management between private and public clouds. CERN used Open Telekom Cloud to validate whether outsourced commercial cloud providers could process physics data flexibly through the deployment of one thousand virtual machines with associated cluster storage exceeding 500 terabytes.

T Systems

Open Telekom Cloud

Amazon Web Services and ATLAS

In 2018, the ATLAS experiment submitted a request for $250,000 worth of Amazon Web Services (AWS) credits to establish the first virtual Tier 3 cluster on the cloud. The objective was to examine cloud-based solutions for physics analysis and simulation for the ATLAS experiment. The proposal was approved and funded in December 2019, with support from ATLAS and US ATLAS computing management teams. The proposed work, approved by LBNL and University of Texas at Arlington, included setting up ATLAS computing environments for physical analysis frameworks, Monte Carlo simulations, generation and production on the cloud, running sizable analysis jobs, Monte Carlo generation, and simulation for the ATLAS collaboration, and exploring an economic model for future ATLAS computing on the cloud.

aws

Google and ATLAS Projects

The collaborative project between ATLAS and Google was initiated in 2017 to demonstrate a transparent use of commercial cloud resources for scientific experiments [15]. The next phase of the collaboration was launched

Google Cloud

[43] https://www.t-systems.com/

in 2020 [62]. The scope of this project was to integrate Google cloud resources (Storage and Compute) into the ATLAS distributed computing environment in order to enable ATLAS to explore different computing models in preparation for the High-Luminosity LHC, to allow ATLAS users to leverage the Google infrastructure for their analyses, and provide Google with real scientific use cases to enhance their cloud platform. The success of the above projects has triggered an interest and a 15-month ATLAS-Google project [63] was started in May 2022. Recently, long-term projects with Google and Amazon, supported by US ATLAS and California State University Fresno grants respectively were initiated. These projects have gone beyond infrastructure rental, fostering relationships through regular meetings and information exchange [64].

ATLAS has successfully demonstrated the advantage of integration of on-prem and commercial cloud resources as a universal discovery cyberinfrastructure which can be used by large international collaborations. The common perception amongst site administrators is that the cloud is more expensive compared to on-premise options, although cost analysis may be influenced by comparisons to dedicated compute instances rather than spot instances. Previous experiences with spot instances showed eviction rates of up to 15%, which is deemed unacceptable, particularly at multi VO (Virtual Organizations) sites. However, during this project, the observed eviction rate ranged between 1–2%, with 20% of failed jobs resulting from preemptions. Performance variations over time were found to be unpredictable in cloud environments, but the CPU provided by Google remained stable and experienced minimal downtime. The administrators expressed worries about vendor lock-in and emphasized the importance of maintaining flexibility and the ability to switch cloud providers. Notably, the solutions developed in this project are cloud-agnostic and have also been used on AWS at a smaller scale.

The integration of the ATLAS Google site into ATLAS distributed computing has been overwhelmingly successful, showcasing that elastic increase of on-prem resources with a commercial cloud at a large scale requires minimal operational effort. The current ATLAS workflow and data management tools have proven to be adequate for adjusting the configuration of the cloud site. Commercial cloud computing offers a new avenue for HEP computing, providing additional CPU resources on-demand, although network costs may pose a significant factor. Different ATLAS workflows have shown varying degrees of success regarding egress, with derivation production being the most resource-intensive. Resource bursting has been effectively demonstrated but comes at a notable cost. The project has also fostered parallel research and development efforts, allowing for the utilization of diverse resources such as GPUs and ARM CPUs on an elastic basis with seamless integration. The subscription pricing model employed in this project has proven to be suitable for ATLAS; however, future considerations regarding this model remain uncertain. Overall, the project has made significant advancements and laid the groundwork for the continued exploration of cloud-based solutions for ATLAS [81].

8. Grid Applications Ended Up with Greatest Scientific Discoveries

Grid computing, with its ability to connect large-scale computing resources around the world and make them available to researchers and scientists, has played a significant role in some of the greatest scientific discoveries and achievements of the last decade.

Some of the most significant scientific discoveries made possible by grid computing are the Higgs Boson discovery and the detection of gravitational waves.

8.1. Higgs Boson Discovery

In 2012, a team of researchers at CERN used the WLCG to identify the Higgs boson particle, which helps explain the origins of mass in the universe. The discovery was a major milestone in the field of particle physics and led to a Nobel Prize in Physics in 2013 (awarded to Franois Englert and Peter Higgs) [7].

CERN scientists, along with thousands of researchers from around the world, contributed to the discovery of the Higgs boson. Two experiments conducted at the LHC, called ATLAS and CMS, independently announced the discovery on July 4, 2012. These experiments involved analyzing the data from billions of proton-proton collisions to identify the telltale signature of the Higgs boson production.

The Higgs boson is associated with the Higgs field, a field permeating space that interacts with particles, giving them mass. Its discovery confirmed a crucial aspect of the Standard Model and provided experimental evidence for the Higgs mechanism, proposed by Franois Englert and Peter Higgs in the 1960s.

To observe the Higgs boson, the ATLAS and CMS experiments at the LHC required a substantial amount of data accumulation. More specifically, both collaborations analyzed a dataset of around 25–30 petabytes of collision data.

Dr. Fabiola Gianotti (Spokesperson of the ATLAS experiment in 2008/2013, General Director of CERN since 2014) said at a seminar on the discovery of the Higgs boson: *"We are observing a new particle with a mass of about 126 GeV. We would not have been able to process and analyze data so quickly if we had not used the grid. Centers in all countries participating in the experiment were involved in processing LHC data; practically it was a stress test for computing power, and the grid proved to be a highly efficient and reliable system."*[44]

8.2. Gravitational Waves Detection

In 2015, the Laser Interferometer Gravitational-Wave Observatory (LIGO) used grid computing to detect the first direct evidence of gravitational waves, a phenomenon predicted by the Einstein's theory of general relativity nearly a century ago. Gravitational waves are ripples in the fabric of spacetime that travel at the speed of light, carrying information about extremely energetic cosmic events.

On September 14, 2015 (Event GW150914), LIGO made history by detecting the collision of two black holes, an event that emitted gravitational waves. This marked the first direct observation of gravitational waves, opening up an entirely new way to observe and study the universe.

While the scale of gravitational-wave data analysis may no longer be considered as "big data" by current standards, data management remains complex in a distributed HTC environment: LIGO/Virgo generates approximately 20TB of data per interferometer per observing year. For the discovery of gravitational waves in the particular event GW150914, LIGO utilized data spanning approximately 16 days and required 50 million CPU hours for computations[45].

For this groundbreaking achievement, the Nobel Prize in Physics was awarded in 2017 to three key scientists involved in the LIGO project: Rainer Weiss, Barry C. Barish, and Kip S. Thorne.

[44] https://www.mpg.de/5882007/higgs_boson
[45] https://www.ligo.org/detections/GW150914/fact-sheet.pdf

The discovery of gravitational waves opened up a new window of observation, allowing scientists to explore astrophysical phenomena that had previously been invisible. Since then, LIGO and other gravitational wave observatories have continued to detect gravitational waves from various cosmic events, including the merger of neutron stars and black holes. This field of research has furthered our understanding of black holes, neutron stars, and the nature of gravity itself.

Grid computing plays a pivotal role in LIGO's computational infrastructure, enhancing the capabilities of the LIGO Data Grid. In addition to dedicated clusters, LIGO has integrated external resources like the Open Science Grid (OSG) to introduce flexibility and accommodate diverse computational needs. OSG's elasticity allows LIGO to incorporate non-traditional systems alongside its existing infrastructure, catering to both regular workloads and dynamic tasks with efficiency.

Furthermore, LIGO leverages the Berkely Open Infrastructure for Network Computing (BOINC) to harness idle volunteer computers worldwide through projects like Einstein@Home(E@H). This distributed computing model enables LIGO to engage a vast network of distributed resources for computationally intensive tasks, such as Pulsar searches [4].

Conclusion

The following factors have played a crucial role in the development of distributed computing: the challenges of big science, the advancement of data transmission networks, and the development of middleware systems. Grid technologies have made significant scientific discoveries possible on a global scale, such as the discovery of the Higgs boson at CERN and of gravitational waves in the LIGO project. Countless specialists from around the world have contributed their intellectual efforts to the advancement of distributed technologies. Large international grid infrastructures were created, and middleware was developed to ensure the seamless functioning of major scientific projects.

Until the early 2000s, technology followed science, developed technological and software approaches meeting the needs of science. But afterward, when technology reached a new level, scientific projects have begun to adapt to modern technologies by integrating commercial cloud services and containerization into their grid infrastructure.

Despite the success of cloud technologies in both science and business, many existing scientific projects are built on grid infrastructures and will continue to operate for approximately 10 more years. As for new scientific projects currently being developed, some of them may utilize HPC (High-Performance Computing) resources, such as the International Thermonuclear Experimental Reactor (ITER), while others may opt for hybrid systems combining grid, HPC, and cloud technologies, like Nuclotron-based Ion Collider fAcility (NICA) or Deep Underground Neutrino Experiment (DUNE).

Therefore, despite the emergence of more modern technologies, grid computing still maintains its position thanks to the software developed over the past decades, successful application experience, and reliability.

References

1. Gigabit network testbeds. Computer 23(9), 77–80 (1990). https://doi.org/10.1109/2.58220

2. The Gigabit Testbed Initiative. Final Report (1996), http://www.cnri.reston.va.us/gigafr/Gigabit_Final_Rpt.pdf, accessed: 2024-02-16

3. ARC middleware. The NorduGrid Collaboration (2005), http://www.nordugrid.org/documents/whitepaper.pdf, Accessed: 2024-02-16

4. OSG helps LIGO scientists confirm Einstein's unproven theory (2016), https://osg-htc.org/spotlights/osg-helps-ligo-scientists-confirm-einsteins-last-unproven-theory.html, accessed: 2024-04-27

5. 2017 ESGF Brochure (2017), https://esgf.llnl.gov/esgf-media/pdf/2017-ESGF-Brochure.pdf, accessed: 2024-04-29

6. OSPool Usage Hits Daily Record (2021), https://osg-htc.org/news/2021/06/10/OSPool-Hits-Record-Utilization.html, accessed: 2024-02-16

7. Aad, G., *et al.*: Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. Physics Letters B 716(1), 1–29 (2012). https://doi.org/10.1016/j.physletb.2012.08.020

8. Aasi, J., *et al.*: Einstein@Home all-sky search for periodic gravitational waves in LIGO S5 data. Phys. Rev. D 87, 042001 (Feb 2013). https://doi.org/10.1103/PhysRevD.87.042001

9. Abbott, B.P., *et al.*: LIGO: the laser interferometer gravitational-wave observatory. Reports on Progress in Physics 72(7), 076901 (jun 2009). https://doi.org/10.1088/0034-4885/72/7/076901

10. Alexandrov, A., Ibel, M., Schauser, K., *et al.*: SuperWeb: towards a global Web-based parallel computing infrastructure. In: Proceedings 11th International Parallel Processing Symposium. pp. 100–106. IEEE Comput. Soc. Press, Genva, Switzerland (1997). https://doi.org/10.1109/IPPS.1997.580858

11. Anderson, D.P.: BOINC: A Platform for Volunteer Computing. Journal of Grid Computing 18(1), 99122 (Mar 2020). https://doi.org/10.1007/s10723-019-09497-9

12. Anderson, D.P., Cobb, J., Korpela, E., *et al.*: SETI@home: An Experiment in Public-Resource Computing. Commun. ACM 45(11), 56–61 (nov 2002). https://doi.org/10.1145/581571.581573

13. Antonioletti, M., Krause, A., Paton, N.W.: An Outline of the Global Grid Forum Data Access and Integration Service Specifications. In: Pierson, J.M. (ed.) Data Management in Grids. pp. 71–84. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

14. Baratloo, A., Karaul, M., Kedem, Z., *et al.*: Charlotte: Metacomputing on the Web. Future Generation Computer Systems 15(5), 559–570 (1999). https://doi.org/10.1016/S0167-739X(99)00009-6

15. Barisits, Martin, Barreiro, Fernando, *et al.*, B.: The Data Ocean Project – An ATLAS and Google R&D collaboration. EPJ Web Conf. 214, 04020 (2019). https://doi.org/10.1051/epjconf/201921404020

16. Bauerfeld, W.: A tutorial on network gateways and interworking of LANs and WANs. Computer Networks and ISDN Systems 13(3), 187–193 (Jan 1987). https://doi.org/10.1016/0169-7552(87)90025-0

17. Beberg, A.L., Ensign, D.L., Jayachandran, G., *et al.*: Folding@home: Lessons from Eight Years of Volunteer Distributed Computing. In: Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing. pp. 1–8. IPDPS '09, IEEE Computer Society, USA (2009). https://doi.org/10.1109/IPDPS.2009.5160922

18. Bell, T., Bompastor, B., Bukowiec, S., *et al.*: Scaling the cern openstack cloud. Journal of Physics: Conference Series 664, 022003 (12 2015). https://doi.org/10.1088/1742-6596/664/2/022003

19. Berners-Lee, T., Cailliau, R., Luotonen, A.t.: The World-Wide Web. Commun. ACM 37(8), 76–82 (aug 1994). https://doi.org/10.1145/179606.179671

20. Bernstein, P.A.: Middleware: A Model for Distributed System Services. Commun. ACM 39(2), 86–98 (feb 1996). https://doi.org/10.1145/230798.230809

21. Bogdanov, A., et al.: GRID processing and analysis of ALICE data at distributed Russian Tier2 centre: RDIG. J. Phys. Conf. Ser. 219, 072054 (2010). https://doi.org/10.1088/1742-6596/219/7/072054

22. Bonacorsi, D., Ferrari, T.: WLCG Service Challenges and Tiered architecture in the LHC era. In: Montagna, G., Nicrosini, O., Vercesi, V. (eds.) IFAE 2006. pp. 365–368. Springer Milan, Milano (2007). https://doi.org/10.1007/978-88-470-0530-3_68

23. Brecht, T., Sandhu, H., Shan, M., *et al.*: ParaWeb: Towards World-Wide Supercomputing. In: Proceedings of the 7th Workshop on ACM SIGOPS European Workshop: Systems Support for Worldwide Applications. pp. 181–188. EW 7, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/504450.504484

24. Buyya, R., Abramson, D., Giddy, J.: Nimrod/G: an architecture for a resource management and scheduling system in a global computational grid. In: Proceedings Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region. vol. 1, pp. 283–289 (2000). https://doi.org/10.1109/HPC.2000.846563

25. Casajus, A., Graciani, R., Paterson, S., *et al.*: DIRAC pilot framework and the DIRAC Workload Management System. Journal of Physics: Conference Series 219(6), 062049 (Apr 2010). https://doi.org/10.1088/1742-6596/219/6/062049

26. Casanova, H.: Simgrid: a toolkit for the simulation of application scheduling. In: Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid. pp. 430–437 (2001). https://doi.org/10.1109/CCGRID.2001.923223

27. Catlett, C., Allcock, W., Andrews, P., *et al.*: TeraGrid: Analysis of Organization, System Architecture, and Middleware Enabling New Types of Applications, vol. 16, pp. 225–249 (01 2008)

28. Chapin, S.J., Katramatos, D., Karpovich, J., *et al.*: The Legion Resource Management System. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. pp. 162–178. Springer Berlin Heidelberg, Berlin, Heidelberg (1999). https://doi.org/10.1007/3-540-47954-6_9

29. Chiappa, J.N., Steenstrup, M.E., Castineyra, I.M.: The Nimrod Routing Architecture. RFC 1992 (Aug 1996). https://doi.org/10.17487/RFC1992, accessed: 2024-02-16

30. Cinquini, L., *et al.*: The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data. Future Generation Computer Systems 36, 400–417 (Jul 2014). https://doi.org/10.1016/j.future.2013.07.002

31. Clarke, L., Glendinning, I., Hempel, R.: The MPI Message Passing Interface Standard. In: Decker, K.M., Rehmann, R.M. (eds.) Programming Environments for Massively Parallel Distributed Systems. pp. 213–218. Birkhäuser Basel, Basel (1994)

32. Clissa, L., Lassnig, M., Rinaldi, L.: How big is Big Data? A comprehensive survey of data production, storage, and streaming in science and industry. Frontiers in Big Data 6 (2023). https://doi.org/10.3389/fdata.2023.1271639

33. Cole, N., Desell, T., Lombraña González, D., *et al.*: Evolutionary Algorithms on Volunteer Computing Platforms: The MilkyWay@Home Project, pp. 63–90. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-10675-0_4

34. Cordeiro, C., Field, L., *et al.*: CERN Computing in Commercial Clouds. Journal of Physics: Conference Series 898(8), 082030 (oct 2017). https://doi.org/10.1088/1742-6596/898/8/082030

35. DeFanti, T.A., Foster, I., Papka, M.E., *et al.*: Overview of the I-Way: Wide-Area Visual Supercomputing. The International Journal of Supercomputer Applications and High Performance Computing 10(23), 123–131 (Jun 1996). https://doi.org/10.1177/109434209601000201

36. Erwin, D.W., Snelling, D.F.: UNICORE: A Grid Computing Environment. In: Sakellariou, Rizos and Gurd, John and Freeman, Len *et al.* (ed.) Euro-Par 2001 Parallel Processing. pp. 825–834. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)

37. Ferguson, A.: Where are they now? Superuser Awards winner: CERN, https://superuser.openinfra.dev/articles/cern-openstack-update/, accessed: 2024-02-16

38. Filipi, A., Cameron, D., Smirnova, O., *et al.*: The Next Generation ARC Middleware and ATLAS Computing Model. Journal of Physics: Conference Series 396(3), 032039 (dec 2012). https://doi.org/10.1088/1742-6596/396/3/032039

39. Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: Jin, H., Reed, D., Jiang, W. (eds.) Network and Parallel Computing. pp. 2–13. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)

40. Foster, I.: Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. IEEE Internet Computing 15(3), 70–73 (2011). https://doi.org/10.1109/MIC.2011.64

41. Foster, I., Kesselman, C.: Globus: a Metacomputing Infrastructure Toolkit. The International Journal of Supercomputer Applications and High Performance Computing 11(2), 115–128 (1997). https://doi.org/10.1177/109434209701100205

42. Foster, I., Kesselman, C., Tuecke, S.: Chapter 17 – The Open Grid Services Architecture, pp. 215–257. The Morgan Kaufmann Series in Computer Architecture and Design, Morgan Kaufmann, Burlington (Jan 2004). https://doi.org/10.1016/B978-155860933-4/50022-5

43. Foster, Ian and Kesselman, Carl: The History of the Grid 20(21), 22 (2010)

44. Freitag, S., Wieder, P.: The German Grid Initiative D-Grid: Current State and Future Perspectives, pp. 29–52. Springer London, London (2011). https://doi.org/10.1007/978-0-85729-439-5_2

45. Frey, J., Tannenbaum, T., Livny, M., *et al.*: Condor-G: a computation management agent for multi-institutional grids. In: Proceedings 10th IEEE International Symposium on High Performance Distributed Computing. pp. 55–63 (2001). https://doi.org/10.1109/HPDC.2001.945176

46. Gagliardi, F.: The EGEE European Grid Infrastructure Project. In: Daydé, Michel and Dongarra, Jack and Hernández, Vicente and *et al.* (ed.) High Performance Computing for Computational Science – VECPAR 2004. pp. 194–203. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). https://doi.org/10.1007/11403937_16

47. Garonne, V., Vigne, R., Stewart, G., *et al.*: Rucio – The next generation of large scale distributed system for ATLAS Data Management. Journal of Physics: Conference Series 513(4), 042021 (Jun 2014). https://doi.org/10.1088/1742-6596/513/4/042021

48. Gentzsch, W., Girou, D., Kennedy, A., *et al.*: DEISA – Distributed European Infrastructure for Supercomputing Applications. Journal of Grid Computing 9(2), 259–277 (Jun 2011). https://doi.org/10.1007/s10723-011-9183-2

49. Glasser, A.H., Sovinec, C.R., Nebel, R.A., *et al.*: The NIMROD code: a new approach to numerical plasma physics. Plasma Physics and Controlled Fusion 41(3A), A747 (mar 1999). https://doi.org/10.1088/0741-3335/41/3A/067

50. Haber, R.: Scientific visualization and the Rivers Project at the National Center for Supercomputing Applications. Computer 22(8), 84–89 (1989). https://doi.org/10.1109/2.35205

51. Henderson, R.L.: Job scheduling under the Portable Batch System. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. pp. 279–294. Springer Berlin Heidelberg, Berlin, Heidelberg (1995). https://doi.org/10.1007/3-540-60153-8_34

52. Henning, M.: The Rise and Fall of CORBA: Theres a lot we can learn from CORBAs mistakes. Queue 4(5), 28–34 (jun 2006). https://doi.org/10.1145/1142031.1142044

53. Holzman, B., Bauerdick, L.A.T., Bockelman, B., *et al.*: HEPCloud, a New Paradigm for HEP Facilities: CMS Amazon Web Services Investigation. Computing and Software for Big Science 1(1), 1 (Nov 2017). https://doi.org/10.1007/s41781-017-0001-9

54. Jackson, D., Snell, Q., Clement, M.: Core Algorithms of the Maui Scheduler. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. pp. 87–102. Springer Berlin Heidelberg, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-45540-X_6

55. Jackson, T., Austin, J., Fletcher, M., *et al.*: Diagnostics and Prognostics on the Grid: the Distributed Aircraft Maintenance Environment project (DAME) (01 2004)

56. Karsch, F., Simma, H., Yoshie, T.: The International Lattice Data Grid – towards FAIR data. In: Proceedings of The 39th International Symposium on Lattice Field Theory PoS(LATTICE2022). p. 244. Sissa Medialab, Bonn, Germany (Feb 2023). https://doi.org/10.22323/1.430.0244

57. Kónya, B.: Advanced Resource Connector (ARC) – The Grid Middleware of the NorduGrid. In: Kranzlmüller, D., Kacsuk, P., Dongarra, J. (eds.) Recent Advances in Parallel Virtual Machine and Message Passing Interface. p. 10. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)

58. Litzkow, M., Livny, M., Mutka, M.: Condor-a hunter of idle workstations. In: [1988] Proceedings. The 8th International Conference on Distributed. pp. 104–111. IEEE Comput. Soc. Press, San Jose, CA, USA (1988). https://doi.org/10.1109/DCS.1988.12507

59. Lukasik, S.: Why the Arpanet Was Built. IEEE Annals of the History of Computing 33(3), 4–21 (Mar 2011). https://doi.org/10.1109/MAHC.2010.11

60. Maeno, T., De, K., Wenaus, T., *et al.*: Overview of ATLAS PanDA Workload Management. Journal of Physics: Conference Series 331(7), 072024 (Dec 2011). https://doi.org/10.1088/1742-6596/331/7/072024

61. Maffeis, S., Schmidt, D.: Constructing reliable distributed communication systems with CORBA. IEEE Communications Magazine 35(2), 56–60 (1997). https://doi.org/10.1109/35.565656

62. Megino, F.B., Bawa Harinder, S., De, K., *et al.*: Seamless integration of commercial Clouds with ATLAS Distributed Computing. EPJ Web Conf. 251, 02005 (2021). https://doi.org/10.1051/epjconf/202125102005

63. Megino, F.B., De, K., Elmsheuser, J., *et al*: Operational Experience and R&D results using the Google Cloud for High Energy Physics in the ATLAS experiment (2024). https://doi.org/10.48550/arXiv.2403.15873

64. Megino, F.B., *et al.*: Accelerating science: the usage of commercial clouds in ATLAS distributed computing (5 2023), https://indico.jlab.org/event/459/contributions/11636/, accessed: 2024-02-16

65. Megino, F.H.B., Jones, R., Kucharczyk, K., *et al.*: Helix nebula and cern: A symbiotic approach to exploiting commercial clouds. Journal of Physics: Conference Series 513(3), 032067 (jun 2014). https://doi.org/10.1088/1742-6596/513/3/032067

66. Metcalfe, R.M., Boggs, D.R.: Ethernet: distributed packet switching for local computer networks. Communications of the ACM 19(7), 395–404 (Jul 1976). https://doi.org/10.1145/360248.360253

67. Mhashilkar, P., Altunay, M., Berman, E., *et al.*: HEPCloud, an Elastic Hybrid HEP Facility using an Intelligent Decision Support System. EPJ Web of Conferences 214, 03060 (2019). https://doi.org/10.1051/epjconf/201921403060

68. Miura, K.: Overview of Japanese science Grid project NAREGI. Progress in Informatics (3), 67 (Apr 2006). https://doi.org/10.2201/NiiPi.2006.3.7

69. Nisan, N., London, S., Regev, O., *et al.*: Globally distributed computation over the Internet-the POPCORN project. In: Proceedings. 18th International Conference on Distributed Computing Systems (Cat. No.98CB36183). p. 592601. IEEE Comput. Soc, Amsterdam, Netherlands (1998). https://doi.org/10.1109/ICDCS.1998.679836

70. Pickles, S., Brooke, J., Costen, F., *et al.*: Metacomputing across intercontinental networks. Future Generation Computer Systems 17(8), 911–918 (2001). https://doi.org/10.1016/S0167-739X(01)00032-2, high Performance Computing and Networking

71. Pickles, S., Brooke, J., Costen, F., *et al.*: Metacomputing across intercontinental networks. Future Generation Computer Systems 17(8), 911–918 (2001). https://doi.org/10.1016/S0167-739X(01)00032-2, high Performance Computing and Networking

72. Pordes, R., Petravick, D., Kramer, B., *et al.*: The Open Science Grid. Journal of Physics: Conference Series 78(1), 012057 (jul 2007). https://doi.org/10.1088/1742-6596/78/1/012057

73. Resch, M.M., Rantzau, D., Stoy, R.: Metacomputing experience in a transatlantic wide area application test-bed. Future Generation Computer Systems 15(5), 807–816 (1999). https://doi.org/10.1016/S0167-739X(99)00028-X

74. Rutkowski, A.M.: The Integrated Serviced digital Network: Issues and Options for the Future. Jurimetrics 24(1), 19–42 (1983), http://www.jstor.org/stable/29761846

75. Saiz, P., Aphecetche, L., Buni, P., *et al.*: AliEnALICE environment on the GRID. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 502(23), 437–440 (Apr 2003). https://doi.org/10.1016/S0168-9002(03)00462-5

76. Sfiligoi, I.: glideinWMSa generic pilot-based workload management system. Journal of Physics: Conference Series 119(6), 062044 (Jul 2008). https://doi.org/10.1088/1742-6596/119/6/062044

77. Skovira, J., Chan, W., Zhou, H., *et al.*: The EASY – LoadLeveler API project. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. pp. 41–47. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)

78. Smarr, L., Catlett, C.E.: Metacomputing. Commun. ACM 35(6), 44–52 (jun 1992). https://doi.org/10.1145/129888.129890

79. Talia, D.: The Open Grid Services Architecture: where the grid meets the Web. IEEE Internet Computing 6(6), 67–71 (2002). https://doi.org/10.1109/MIC.2002.1067739

80. Thain, D., Tannenbaum, T., Livny, M.: Distributed computing in practice: the Condor experience. Concurrency and Computation: Practice and Experience 17(2-4), 323–356 (2005). https://doi.org/10.1002/cpe.938

81. The ATLAS Collaboration: Total cost of ownership and evaluation of Google cloud resources for the ATLAS experiment at the LHC (2024). https://doi.org/10.48550/arXiv.2405.13695

82. Ďurech, J., Hanu, J., Vano, R.: Asteroids@homeA BOINC distributed computing project for asteroid shape reconstruction. Astronomy and Computing 13, 80–84 (2015). https://doi.org/10.1016/j.ascom.2015.09.004

83. Warren, R., *et al.*: MammoGrid – a prototype distributed mammographic database for Europe. Clinical Radiology 62(11), 1044–1051 (2007). https://doi.org/10.1016/j.crad.2006.09.032

84. Xu, M.: Effective metacomputing using LSF Multicluster. In: Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid. pp. 100–105 (2001). https://doi.org/10.1109/CCGRID.2001.923181