Recent Developments in High Performance Computing for Remote Sensing: A Review

Craig A. Lee, *Member, IEEE*, Samuel D. Gasster, *Senior Member, IEEE*, Antonio Plaza, *Senior Member, IEEE*, Chein-I Chang, *Fellow, IEEE*, and Bormin Huang

Abstract—Remote sensing data have become very widespread in recent years, and the exploitation of this technology has gone from developments mainly conducted by government intelligence agencies to those carried out by general users and companies. There is a great deal more to remote sensing data than meets the eye, and extracting that information turns out to be a major computational challenge. For this purpose, high performance computing (HPC) infrastructure such as clusters, distributed networks or specialized hardware devices provide important architectural developments to accelerate the computations related with information extraction in remote sensing. In this paper, we review recent advances in HPC applied to remote sensing problems; in particular, the HPC-based paradigms included in this review comprise multiprocessor systems, large-scale and heterogeneous networks of computers, grid and cloud computing environments, and hardware systems such as field programmable gate arrays (FPGAs) and graphics processing units (GPUs). Combined, these parts deliver a snapshot of the state-of-the-art and most recent developments in those areas, and offer a thoughtful perspective of the potential and emerging challenges of applying HPC paradigms to remote sensing problems.

Index Terms—Cloud, distributed computing infrastructures (DCIs), field programmable gate arrays (FPGAs), graphics processing units (GPUs), grids, high performance computing (HPC), multiprocessor systems, remote sensing applications, service architectures, specialized hardware architectures.

I. INTRODUCTION

ADVANCES in sensor and computer technology are revolutionizing the way remotely sensed data is collected, managed and analyzed [1]–[4]. In particular, the incorporation of latest-generation sensors to different platforms for Earth and planetary observation is currently producing a nearly continual stream of high-dimensional data, and this explosion in the amount of collected information has rapidly created new processing challenges [5]. The development of computationally efficient techniques for transforming the massive amount of remote sensing data into scientific understanding is critical for Earth science [6]. The rate of increase in the volume of remote sensing data continues to grow, as the number of organizations and users of these data which conform a world-wide community now demanding efficient mechanisms to share these data and resources. To address the aforementioned needs, several research efforts have been recently directed towards the incorporation of high-performance computing (HPC) techniques and practices into remote sensing missions [7]. HPC comprises a set of integrated computing environments and programming techniques which can greatly assist in the task of solving large-scale problems such as those involved in many remote sensing studies. For instance, many current and future applications of remote sensing in Earth science, space science, and soon in exploration science require real- or near real-time processing capabilities [8]. Relevant examples include environmental studies, military applications, tracking and monitoring of hazards such as wild land and forest fires, oil spills and other types of chemical/biological contamination [9].

The utilization of HPC systems in remote sensing applications has become more and more widespread in recent years. The idea (originally developed by the computer science community) of using commercial off-the-shelf (COTS) computer equipment [10], [11], clustered together to work as a computational "team," has inspired many developments based on exploiting multi-processor systems [12]–[15]. Although most parallel techniques and systems for image information during the last decade have chiefly been homogeneous in nature (i.e., they are made up of identical processing units, thus largely simplifying the design of parallel solutions adapted to those systems), a recent trend in the design of HPC systems for data-intensive problems is to utilize highly heterogeneous computing resources [16], [17]. This heterogeneity is seldom planned, arising mainly as a result of technology evolution over time and computer market sales and trends. In this regard, distributed networks of heterogeneous COTS resources can realize a very high level of aggregate performance in remote sensing applications [18], and the pervasive availability of these resources has resulted in the current notion of grid computing [19], [20] and its evolution, cloud computing, which both endeavor to make such heterogeneous and distributed computing platforms easy to use. Such systems currently represent a tool of choice for efficient

distribution and management of very high-dimensional data sets in remote sensing and other fields.

Finally, although remote sensing data processing algorithms generally map quite nicely to multi-processor systems made up of clusters or networks of CPUs, these systems are generally expensive and difficult to adapt to onboard remote sensing data processing scenarios, in which low-weight and low-power integrated components are essential to reduce mission payload and obtain analysis results in real-time, i.e., at the same time as the data is collected by the sensor. In this regard, the emergence of specialized hardware devices such as field programmable gate arrays (FPGAs) [21] or graphic processing units (GPUs) [22] exhibit the potential to bridge the gap towards onboard and real-time analysis of remote sensing data. The increasing computational demands of remote sensing applications can now benefit from these compact hardware components, taking advantage of the small size and relatively low cost of these units as compared to clusters or networks of computers. These aspects are of great importance in the definition of remote sensing missions, in which the payload is an important parameter.

In this review paper, we specifically focus on describing recent advances in the field of HPC applied to remote sensing problems, covering developments using different architectures such as clusters, grids, clouds and specialized hardware components. The remainder of the paper is organized following the general order of increasing system building block size in HPC. Specifically, Section II first describes systems and architectures for onboard processing of remote sensing data using specialized hardware such as FPGAs and GPUs. Section III includes a compendium of algorithms and techniques for HPC-based remote sensing data processing using clusters, from traditional systems such as Beowulf clusters to modern systems based on multi-core processors and GPUs. Section IV focuses on parallel techniques for remote sensing data interpretation using large-scale distributed platforms, with special emphasis on grid and cloud computing environments. Section V provides a summary and general discussion, and further anticipates future directions and challenges in the application of HPC-based systems to remote sensing problems. Section VI concludes the paper with some remarks.

II. SPECIALIZED HARDWARE ARCHITECTURES

In this section we describe several recent research efforts which have been directed towards the incorporation of specialized hardware for accelerating remote sensing-related calculations on-board airborne and satellite sensor platforms. Enabling on-board data processing introduces many advantages, such as the possibility to reduce the data down-link bandwidth requirements at the sensor by both preprocessing data and selecting data to be transmitted based upon predetermined content-based criteria. On-board processing, as a solution, allows for a good reutilization of expensive hardware resources. Furthermore, it allows making autonomous decisions on-board that can potentially reduce the delay between the image capture, analysis and the related action. Implementations of on-board processing algorithms to perform data reduction can dramatically reduce data transmission rates. On-board

processing also reduces the cost and the complexity of ground processing systems so that they can be affordable to a larger community. Among the remote sensing applications that will most greatly benefit from this kind of processing developments are not only Earth observation missions which are now considering the inclusion of specialized hardware components, but also future web sensors missions and planetary exploration missions, for which hardware developments would enable autonomous decisions to be taken on-board. Specifically, in this section we focus on recent advances based on two types of platforms: GPUs and FPGAs. An exhaustive comparison of both types of platforms has been recently presented in the framework of remotely sensed hyperspectral image processing in [23].

A. Application of GPUs to Remote Sensing Problems

In recent years GPUs have evolved into a highly parallel, multithreaded, many-core processors with tremendous computational speed and very high memory bandwidth [24]. The combined features of general-purpose supercomputing, high parallelism, high memory bandwidth, low cost, and compact size are what make a GPU-based desktop computer an appealing alternative to a massively parallel system made up of commodity CPUs. The advent of NVidia CUDA, an extension to the C programming language offering programming capabilities of GPU's in general-purpose fashion (GPGPU), has introduced the possibility of including GPUs in many science and engineering applications. The exploding GPU capability has attracted more and more scientists and engineers to use it as a cost-effective high-performance computing platform, including scientists in remote sensing sensing areas.

Despite the very recent emergence of GPGPU in scientific applications, several relevant efforts oriented towards GPU-based processing of remote sensing data sets can already be found in the literature. Among several others [7]–[9], we outline the recent presentation of a GPU-based synthetic aperture radar (SAR) simulation system in [25] or the description of a GPU system for the computation of ray-traced troposphere delays which can be utilized for space geodetic applications [26]. A Fortran to CUDA compiler was used in [27] to parallelize the dynamics portion of a weather model and achieved $34\times$ speedup. A GPU implementation of a linear prediction method using constant coefficients for hyperspectral images was described in [28]. A GPU-based implementation of an automated morphological algorithm for pure spectral signature (endmember) extraction from remotely sensed hyperspectral data sets was described in [29], [30], whereas $15\times$ speedup for the hyperspectral pixel purity index endmember extraction and unmixing algorithm was reported in [31]. Several GPU implementations have been proposed to significantly accelerate the radiative transfer model by 2 orders of magnitude for the Infrared Atmospheric Sounding Interferometer (IASI) onboard the first European meteorological polar-orbiting satellite in 2006 [32], [33]. The GPU-based channel and source decoding system for China's Chang'e II lunar satellite launched in October 2010 achieved $87\times$ speedup in [34]. A GPU-accelerated lossless ultraspectral compression system for NASA New Millennium Program's Geosynchronous Imaging Fourier

Transform Spectrometer (GIFTS) was also reported in [35]. A GPU-based Low-density parity-check (LDPC) decoder with $271\times$ speedup for data transmission over error-prone wireless links was presented in [36]. Additional efforts towards real-time and on-board target detection in remotely sensed data sets are also reported in [37], although radiation-tolerance and power consumption issues for these hardware devices should be explored in future developments in order to allow their full incorporation to spaceborne Earth observation missions.

B. Application of FPGAs to Remote Sensing Problems

Recently, FPGA-based computing, also known as reconfigurable computing, has become a viable target for the implementation of algorithms suited to remote sensing applications. FPGAs represent an evolution over application-specific integrated circuits (ASICs) [38], in the sense that ASICs are designed specifically to solve a given problem. However, an ASIC circuit cannot be modified after fabrication as opposed to FPGAs. As a result, reconfigurable hardware introduces a trade-off between traditional hardware and software by achieving hardware-like performance with software-like flexibility, which is of great interest in the context of remote sensing applications [39]. Specifically, when the data are high-dimensional, FPGAs offer the possibility of performing on-board, real-time processing [40]–[44]. On the one hand, FPGAs are now fully reconfigurable, which allows one to adaptively select a data processing algorithm (out of a pool of available ones) to be applied onboard the sensor from a control station on Earth [39]. For instance, recently developed (hybrid) FPGAs such as the Xilinx Virtex-4FX60 and Virtex-5 not only include a larger hardware area to implement custom accelerators but also embedded processors and memory resources. This option offers versatility in running remote sensing applications on embedded processors [43], while taking advantage of reconfigurable hardware resources, all on the same chip package. These tightly coupled hardware/software co-designed systems [39] combine the flexibility of traditional microprocessors with the power and performance of custom hardware implementations, leading to new architectures for remote sensing missions. With reconfigurable hardware it is possible to apply much of the flexibility that was formally restricted to software developments only, to highly parallel hardware resources. The idea is that FPGAs can be reconfigured on the fly. This approach is called *temporal partitioning* or *run-time reconfiguration* [45]. Basically the FPGA (or a region of the FPGA) executes a series of tasks one after another by reconfiguring itself between tasks [38]. The reconfiguration process updates the functionality implemented in the FPGA, and a new task can then be executed. This time-multiplexing approach allows for the reduction of hardware components on-board since one single reconfigurable module can substitute several hardware peripherals carrying out different functions during different phases of the mission.

Moreover, satellite-based remote sensing instruments can only include chips that have been certified for space operation. This is because space-based systems must operate in an environment in which radiation effects have an adverse impact on integrated circuit operation.[1] Ionizing radiation can cause soft-errors in the static cells used to hold the configuration data. This will affect the circuit functionality and ultimately result in system failure. This requires special FPGAs that provide on-chip reconfiguration error-detection and/or correction circuitry. High-speed, radiation-hardened FPGA chips with million gate densities have recently emerged to support the high throughput requirements for remote sensing applications [42]. In fact, radiation-hardened FPGAs are in great demand for military and space applications. For instance, industrial partners such as Actel Corporation[2] or Xilinx[3] have been producing radiation-tolerant anti-fuse FPGAs for several years, intended for high-reliability space-flight systems. Actel FPGAs have been on-board more than 100 launches and Xilinx FPGAs have been used in more than 50 missions.

III. CLUSTER COMPUTING

This section first provides an overview of the evolution of cluster computing architectures in the context of remote sensing applications, from initial developments in so-called Beowulf systems at NASA centers to the current clusters regularly employed for remote sensing data processing. Then, an overview of recent developments in architectures using multiple processing cores (including clusters based on hardware accelerators such as GPUs) is given. These systems are soon becoming a standard in the application of HPC techniques to remote sensing and other problems.

A. Cluster Computing in Remote Sensing: Evolution

Remote sensing fostered the development of cluster computing [10] as a cost-effective parallel computing system able to satisfy specific computational requirements, since the Earth and space sciences community initially adopted this solution as the standard for parallel processing in this particular context [2]. As sensor instruments for Earth observation incorporated more sophisticated capabilities for improved data acquisition, it was soon recognized that desktop computers could not provide sufficient power for effectively processing this kind of data [11].

The first efforts targeted towards the exploitation of clusters in the remote sensing community were carried out in the mid-nineties, when a team was put together at NASA's Goddard Space Flight Center (GSFC) in Maryland to build a cluster consisting only of commodity hardware (PCs) working in cooperative fashion, which resulted in the first Beowulf cluster [11]. It consisted of 16 identical PCs with central processing units (CPUs) working at clock frequency of 100 MHz, connected with two hub-based Ethernet networks tied together with channel bonding software so that the two networks acted like one network running at twice the speed. The next year Beowulf-II, a 16-PC cluster based on 100 MHz Pentium PCs, was built and performed about 3 times faster, but also demonstrated a much higher reliability. In 1996, Pentium-Pro cluster at California Institute of Technology (Caltech) demonstrated a sustained per-

---

[1]http://www.xilinx.com/publications/prod_mktg/virtex5qv-product-table.pdf

[2]http://www.actel.com

[3]http://www.xilinx.com

Fig. 1. Thunderhead Beowulf cluster at NASA's Goddard Space Flight Center in Maryland.

formance of one Gigaflop on a remote sensing application. This was the first time a commodity cluster had shown high performance potential in this context.

Up until 1997, clusters were in essence engineering prototypes, that is, they were built by those who were going to use them. However, in 1997 a project was started at GSFC to build a commodity cluster that was intended to be used by those who had not built it, the HIVE (highly parallel virtual environment) project [13]. The idea was to have workstations distributed among different locations and a large number of compute nodes (the compute core) concentrated in one area. The workstations would share the compute core as though it was apart of each. Although the original HIVE only had one workstation, many users were able to access it from their own workstations over the Internet. The HIVE was also the first commodity cluster to exceed a sustained peak performance of 10 Gigaflops on a remote sensing data processing. Later, an evolution of the HIVE was used at GSFC for remote sensing applications. The system, called Thunderhead (see Fig. 1), was a 512-processor homogeneous cluster composed of 256 dual 2.4 GHz Intel Xeon nodes, each with 1 Gigabyte of memory and 80 Gigabytes of main memory.[4] The total peak performance of the system was 2457.6 Gigaflops. Along with the 512-processor computer core, Thunderhead has several nodes attached to the core with 2 GHz Myrinet network connectivity. This system has been employed in several remote sensing studies over the last few years [14], [40], [46], [47].

It is worth noting that NASA and the European Space Agency (ESA) are currently supporting additional massively parallel clusters for remote sensing applications, such as the Columbia supercomputer[5] at NASA Ames Research Center, a 10,240-CPU SGI Altix supercluster, with Intel Itanium 2 processors, 20 terabytes total memory and heterogeneous interconnects including InfiniBand network and 10 gigabit Ethernet. Another massively parallel system which has been

actively exploited for remote sensing applications is MareNostrum,[6] an IBM cluster with 10,240 GPUs, 2.3 GHz Myrinet network connectivity and 20,480 GB of main memory available at Barcelona Supercomputing Center [48]. Finally, the High Performance Computing Collaboratory (HPC[2]) at Mississippi State University[7] has several supercomputing facilities that have been used in different remote sensing studies. Resulting from the efforts and new developments conducted in the area of cluster computing applied to remote sensing, many remote sensing data processing applications have increased their computational performance in a significant way [6], [12]–[14], [18], [40].

B. Recent Developments: Multi-Core Platforms and Clusters Based on Hardware Accelerators

In the past, much of the innovation in CPUs had previously focused on increasing the number of clock cycles in a single core. Nowadays most computers are equipped with multi-core CPUs, which are more power efficient than machines with multiple CPUs. These platforms have already been adopted in a number of remote sensing applications [8], [49]. Even though several remote sensing applications map nicely to parallel systems made up of multi-core CPUs, these systems may not cope with extremely high computational requirements in many remote sensing applications [8]. In this regard, an exciting new development in the field of commodity computing is the emergence of cluster systems with hardware accelerators. An important hardware component to consider in this context is the GPU, which until recently, had traditionally been limited to perform graphical operations. GPU hardware improvements and increased accessibility to software packages such as NVidia[8] CUDA (Compute Unified Device Architecture)[9] has generated tremendous interest in using the GPU for scientific computing. In addition, GPUs can also significantly increase the computational power of cluster-based systems (e.g., the fastest supercomputers in the world are now clusters of GPUs).[10]

Although clusters based on other types of accelerators such as FPGAs also exist [41], these platforms are more intended towards onboard processing due to their capacity to adaptively select a data processing algorithm (out of a pool of available ones) to be applied on-board the sensor from a control station on Earth [42]. Quite opposite, the emergence of GPUs (driven by the ever-growing demands of the video-game industry) has allowed these systems to evolve from expensive application-specific units into highly parallel and programmable commodity components that can be exploited as co-processors in the framework of cluster computing [29]. For instance, the latest-generation GPU architectures from NVidia (*Tesla* and *Fermi* series) now offer cards able to deliver up to 515 Gigaflops of double-precision peak performance,[11] which is several times the performance of the fastest quad-core processor available. The

ever-growing computational demands of remote sensing applications are now taking advantage from the most recent developments in HPC and cluster computing, including the advent of GPU clusters (or clusters with other special *add-ons* and hardware accelerators) and the gradual increase in the number of cores for each cluster node. Today, most newly installed cluster systems have such special-purpose extensions and/or many-core nodes (even up to 48 cores, such as AMD's Magny-Cours).[12] In the near future, these systems may introduce significant advances in the way remotely sensed data sets are processed, stored and managed.

IV. Distributed Computing Infrastructures

The field of distributed computing has been evolving over the last forty years, ever since the advent of computer networks. This evolution has been especially rapid since the advent of the World Wide Web. This particular Distributed Computing Infrastructure (DCI) transformed society's understanding, use and dependence on DCIs for all manner of activities, starting in academia and science, but rapidly expanding to include commerce, entertainment, and government. With this rapid development of uses and implementations, it should not be surprising that a confusing number of names and "buzzwords" have been used for different DCI implementations, in different application domains, by different user groups and industries.

Hence, prior to reviewing DCIs for remote sensing, we review the fundamental concepts and capabilities necessary to create and manage DCIs. Besides sorting out the plethora of names used for various distributed computing technologies, this will also enable us to place these technologies in a spectrum—from very simple technologies that individual researchers may be able to deploy and use, to very large, complex systems that are being deployed and used by international organizations. Once we have done this, we can review existing DCIs for remote sensing within the context of a notional reference architecture for satellite ground systems. We then identify fundamental challenge areas for DCIs of the future.

A. Distributed Computing Terminology

The field of distributed computing contains a rich, but sometimes confusing, vocabulary. Before we begin our review of how this field contributes to Earth remote sensing we would like discuss this terminology, how the terms are related and what they imply about the capabilities and benefits of distributed computing systems and infrastructure. Our summary of this terminology based on the top-down perspective as shown in Fig. 2. *Systems* (and systems of systems) are designed using a variety *architectures* (architectural types) and these designs can be implemented using different *frameworks* and *enabling technologies*. We define and describe each of these concepts and how they relate to one another. We also identify the key characteristics of each of these concepts as it relates to the overall goal of designing, building and operating a distributed computing system to support Earth science applications and remote sensing.
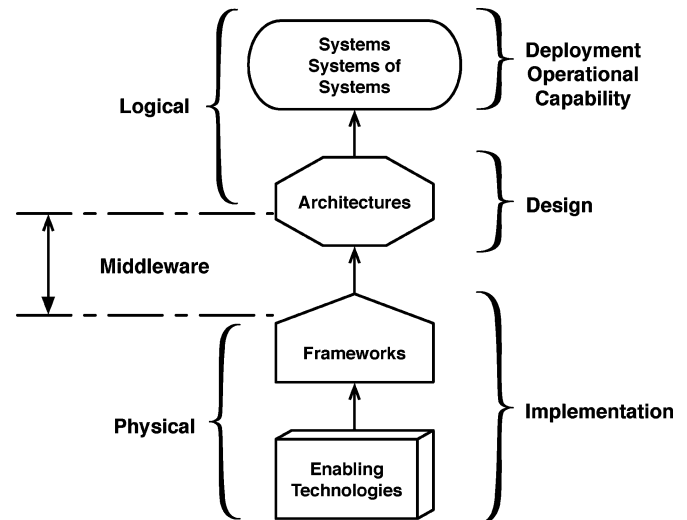
Fig. 2. Relationships between system, architectural, framework and enabling technology concepts.

A system is a collection of elements that are combined in order to achieve a goal or results not achievable by the elements alone [50]. Systems are created based on a set of goals and objectives. By composing multiple systems, based on a new set of goals, one can create a system of systems (SoS). Systems are designed using architectural concepts and types and it is the system that is deployed to provide the users with an operational capability to support their goals.

A virtual organization (VO) is a system of systems composed of different entities that have separate administrative domains,[13] and are physical distributed from one another. The VO is a construct that allows these multiple organizations to jointly manage resources and corresponding user authorizations to these resources. Individual members of a VO have a role that determines what functions they can perform, such read/write of data and and what services they can create and execute. Organizational members of a VO, i.e., the owners of local data and services, can determine the authorization privileges associated with individual roles for those local resources. By providing authorizations through the use of roles, rather than individual users, results in a much more scalable system.

An architecture is the set of structures needed to reason about the system, which comprises the elements that make up the system, relationships among them, and properties of both. The architecture must be capable of describing both software and hardware elements, and their interfaces. The architecture provides a logical description of what elements comprise the system, the context for the system, and the interactions between the elements necessary to achieve the goals or purpose of the system. An architectural description is at the logical functional level (the "what"), and does not provide any information regarding specific implementation (the "how"). A systems engineer must map the system architecture to a specific implementation using frameworks and enabling technologies.

This is illustrated in Fig. 2. The logical requirements for a desired operational capability is expressed as an architectural design. The physical instantiation or implementation of a logical design is realized through the use of appropriate middleware that provides the necessary set of basic functional capabilities. These middleware capabilities are expressed as models or frameworks that make the enabling technologies easy to use.

The different architectural types emphasize different high level system objectives. For example, *cloud computing* is about provisioning of computing resources (providing resources on-demand) and *grid computing* is about federation (distributing the physical resources but providing a single logical interface within the system). The *service oriented architecture (SOA)* style is about providing services and the associated policies, protocols, interfaces and communication infrastructure to allow access to, and utilization of, these services. A service has a well defined function that is self-contained and does not depend on the context or state of other services. One example of a SOA is the *sensor web* in which the services include raw data services provided by sensors (service provider) and various types of data consumer users (data consumer services). The data consumer services can utilize the raw data (NASA Level 0 data) provided by the sensors for a variety of purposes, including the creation of various other data products, which are in turn made available as a data service (e.g., NASA Level 1–3 data products).

Frameworks provide the basic structure or abstraction underlying a system design that provides the building blocks to construct the system or application. In this survey we are emphasizing software frameworks, but one can also reason about hardware frameworks as well. Frameworks provide a set of libraries or classes as the fundamental building blocks and a set of rules or instructions regarding composition through well defined interfaces and data. Frameworks provide flow control, default behavior, extensibility, an other constructs necessary to implement the design. The frameworks are used to implement the middleware and as such also provide the "glue" that binds the physical layer (specific implementation technologies and underlying hardware) with the logical layer. Examples include the Globus Grid toolkit and the Grid Data Farm (Gfarm) open source distributed file system. The Apache Hadoop framework enables the execution of applications on large computer cluster systems through the implementation of the Map/Reduce computational paradigm [51].

Enabling technologies are the underlying components (hardware and software) and protocols that allow one to implement the frameworks and libraries that express a given architectural type. Examples include web services (SOAP), HTTP/HTTPS, and network protocols (TCP, IP), commodity based hardware to build up clusters, high speed fiber optic networks, etc.

The term *distributed computing infrastructure* (DCI) refers to the collection of logical, physical and organization elements needed for the creation and operation of a distributed system. These systems may be both logically and physically distributed and an objective of most implementations of DCI is to make this distinction totally transparent to the user through the concept of virtualization (discussed below).

In the remaining sections we will use the term "data" in several different contexts. This term can refer to the Earth observation data (spanning the range from raw sensor data to highly processed products such a global sea surface temperature maps), it can also refer to the information passed as part of the management and administration of the DCI. Rather than continuously qualify the term we hope the meaning will be clear from the context. There is also the metadata ("data about the data") and when speaking about this we will explicitly use the term metadata.

B. Capabilities, Scaling and Benefits of DCI

1) Capabilities: Remote sensing systems must have specific capabilities to achieve the overall system goals, e.g., presentation of calibrated and geolocated satellite imagery, but they must also include a set of common capabilities to manage and operate their infrastructure. The functionality defined to support both the user and administrative capabilities must be supported by the frameworks and underlying technologies. One of the most important capabilities provided by DCI is the separation of the logical organization and functionality from the physical (virtualization). This separation frees the user and applications from managing resources and infrastructure, allowing them to focus on their specific workflow.

Examples of the types of capabilities necessary for DCI are summarized below. This discussion considers both human users as well as application clients (application programming interfaces). To simplify the discussion we will use the term client to refer to both human and application clients. While we have listed each of these capabilities separately there is almost always a significant need for these capabilities to interoperate and support one another.

- **Resource discovery and catalogues**—Within the DCI resources need to be discoverable. A resource is generally considered to be any type of data or service. In order for users to find these resources, they need to be cataloged in searchable databases with well defined interfaces and query languages. The catalogs, along with the associated metadata and query syntax, allow clients to discover and access resources based on a logical identity. The query can return links or mappings from the logical identity to one or more possible physical entities that the client can access.

- **Data interoperability**—This involves the addressing of potentially different data storage, management approaches and implementations across different administrative domains (a simple example being the different data formats used in different cloud computing infrastructure provided by different vendors). This capability is fundamental for data sharing and access across domains. The capability will have to address semantic interoperability, data translations and transformations, data provenance and security between systems. Addressing this capability will require advances in both tools and policies and will need to be implemented in a manner that is transparent to the users [52], [53].

- **Service/Job/Workflow Management**—The orchestration of different resources will require the capability to manage requests for services, job execution and workflows as specified by various clients. This management capability needs

to provide mechanisms for the allocation of resources to handle service requests, instantiate services, prioritize service requests, and respond to service level agreements (these may involve specifying timeliness requirements on execution and job completion) [54], [55].

- **Resource instantiation and provisioning (allocation)**—In addressing this capability the different perspectives of grid and cloud computing come together. In many cases, it may not be desirable to dedicate resources to one application but rather provide on-demand resource allocation and provisioning as the requests arise. This will require the capability to balance resource supply with demands. This type of provisioning may require models to predict possible surges in demand and methods to access additional resources only when they are needed. The resource allocation problem is very complex and an active area of research to find optimal approaches based on different constraints and conditions [56]–[58].

- **Monitoring**—This capability addresses several different dimensions of DCI operation. Basic system availability and reliability must be maintained to provide the services and resources to clients when they are needed. Not only are tools needed to monitor resources within a given administrative domain, but there must be sharing of this information across domains (federation of the DCI management data). Resource faults and failures should be monitored and reported to avoid service outages. Clients will require a minimum quality of service for many tasks or have in place specific service level agreements (SLA). The capability to monitor resource and service performance is therefore required. These administrative monitoring capabilities allow the various DCI system administrators to observe the overall health and status of their local domains and the system as a whole. Finally, the ability to monitor overall system security is essential given the threat landscape that exists and continues to expand [59]–[61].

- **Event Notification**—This capability is essential to enable asynchronous communications among the different elements of a DCI based system. Event notifications are disseminated for various purposes in DCI applications, such as logging, monitoring and auditing, and a variety of other events which involve a change in state of a resource or service. Possible events include computation results, status updates, errors and exceptions, and progress for execution of client workflow [62], [63].

- **Security**—We cannot overstate the importance of security, or information assurance, as a foundational capability for any DCI system. Just about every aspect of the system operation will involve the security capabilities. This is necessary in order to provide the clients assurance with respect to their data integrity and analysis results. The basic components include mechanisms for client and process authentication and authorization. These capabilities must allow complex cross domain operation, for example, single sign-on, while maintaining system security. Information integrity is the capability to protect against unauthorized modification or destruction of information. Given the importance of Earth observation data for many national and global policy debates, data integrity and provenance are of critical importance [64], [65].

- **Accounting and Auditing**—Given that many of the resources used to construct the DCI will come from a variety of sources, include commercial entities, it will be necessary to track resource utilization for fee-based services. Internal to any system, auditing tools can track usage patterns to determine where additional resources maybe required or where under utilized resources exist [66].

2) Performance Scaling: An important aspect of the DCI approach is the ability to scale the system in response to changing system requirements and resource demands. To assess these changing needs it is necessary to quantify system performance over many different types of performance parameters or scaling dimensions. The scaling dimensions relate to the performance of the different services and resources provided by the system. For example, latency is often a key performance parameter since many systems have requirements for near-real-time support, as in the case of disaster management. In such systems latency requirements might be imposed on arrival time of data from sensors, or computational results from forecast models. The system must be capable of provisioning the network bandwidth and compute power to support latency requirements, especially during those times when resource demands are changing. In a SOA, the time required to complete various service requests is an interesting measure of performance since it can depend on a variety of factors, such as inherent reporting and sampling capabilities/limitations of a sensor service, resource limitations for a particular service, utilization and bandwidth. Systems are generally designed with specific performance requirements that may also include possible growth in demand over time. Number of clients is also an important consideration, which can range from small work groups ($\leq$10 users) to large scale VOs ($\sim$1000 users).

3) Benefits: There are many benefits to employing the DCI approach for Earth observation and remote sensing applications. The use of well defined architectural concepts and types, and implementing these using standards-based frameworks and enabling technologies has several benefits. This notion of virtualization of sensors underlies the concept of a sensor web mentioned previously.

One of the primary benefits is the concept of virtualization. We previously mentioned the concept of a virtual organization, but the notion of virtualization is fundamental to how the middleware shown in Fig. 2 provides the interface between the physical implementation and the logical user interface. The goal is to free users from the management of the resources needed to achieve their workflow needs and allow them to focus on their scientific studies and analyses. Users are able to logically discover and access data or computing resources and combine these as part of their workflow without concern for their physical implementation. The ultimate goal, of course, is to provide these capabilities on-demand and meet the user requirements for performance and timeliness. This concept of virtualization of resources can be applied to any type of resources, from computing infrastructure (CPUs, storage, bandwidth), data sources such as sensors and instrumentation. Sensors can be virtualized so that both remote sensing capabilities and in-situ measurements are
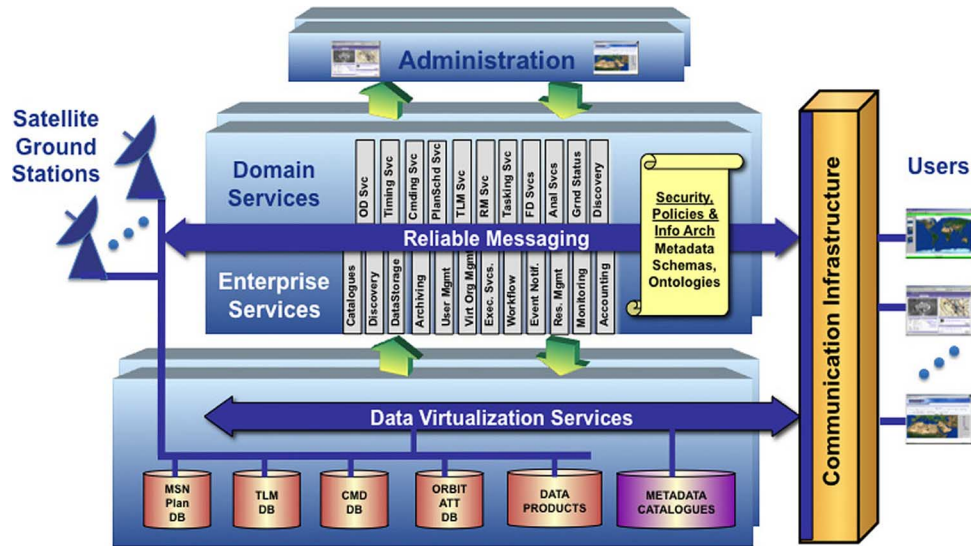
Fig. 3. A notional satellite ground system reference architecture.

made available as a service. Users can specify their data needs using "natural" syntax and semantics that the system can then translate into specific workflow to fulfill this data request (e.g., a user might specify a space-time bounding box, with spatial-temporal and spectral resolution and sampling requirements, and the system then determines which sensor can best satisfy this request).

Virtualization also allows users can search and discover measurements and observational data based on the metadata characteristics specific to their analysis. This might involve searching using a spatial (geographical) and temporal bounding box, sampling characteristics (spatial and temporal) and measurement or geophysical parameter. Another benefit of virtualization is the ability to provide upgradability with little or no impact to availability.

Another benefit of DCI is the interoperability achieved through the use of architectural and implementation standards for protocols and interfaces, such as those provided by a SOA. These standards support benefits such a composability and extensibility that provide the infrastructure to meet the performance requirements previously described. Additional attributes such as reuse and rapid deployment are also important benefits of the DCI approach. As will be discussed later in this section, many Earth observation applications are event triggered, and do not require continuous availability of system resources. The capability to marshall all the necessary resources on-demand based on the trigger event, for example, a natural disaster such as a hurricane or earthquake, is important. Such a system can maintain readiness with limited resource utilization until the full capability is required. Thus, these resources are not sitting idle, but rather could be used by other applications and only accessed in time of need, using a priority based utilization scheme (disaster events receive higher priority than routine scientific studies).

An additional benefit of using the service architectural concepts, combined with the right frameworks and technologies for implementation, is the ability to create what have become

known as Problem Solving Environment (PSE) [67]. A PSE is designed to provide the framework to target a specific class of problems within a given scientific domain. The framework provides the tools in the natural language of the specific scientific discipline so that the user can marshal these resources with very little learning curve. The framework can encapsulate very powerful data processing and analysis capabilities coupled to the underlying computing and data resources in a manner that is transparent to the user [68]–[70].

C. A Notional Ground System Reference Architecture for Remote Sensing

With this clear understanding of fundamental DCI capabilities and terminology, we now take a systematic approach to identify the required capabilities to manage the acquisition of data from both on-orbit and terrestrial sensors, the production of the resulting data products, their use by a large distributed user community, and the management of the DCI enterprise, as a whole. To this end, we present a notional satellite ground system reference architecture, shown in Fig. 3, that is cast as a services architecture where administrator and user access is through browser-based tools. A key aspect of this reference architecture is that services are broadly categorized into *domain services* and *enterprise services*. Domain services include those that are specific to the management of satellite systems (and possibly other sensor systems), e.g., command and control, orbital determination, tasking/planning/scheduling (for on-orbit resources), telemetry, etc. The specifics of these services are not the focus of this paper, so we do not discuss them any further.

Germane to this discussion are the enterprise services, providing the capabilities to use and manage all other aspects of the infrastructure. Clearly there are services for cataloguing and discovering resources, i.e., for data and services. Execution and workflow services manage the lifetime of individual service instantiations and also sequences of service executions and any

necessary data transfers. Monitoring services, event notification, and accounting services are used to evaluate system performance, monitor for faults, and to accumulate audit trails. All of these services could be dynamically allocated from a pool of resources, i.e., a cloud, as part of resource management.

A number of other services cut across all other aspects of the infrastructure. These include reliable messaging, security, and governance policies that must be enforced throughout the user environment. We note that all of these services could, in fact, be distributed across different sites. Reliable messaging simply means that if message delivery fails, an error condition is guaranteed to be raised. That is to say, there are no silent communication failures.

The security mechanisms and infrastructure provide support for integrity and privacy for both data at rest (on disk) and in flight (on the network). Checksums and other methods are used to ensure integrity. What most users must deal with directly, however, is authentication and authorization. In distributed environments, user account management requires *federated identity management* and *virtual organization management*. Federated identity management enables different organizations to trust one another's users. Virtual organizations (VOs) provide a mechanism whereby *role-based authorization* can be enforced based on a user's identity and role with the VO which may can span multiple administrative domains. VOs can also be used to manage common data and collaboration tools for the VO's members.

Enterprise governance is typically expressed through policies. These policies can be enforced either by administrators or automatically by the system. Usage policy can be largely addressed through role-based authorization for users. There can also be system management policies that determine how much processing time is available for given tasks, data replication among sites, etc.

Broken out separately here are the *data virtualization services*. The data produced by on-orbit sensors must be collected, calibrated, catalogued, archived, and made available to authorized users. Databases will be used to maintain operational data, such as mission planning, telemetry, commanding, and orbital attitude data. Most users, however, will be focused on the data products and their metadata. These catalogues can be massive and distributed. Data must also be archived indefinitely. Hence, it is far better for data to be virtualized, such that it can be accessed by their attributes, rather than requiring users to know anything about data physical location, storage format, etc. This requires that an *information architecture* be in place that defines metadata schemas and ontologies. With data virtualized in an information architecture, data provenance, the understanding of data evolution, and long-term data preservation all become easier to accomplish.

Furthermore, we note that virtualization of data also facilitates the *virtualization of the sensors*. Accessing data by attribute can be applied to data that *will be* produced by a sensor network, as well as to data that has already been collected and archived. This approach provides a clean, logical interface for requesting, acquiring, and using sensor data that isolates the user from the immense technical detail of operating a remote sensor system. Many of these issues are discussed in more detail in the context of geospatial data in [71].

D. Survey of Key DCI Examples

This reference architecture now gives us a context in which to discuss and evaluate a number of key DCI examples that involve remote sensing. These examples represent different aspects of the reference architecture, and cover the spectrum from simple, lightweight implementation approaches typically used by small research teams, to large, enterprise-scale systems that can only be deployed by large institutions.

1) The Matsu Project: The goal of the Matsu Project [72] is to provide an on-demand (cloud-based) disaster assessment capability through satellite image comparisons. This is an on-going research collaboration under the umbrella of the Open Cloud Consortium (OCC) [73] whose participants include the NASA GFSC, the University of Illinois at Chicago, the Starlight optical network, and others. OCC operates a distributed cloud infrastructure hosted by OCC members, and participants in the Large Data Working Group. This infrastructure provides a Eucalyptus-based cloud with over 300 cores, 80 TB of storage, and 10 Gbps network connections (being upgraded to 80 Gbps) using network equipment provided by Cisco.

The initial processing scenario for the Matsu project is a flood prediction and assessment capability being applied to Namibia. Fig. 4 illustrates Matsu's general workflow and *flood dashboard*. Relying on relatively simple Web 2.0 mash-ups, Matsu implements a sensor web that collects sensor data from a variety of sources, including six Namibian river stations. Matsu also ingests data from on-line sources, such as the Global Disaster Alert and Coordination System [74], and the on-line daily flood masks produced by the NASA MODIS data processing center.

Most importantly, though, Matsu users can propose tasking for the Hyperion and ALI sensors on the EO-1 satellite to collect hyperspectral images over regions of interest. The tasking operation is managed by the Geospatial Business Process Management System. Once collected, the images are radiometrically and geometrically corrected and stored on the OCC cloud. Image comparisons can be done to assess flooding using Hadoop. The final data is served to end-users using the standard OGC Web Map Service and the Web Coverage Processing Service tools.

2) GENESI-DR and GENESI-DEC: The initial goal of the Ground European Network for Earth Science Interoperations—Digital Repositories project (GENESI-DR) was to provide a large, distributed data infrastructure for world-wide community needs [75]. This was a Seventh European Framework project funded from 2008 to 2010. Its follow-on project, GENESI-DEC—Digital Earth Communities—runs until 2012 with the goal of enhancing support to specific user communities and other existing data archives [76].

Through a simple web portal and web services API, users can register their data sets and make them available to other Earth scientists. The cataloguing of disparate data sets was a major challenge that GENESI-DR addressed by basing their core metadata properties on the metadata rules for INSPIRE, the Infrastructure for Spatial Information in Europe [77]. This was represented as a Resource Description Framework model that reuses common vocabularies. OpenSearch [78] was integrated enabling GENESI-DR to support both geospatial and temporal
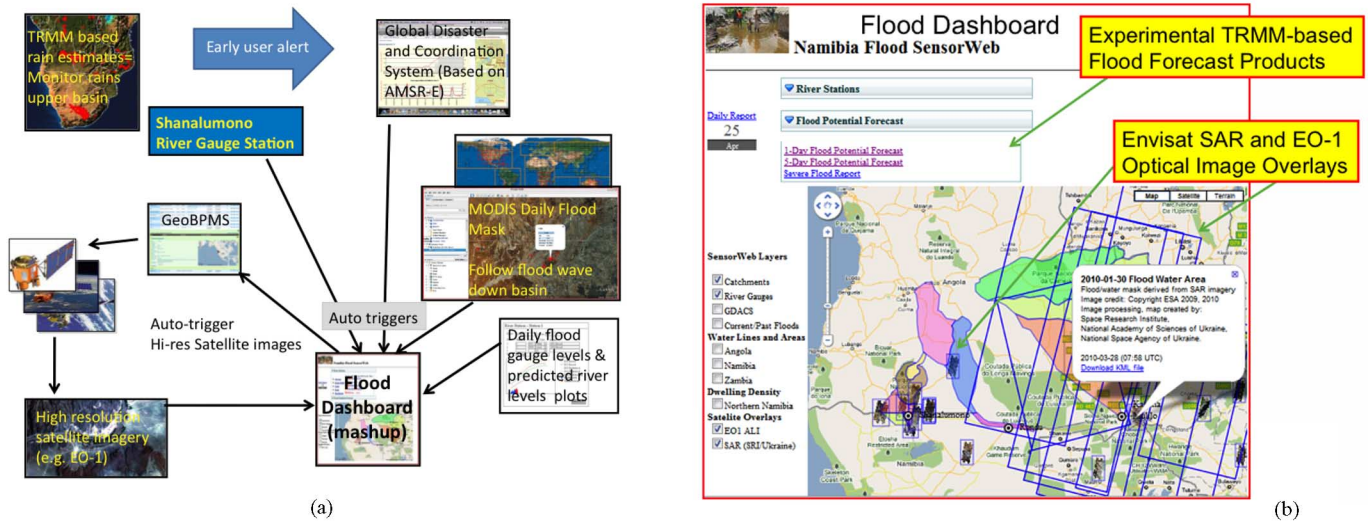
Fig. 4. The Project Matsu general workflow (left) and flood dashboard (right).

search queries based on either free text or specific metadata parameters. By the end of GENESI-DR, more than twelve European sites and more than fifty data sets (including satellite data sets) were available.

Once this basic project infrastructure was in place, it became clear that there was a need for an authorization model that observed different intellectual property rights as specified by the data owners. Another issue was providing single sign-on across multiple digital repositories operated in different administrative domains, to support cross-site workflows. The GENESI-DEC project addressed of these issues by using OpenID[14] with a virtual organization approach. Besides working with multiple data communities, such as the Ocean Observation community, and Global Atmosphere Observation community, GENESI-DEC is part of an alliance to promote a common data approach for the GEOSS Common Infrastructure. (See Subsection 6.)

3) G-POD: The goal of the Grid Processing on Demand (G-POD) project is to provide on-demand processing for Earth observation data [79]. G-POD was started by the European Space Agency in 2002 using a grid architecture, but has since incorporated a cloud approach.

G-POD provides a portal whereby users can search a catalog for available data products. Desired data sets can be accessed through PUT, GET, LIST, and DELETE commands. These include data sets from the ERS-1 and ERS-2 satellites, and the Envisat ASAR and MERIS sensors. The portal provides services whereby the user can use various tools and algorithms to process these data sets from Level 0, unprocessed sensor data with communication artifacts removed, to Level 3, radiometrically and geometrically calibrated geophysical variables mapped on to a uniform space-time frame of reference. Once processing jobs are started, they can be managed and monitored for progress, e.g., queued, running, completed, etc.

G-POD was initially constructed using the Globus grid toolkit [80]. Behind the easy-to-use portal interface, G-POD used GridFTP to transfer data sets and GRAM to submit jobs

on pre-configured computing resources. Despite the traditional batch workflow management, G-POD provided an *on-demand* processing utility.

Subsequently, Terradue Srl was selected by ESA to commercialize and enhance G-POD [81]. As a result of their work, G-POD now has the capability to acquire and release Amazon EC2 computing nodes and S3 storage blocks, without requiring any significant changes to the user interface. That is to say, the portal will move Earth observation data in and out of S3 storage, and run the same services as EC2 instances, while presenting the same interface to the user. This is a prime example of the fact that clouds are first and foremost about the *provisioning* of resources. Fig. 5 illustrates a G-POD services page. Besides the portal, G-POD services are available through HTTP and SOAP. G-POD users are authenticated using PKI certificates issued by the G-POD administrators.

4) GMSEC: The goal of the GFSC Mission Services Evolution Center (GMSEC) project is to enable the rapid prototyping of satellite ground systems [82]. This effort has been underway at GFSC for many years and is gaining acceptance as a viable method to deploy ground systems. The GMSEC architecture is based on the message bus concept, as illustrated in Fig. 6. The critical design element for GMSEC is the carefully controlled API between GMSEC-compliant components and the underlying message bus. This allows rapid integration of new and existing components, from a library of components, to support the rapid deployment of a satellite ground system. Besides basic components to support plotting and other display functions, there are high-level components for managing satellite flight dynamics, planning and scheduling, telemetry, archiving, and system monitoring. Approximately sixty components are available and this number continues to grow.

By controlling the API between components and the message bus, a variety of underlying message busses can actually be supported. Hence, besides using a message bus produced at GFSC, other commercial messaging systems can be used, including Tibco SmartSockets, Tibco Rendezvous, Interface Control System's Software Bus, and the Elvin distributed

---

[14]OpenID is an open standard that describes how users can be authenticated in a decentralized manner. See http://openid.net/.
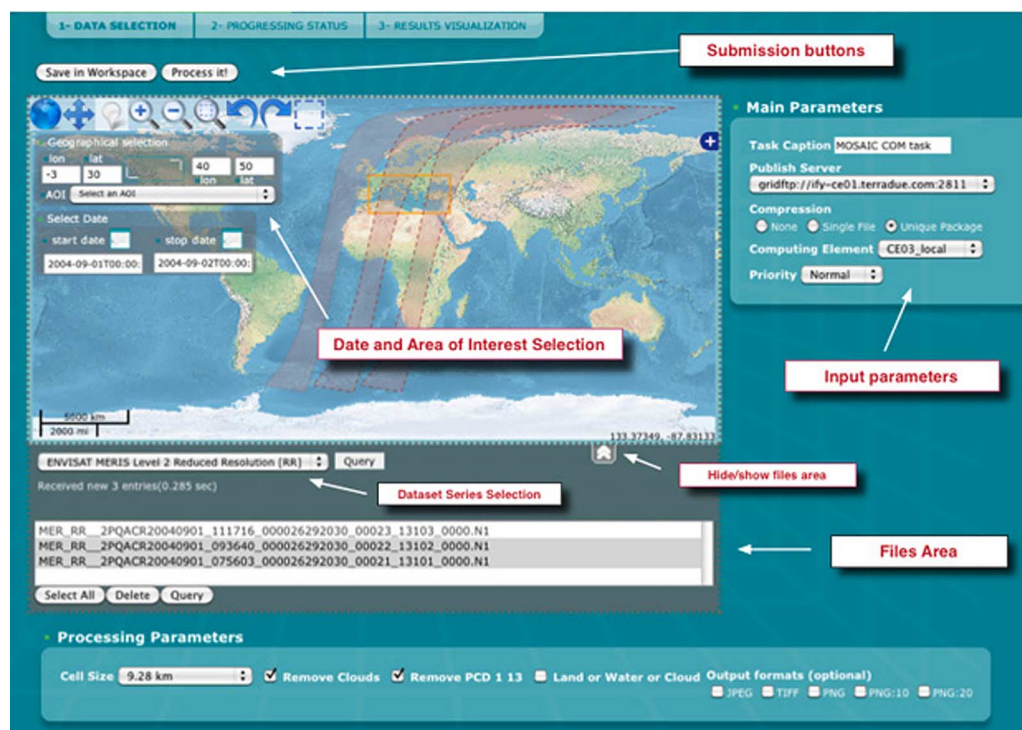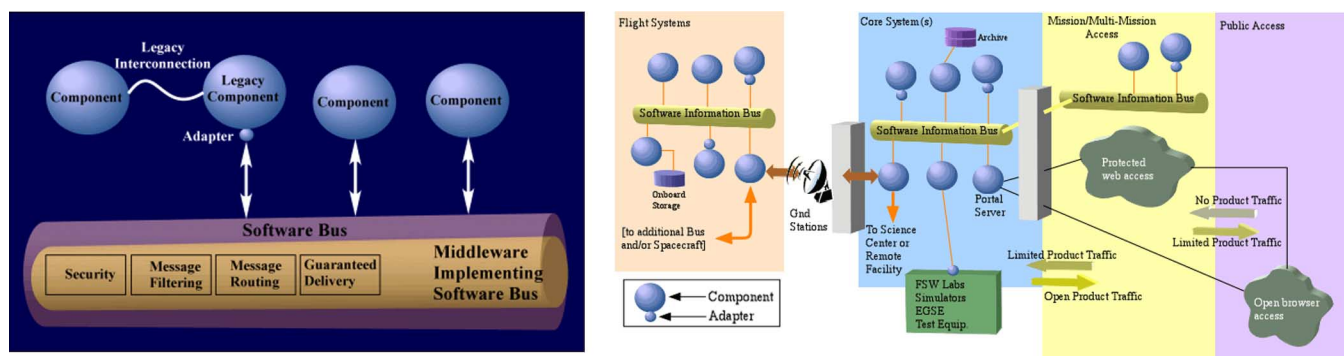
Fig. 5. A G-POD services page.

Fig. 6. The GMSEC message bus concept (left) and the GMSEC reference architecture (right).

event routing service. By doing so, GMSEC can rely on the capabilities provided by the commercial messaging systems, such as publish-subscribe, guaranteed delivery, and security.

The GMSEC Reference Architecture describes how the message bus concept can be deployed across a distributed environment, including on-orbit platforms. Different message bus instances are deployed in different environments, separated by firewalls, or in the case of on-orbit instances, by the actual ground link. All data presented to external web servers must also go through a portal server component where data dissemination policies are enforced. The portal server, and the firewall, also control any traffic back into the message bus system. The use of a GMSEC segment on-board a satellite was demonstrated in 2006 in the ST5 micro-satellite constellation mission that measured the Earth's magnetic field.

5) GEO Grid: The goal of GEO Grid is to provide a disaster assessment capability, and could be considered a prototype for an operational disaster monitoring system. GEO Grid integrates grid technology to securely manage federated resources with standard geospatial tools for a variety of applications focusing on utilization of various remote sensing data sources. GEO Grid is an on-going project funded by the Japanese government and built by the Grid Technology Research Center at the National Institute of Advanced Information Science Technology, Japan GEO Grid ingests both ASTER and MODIS data, storing this data using the Gfarm data grid middleware [83] to achieve the desired scalability and distribution. Like many other systems, GEO Grid can be accessed through a portal. However, GEO Grid offers both a *portal development kit (PDK)* and a *service development kit (SDK)*. The GEO Grid PDK enables users to build customized portals from a library of portlets that include workflow engines, data access tools, and OGC web services. The GEO Grid SDK enables users to create their own services that can be registered and shared with other users and sites. Many of these services are based on the widely adopted OGC services for serving geospatial data, e.g., WMS, WFS, WCS, etc. [84].
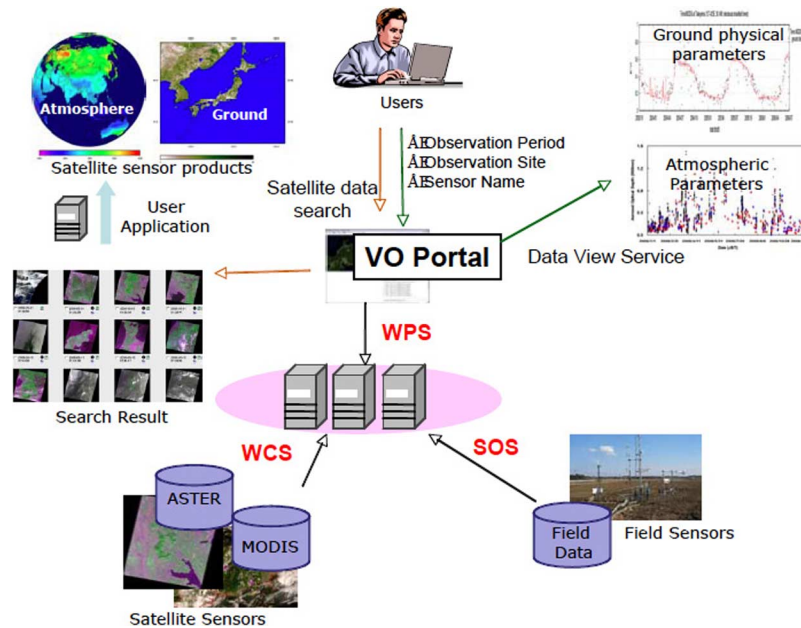
Fig. 7.   The GEO grid field observation network virtual organization.

GEO Grid uses the Grid Security Infrastructure (GSI) in conjunction with the VO concept to realize a scalable authorization mechanism for different groups of users. GEO Grid currently operates VOs for Geological Hazards and for "Business, IT and GIS". To illustrate the capabilities of GEO Grid consider the Field Observation Network (FON) virtual organization [85] that is design to support the calibration and validation of on-orbit sensors, through the comparison of on-orbit data with other data sources, e.g., ground-based observations. As illustrated in Fig. 7, the FON VO integrates data from a network of ground observatories that capture digital fisheye camera data, hemispherical spectro-radiometer data, and sunphotometer data. FON VO manages these terrestrial sensors using the OGC Sensor Observation Service (SOS) standard. Using GEO Grid portal services, users can evaluate the accuracy and properties of the on-orbit sensors.

6) GEOSS: The goal of the Global Earth Observation System of Systems (GEOSS) is to deploy an international, federated infrastructure for sharing of Earth observation data products worldwide. This is in support of nine areas of societal benefit: disaster management, health, energy, climate, water, weather, ecosystems, agriculture, and biodiversity [86]. The GEOSS project is managed by the Group on Earth Observations (GEO) [87], an international collaboration of many organizations that produce and consume Earth observation data. GEO has defined a ten-year implementation plan for GEOSS running from 2005 to 2015.

The current 2009–2011 work plan targets building an integrated GEOSS Common Infrastructure (GCI). This is being accomplished though the GEOSS Architecture Implementation Pilot (Task AR-09-01b). Fig. 8 illustrates the design of the GCI as a service-oriented architecture. A set of registries are used for service components, user requirements, and standards for interoperability. Here user groups from around the world can register their data sets and services. To facilitate resource dis-

covery, the GEOSS Clearinghouse does global GEOSS searches based on the registered metadata for any type of resource, e.g., systems, services, data, documents, or specific file types. All of these system components are accessed through a portal[15] using either free text input, browsing the areas of societal benefit, or selecting locations on an interactive globe.

As member of GEO, the Committee on Earth Observation Satellites (CEOS) [88], is provide the space segment for this project and thus the data catalogued in these registries. CEOS members operate the satellite programs that are producing this data on a regular basis.

To support GEOSS, CEOS has developed the notion of *virtual constellations*, whereby satellites and ground segments operated by one or more organizations can be managed in a coordinated way to meet overall Earth observation requirements. To accomplish this, GEO and CEOS maintain a set of joint CEOS-GEO actions as part of the GEOSS ten-year plan and the current 2009–2011 work plan. This includes virtual constellations (Task AR-09-02a) and data sharing principles (Task DA-06-01), in addition to support for specific societal benefit areas, such as global agricultural monitoring (Task AG-07-03a).

These efforts are being supported by organizations around the world. For example, Kopernikus, formerly the Global Monitoring for Environment and Security (GMES) project, is the European contribution to GEOSS [89]. Kopernikus/GMES catalogues a tremendous amount of remote sensing data from satellites such as EU Envisat. These data sets are being made available through the EuroGEOSS Broker, whose initial phase maintains 400 data sets and 26 services.

The United States Group on Earth Observations (USGEO) [90] has the charter from the National Science and Technology Council to establish an integrated Earth Observation system, in conjunction with GEO. USGEO has the membership of seventeen federal agencies who are dedicated to the open sharing of
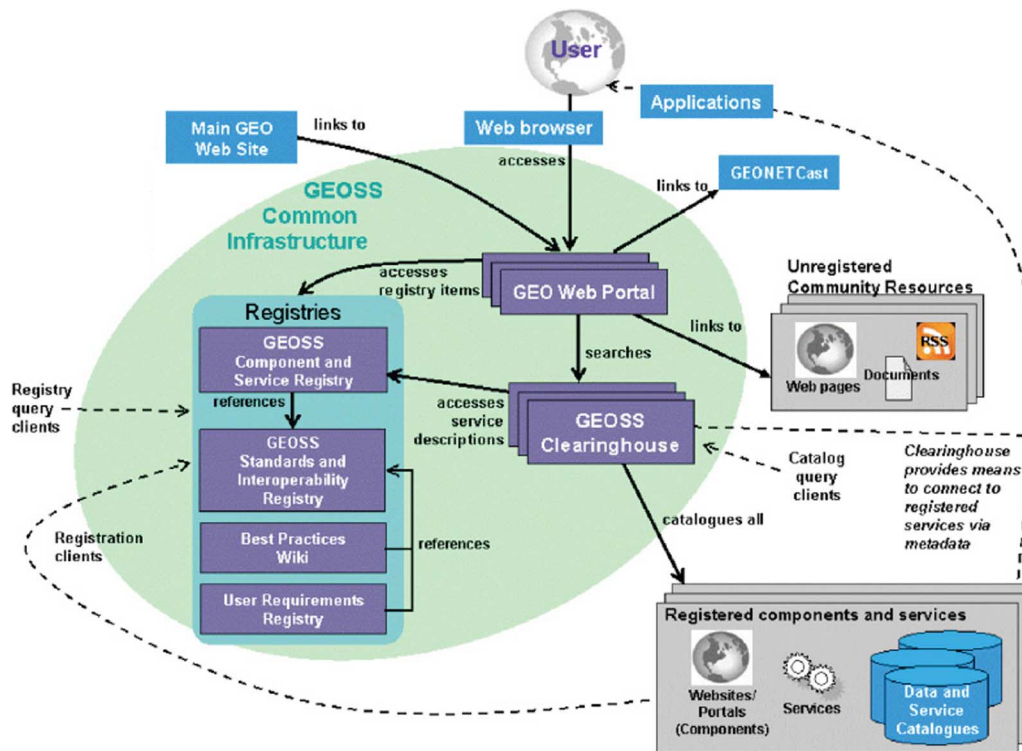
[15]http://www.geoportal.org.

Fig. 8.　The GEOSS common infrastructure.

data. USGEO hosts GEOSS registries and has made available the entire Landsat archive consisting of 2.4 million images from over 37 years.

The Japanese Aerospace Exploration Agency (JAXA) also hosts GEOSS registries to make available the data from many of its Earth Observation satellites. In addition to observing land vegetation, etc., the Advanced Land Observing Satellite DAICHI is also being used for disaster management, e.g., landslides. The Greenhouse Gases Observing Satellite IBUKI (GOSAT) has two sensors for measuring the global distribution of CO_2 and CH_4. The Tropical Rainfall Measuring Mission (TRMM) satellite is a joint project between JAXA and NASA with five sensors on-board dedicated to rainfall observation. The data from these sensors is processed in both countries and is used, in conjunction with other remote sensing sources, to produce a global rainfall map every hour, about four hours after observation.

E. Survey Discussion

Using the six DCI examples above, we can easily identify commonality and trends for supporting Earth remote sensing. Using these insights, we examine how the various elements of the notional satellite reference architecture have been addressed.

All these systems address security, but to varying degrees according to their requirements. There are many aspects to security, but for the purposes of this review, we will focus on *authentication* and *authorization*—establishing *who* you are and *what* you are allowed to do. The Matsu Project uses very simple password authentication since it is an experimental project being implemented as a Web 2.0 mash-up and has modest security

requirements. Several of these systems use passwords, in addition to traditional physical security means, as in GMSEC. To ease the burden of having to authenticate to multiple web sites, existing standards are used, such as OpenID and OAuth,[16] with GENESI-DEC having adopted OpenID. These tools allow the same credentials to be used when authenticating to various sites, thus reducing "password fatigue." This can be further ameliorated by *single sign-on* where a user authenticates once and then has access to multiple systems. There are many single sign-on implementations, which are commonly based on two-factor authentication with one-time passwords. The most advanced security infrastructures are employed in G-POD and GEO Grid, which use the X.500 certificate-based Grid Security Infrastructure. Here, user certificates are issued by a *certificate authority* and establish identity on all transactions. GEO Grid goes a step further by implementing virtual organizations, enabling the management of federated resources from different administrative and organizational domains. To enable truly worldwide VOs, the International Grid Trust Federation certifies the operation of certificate authorities, thereby enabling organizations to trust certificates issues by other IGTF members.

It should not be surprising that all of these systems address data and data access, essentially providing data virtualization services. Most widely used are the OGC standards for cataloguing and serving data. The Catalogue Service for Web (CSW) has had an ebRIM profile defined giving it a metamodel that can handle services, symbol libraries, coordinate reference systems, application profiles, and schemas, as well as geospatial data. CSW also supports catalogue federation that is critical for distributed systems. CSW 3.0 will incorporate functionality

[16]http://oauth.net/.

from OpenSearch [78] which supports both geospatial and temporal queries. Much of this resulted from work done as part of the GENESI-DR and GENESI-DEC projects. There is even work being done with CSW/ebRIM to provide a data provenance service for geospatial data [91]. Finally, work is being done with CSW and ISO/IEC 14863 System Independent Data Format [92] to address long-term data preservation issues, e.g., isolation from media formats to enable 100–500-year data retention [93].

With regards to the human-computer interface and the actual presentation of geospatial data, the web browser is widely adopted. Besides simply serving data through web pages, web portals that collate data from multiple sources are widely used. In the case of GEO Grid, a portal toolkit allows further extension and customization of portal capabilities, depending on user requirements. We note that the basic data service standards of the Open Geospatial Consortium are established and widely used, i.e., the Web Map Service, the Web Feature Service, and the Web Coverage Service. While none of the systems reviewed explicitly mention Google Earth, the use of Google Earth and the Keyhole Markup Language (KML) have found rapid adoption given the robust display and navigation features of Google Earth and the ease of using KML. There are browser plug-ins for Google Earth, along with tools, such as OpenLayers, to integrate map data into web pages using the standard OGC tools.

Job management and workflow management are clearly critical functions for all DCIs, with the details typically hidden from end-users behind a browser or portal. This allows the users to focus on their scientific task rather than IT system management. To make this as easy to manage as possible in a distributed environment, the browser or portal can present the user with a uniform abstraction that hides the details of disparate remote systems. All remote sensing data must be processed to be useful. Besides research and development on calibration algorithms and the like, data is often used in other computational tasks, e.g., weather forecasting, oceanic/atmospheric models, climate models, etc. In this respect, G-POD stands-out since it allows the user to explicit initiate and manage computational jobs.

There are a number of standards for managing remote services. The Simple Object Access Protocol (SOAP) is widely used for request-replies in the web services domain. In the geospatial domain, the Web Processing Service (WPS) provides a similar request-reply protocol, and is gaining wider acceptance since it has a look-and-feel similar to other OGC standards. In the grid domain, the Open Grid Forum (OGF) HPC Basic Profile [94] was defined to support a wider variety of job management requirements beyond simple request-reply, e.g., meta-scheduling policies, "rich clients" that need to run sets of simulations, and also workflow engines that manage sets of tasks chained together in a dependency graph. The HPC Basic Profile accomplishes this through the use of the OGF Job Submission Description Language, the OGF Basic Execution Service, and the WS-I Basic Profile.

Workflow engines become advantageous when multiple tasks have to be executed on a routine basis. The best known workflow standard is the Business Process Execution Language (BPEL) [95] used by web services. The Workflow Management Coalition has also defined standard workflow tools such as the XML Process Definition Language and the Workflow XML protocol. There has, however, been a tremendous amount of work on workflow engines outside of standardization efforts, primarily in the grid community. Workflow engines such as Kepler, Pegasus, Taverna and Triana have been used in large, operational systems to routinely manage thousands of jobs and terabytes of data [55]. Key design features of workflow engines are whether they have centralized or distributed control ("orchestration" vs. "choreography") and how intelligently they can respond to failures in the workflow or environment [54]. Both the Matsu Project and GEO Grid employed workflow engines. The Matsu Project is experimenting with the Geospatial Business Process Management System (GeoBMPS), a commercial product and service from GeoBliki.

While all projects make use of web browsers and portals, with their implied use of HTTP, very little was said about the communication or networking requirements. GMSEC does support publish-subscribe, in addition to event notification. There are related communication paradigms for which a number of commercial tools are available. External users may commonly use the open Internet for accessing remote systems. Satellite ground systems and other organizations, however, could benefit from *bandwidth on-demand*. Besides providing higher bandwidth, optical networks also support bandwidth on-demand through dynamic wavelength allocation and multiplexing. If the communication demands are known or can be predicted in advance, then different wavelengths can be allocated to provide a dedicated path with higher bandwidth, no competing traffic and better reliability. The GÉANT2 project and the OGF Network Service Interface Working Group are jointly working on standard tools to manage such capabilities.

Another very important development is the use of on-demand resources such as those provided by the different types of *clouds*. As previously discussed the concept of virtualization is an established concept in computing, but its successful use for on-demand computing resources has been enabled by the development of massive data centers to support web-based applications and the interconnectedness of society, i.e., the easy access to those data centers. While the vast majority of cloud computing applications will be in the commercial sector with primarily transactional applications, there is strong interest in the scientific and engineering fields, including satellite ground systems, for many of the same reasons as the commercial field [96]. The dynamic allocation of virtual machines, storage and communication enables a flexible response to changing demands. Desired throughput can be maintained by adding additional servers, but only when necessary. The use of virtual machine images can isolate applications from changes in the underlying hardware. The ease of server allocation can be used to improve reliability by spinning up another server automatically when a failure has been detected. From an end-user perspective, it is also more desirable to have the appearance of having a dedicated virtual cluster, rather than having to submit jobs through a job scheduler and waiting for the turn-around time.

Of course, cloud computing does present a performance issue for HPC since virtualization of the network interface can neg-

atively impact communication performance. For very tightly coupled, communication-intensive codes, this performance impact may be unacceptable. For many scientific codes, however, the impact may be tolerable and worth the other benefits described above. In fact, clouds can be designed, configured and deployed to minimize those overheads [97]. Hence, the notion of *science clouds* is rapidly gaining popularity. Here, the communication overhead can be reduced by avoiding full virtualization of the network interface and optimizing data handling between the guest and driver domains. Also, data can be accessed through block storage that is attached directly to running virtual machine instances, rather than through an HTTP interface. In addition, GPU clouds are also being developed to make the computational speed of these devices available, on-demand, for cloud applications. As an example, the DOE Magellan Cloud at Argonne National Lab includes 133 servers that host 266 nVidia 2070 GPU cards. While the GPU servers can be multi-tenant, the virtualization of the GPUs themselves remains to be implemented. The virtualization of GPUs might introduce overheads that would negatively impact the final performance realized by applications. Nonetheless, hosting GPUs in a cloud is a way to integrate their performance advantages with the on-demand flexibility of cloud computing.

The final topics we wish to discuss are those of *policy* and *governance*. While the examples presented a range of basic project management requirements—from research mash-ups (Matsu) to international collaborations (GEOSS)—a key indicator for the maturity of DCIs is the degree of integration and automation of their policy and governance mechanisms. We note that while GMSEC enables rapid prototyping for ground system service integration by controlling the message bus API, its deployment is largely configured statically. It relies on manual configuration of the firewalls and deployment of the message bus infrastructure to control user authorizations. GEO Grid's use of VOs, however, provides a mechanism whereby data owners can flexibly define VO roles and manage user authorizations. This is especially important for DCIs that cross administrative boundaries, such as GEOSS. While simply serving remote sensing data is a major function for satellite DCIs, this will be increasingly integrated with the management of computational tasks. The OGC Web Processing Service is a first step in this direction, but ultimately the general management of federated computational resources will have to be addressed. This has been the focus of the grid community for many years, and continues to be highly relevant in the cloud community. Standards like WS-Agreement and WS-Agreement Negotiation [98] were developed to establish and monitor service level agreements whereby properties—such as performance, security and reliability—could be agreed upon by a provider and client.

V. Discussion of Challenges

While this review has shown us a number of important systems leading the way in the development of complete, end-to-end platforms for remote sensing, we note there are still a large number of outstanding challenges in this field. To establish the scope and scale of these challenges, we use another key, motivating example. In 2005, Hurricane Katrina took over
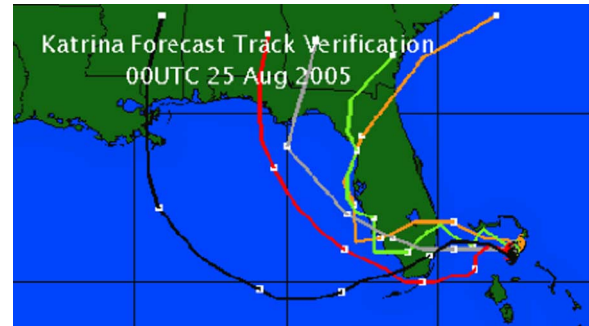

Fig. 9. Predictions of Hurricane Katrina's track four days before landfall in Louisiana. The black line is the actual path.

1500 lives and caused over \$81B in property damage [99]. Four days prior to landfall, multiple hurricane prediction codes gave the results shown in Fig. 9 from [100]. The black line is the hurricane's actual path. Clearly these predictions were useless four days out, and only began to converge to "truth" two days out. To mitigate such disasters, what would be necessary to build and deploy an HPC-based *hurricane disaster mitigation* system?

When considering such a system, one realizes that this represents both a *scientific* and *operational grand challenge problem*. Addressing the basic scientific issues will require a fundamentally enhanced understanding of how the atmospheric and oceanic systems work as part of the overall Earth system, in addition to developing corresponding computational models that accurately represent these systems. The scale of these models may require larger scale computational infrastructure beyond what is currently deployed.

As an example, consider the DCI requirements for tracking a hurricane, from birth as a tropical storm to a full fledge hurricane that makes land fall. The DCI must be able to ingest multiple sources of real-time data that includes satellite, airborne and ground based observations. This data must then feed real-time forecast model to support hurricane track prediction and then feed this data to various organizations and decision support systems. The results of these tracking models will also feed precipitation models to estimate where water will be deposited, which will have to be fed into flooding models to determine where lives and property will be at risk. To be truly effective, this must be done securely across federated organizations and nations, so critical information can be immediately available to public officials to manage evacuation routes, sandbagging, and other mitigation efforts. To realize such a DCI will require an enormous amount of compute power that is just not economically possible to dedicate to this single purpose.

Hence, shared computing resources (including all types of HPC-based platforms discussed in this review) will have to be used in complementary fashion. Although the role of each type of architecture depends heavily on the considered remote sensing application, cluster-based parallel computing seems particularly appropriate for efficient information extraction from very large data archives comprising data sets already transmitted to Earth, while the time-critical constraints introduced by many remote sensing applications such as the one discussed in this section call for on-board and often real-time

processing developments. This includes specialized hardware architectures such as GPUs and FPGAs. In all cases, these computing resources must also be available *on-demand*, possibly from a national cloud resource that can support coupled, HPC codes with strict processing deadlines. Clearly such a grand challenge system could support a wide variety of application domains. Hence, we can distill the following fundamental challenge areas:

- *On-demand Scale and Timeliness*. To date, running large compute jobs has meant submitting a job to a batch scheduler and waiting through a job queue. Cloud computing, however, is based on the notion of acquiring resources on-demand. While commercial cloud computing primarily targets a transactional style of computing, there is also great interest in building *science clouds* that can support more tightly coupled HPC codes on-demand. To support a disaster mitigation DCI will require the allocation of resources to support sets of applications in a workflow that ingests real-world data in real-time, and provides data products to a distributed user base. This scale is tantamount to the allocation *virtual data centers* with hard real-time deadline constraints.

- *Security and Information Assurance*. Such large systems may actually be distributed across multiple data centers in different administrative domains, crossing not only organizational but also national boundaries. Hence, *federated identity management systems* are needed that provide *single sign-on*. Role-based authorization can be managed based on a user's identity and role within a *virtual organization*. The trust relationships necessary to operate such virtual organizations will be managed through *trust federations* that specify how Certificate Authorities are being operated, thereby establishing trust among the participating organizations. A key example is the International Grid Trust Federation that coordinates Certificate Authorities operating on six continents [101]. Irrespective of the security mechanisms that are in use, however, there is a fundamental trade-off between security and system performance and usability. Choosing the right level of security balanced by performance and system usability is always a challenge.

- *Data and Data Access*. Several of the projects reviewed focused on data and data accessibility. With the current exploding "data deluge" being captured or generated and put on-line, the importance of data access cannot be overestimated. There are certainly a number of standards for geospatial data, catalogs, and web-based presentation, but this just scratches the surface [84]. There is still a wide diversity of data and metadata formats that must be contended with, and improved methods for managing this information and providing users with simple methods (and the tools) to access this data is extremely important. When data sets are owned by different institutions, there can also be different authentication, authorization, and data access methods that provide additional challenges for data accessibility. Ultimately, the goal is to create *digital libraries* where current and historical data sets are curated and preserved with their provenance, and can be accessed using a well defined set of standards.

- *Standards and Interoperability*. Clearly none of these systems will be feasible without internationally recognized and adopted standards in all of the fundamental capability areas described earlier. The number of possible technical standards is far beyond what we can review here. With regard to managing distributed resource federation, the Open Grid Forum [102] has developed concepts such as virtual organizations and trust federation. There are also emerging cloud standards, such as the Open Cloud Computing Interface [103], the Open Virtualization Format [104], and the Cloud Data Management interface [105] that together form the basis for standard IaaS clouds. This is covered in a little more detail in [106].

- *Paths for Incremental Adoption*. Finally, if we take note of the lessons learned from previous failed attempts to deploy large scale systems as a monolithic whole, and the fact international standards are never developed or adopted overnight, incremental approaches to development and deployment should be taken. While it is possible to envision such large systems, and the standards to support them, it is much more feasible to incrementally adopt and deploy maturing technologies and develop standards around specific functions. There are significant benefits to this "grow-as-you-go" approach. This allows experience to be gained and risk minimized as new capabilities are deployed and incrementally expanded, while riding the price-performance curve for computing technologies. Hence, we must support all manner of prototyping and pathfinder efforts to build user and marketplace confidence across all aspects of the HPC-based systems discussed in this review.

VI. Conclusions

In this paper, we have reviewed the state-of-the-art in the application of HPC techniques and practices to remote sensing problems. Techniques discussed include specialized hardware devices, multi-processor systems and distributed networks, which provide important architectural developments to accelerate the computations related with information extraction in remote sensing. Our study reveals that specialized hardware systems can now satisfy the time-critical constraints introduced by several remote sensing applications, while the computational power offered by clusters and distributed networks is ready to introduce substantial benefits from the viewpoint of integrating available computing resources and exploiting large volumes of remotely sensed data. While there are still some important challenges in this field, the compendium of techniques and platforms discussed in this work reflect the increasing sophistication of a field that is rapidly maturing at the intersection of many different disciplines.

Acknowledgment

REFERENCES

[1] R. A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing*, 2nd ed. New York: Academic Press, 1997.

[2] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. New York: Wiley, 2003.

[3] C.-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. New York: Kluwer Academic/Plenum, 2003.

[4] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*. New York: Springer, 2006.

[5] A. Plaza, J. A. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, J. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. 110–122, 2009.

[6] S. Kalluri, Z. Zhang, J. JaJa, S. Liang, and J. Townshend, "Characterizing land surface anisotropy from AVHRR data at a global scale using high performance computing," *Int. J. Remote Sens.*, vol. 22, pp. 2171–2191, 2001.

[7] A. Plaza and C.-I Chang, *High Performance Computing in Remote Sensing*. Boca Raton, FL: Taylor & Francis, 2007.

[8] A. Plaza, "Special issue on architectures and techniques for real-time processing of remotely sensed images," *J. Real-Time Image Process.*, vol. 4, pp. 191–193, 2009.

[9] A. Plaza and C.-I Chang, "Preface to the special issue on high performance computing for hyperspectral imaging," *Int. J. High Performance Computing Applications*, vol. 22, no. 4, pp. 363–365, 2008.

[10] R. Brightwell, L. Fisk, D. Greenberg, T. Hudson, M. Levenhagen, A. Maccabe, and R. Riesen, "Massively parallel computing using commodity components," *Parallel Computing*, vol. 26, pp. 243–266, 2000.

[11] J. Dorband, J. Palencia, and U. Ranawake, "Commodity computing clusters at goddard space flight center," *J. Space Commun.*, vol. 3, p. 1, 2003.

[12] K. Itoh, "Massively parallel fourier-transform spectral imaging and hyperspectral image processing," *Optics and Laser Technology*, vol. 25, p. 202, 1993.

[13] T. El-Ghazawi, S. Kaewpijit, and J. L. Moigne., "Parallel and adaptive reduction of hyperspectral data to intrinsic dimensionality," *Cluster Computing*, vol. 1, pp. 102–110, 2001.

[14] A. Plaza, D. Valencia, J. Plaza, and P. Martinez, "Commodity cluster-based parallel processing of hyperspectral Imagery," *J. Parallel and Distributed Computing*, vol. 66, no. 3, pp. 345–358, 2006.

[15] A. Plaza, J. Plaza, and D. Valencia, "Impact of platform heterogeneity on the design of parallel algorithms for morphological processing of high-dimensional image data," *J. Supercomputing*, vol. 40, no. 1, pp. 81–107, 2007.

[16] A. Lastovetsky and J. Dongarra, *High-Performance Heterogeneous Computing*. New York: Wiley, 2009.

[17] A. Plaza, D. Valencia, and J. Plaza, "An experimental comparison of parallel algorithms for hyperspectral analysis using homogeneous and heterogeneous networks of workstations," *Parallel Computing*, vol. 34, no. 2, pp. 92–114, 2008.

[18] S. Tehranian, Y. Zhao, T. Harvey, A. Swaroop, and K. Mckenzie, "A robust framework for real-time distributed processing of satellite data," *J. Parallel and Distributed Computing*, vol. 66, pp. 403–418, 2006.

[19] W. Rivera, C. Carvajal, and W. Lugo, "Service oriented architecture grid based environment for hyperspectral imaging analysis," *Int. J. Information Technology*, vol. 11, no. 4, pp. 104–111, 2005.

[20] J. Brazile, R. A. Neville, K. Staenz, D. Schlaepfer, L. Sun, and K. I. Itten, "Cluster versus grid for operation generation of ATCOR's MODTRAN-based look up table," *Parallel Computing*, vol. 34, pp. 32–46, 2008.

[21] S. Hauck, "The roles of FPGAs in reprogrammable systems," *Proc. IEEE*, vol. 86, pp. 615–638, 1998.

[22] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym, "NVIDIA Tesla: A unified graphics and computing architecture," *IEEE Micro*, vol. 28, pp. 39–55, 2008.

[23] A. Plaza, J. Plaza, A. Paz, and S. Sanchez, "Parallel hyperspectral image and signal processing," *IEEE Signal Process. Mag.*, vol. 28, pp. 119–126, 2011.

[24] J. Nickolls and W. J. Dally, "The GPU computing era," *IEEE Micro*, vol. 30, pp. 56–69, 2010.

[25] T. Balz and U. Stilla, "Hybrid GPU-based single- and double-bounce SAR simulation," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 10, pp. 3519–3529, 2009.

[26] T. Hobiger, R. Ichikawa, Y. Koyama, and T. Kondo, "Computation of troposphere slant delays on a GPU," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 10, pp. 3313–3318, 2009.

[27] M. W. Govett, J. Middlecoff, and T. Henderson, "Running the NIM next-generation weather model on GPUs," in *Proc. 10th IEEE/ACM Int. Conf. Cluster, Cloud and Grid Computing (CCGrid)*, 2010, vol. 1, pp. 792–796.

[28] J. Mielikainen, R. Honkanen, B. Huang, P. Toivanen, and C. Lee, "Constant coefficients linear prediction for lossless compression of ultraspectral sounder data using a graphics processing unit," *J. Applied Remote Sens.*, vol. 4, no. 1, pp. 751–774, 2010.

[29] J. Setoain, M. Prieto, C. Tenllado, and F. Tirado, "GPU for parallel on-board hyperspectral image processing," *Int. J. High Performance Computing Applications*, vol. 22, no. 4, pp. 424–437, 2008.

[30] J. Setoain, M. Prieto, C. Tenllado, A. Plaza, and F. Tirado, "Parallel morphological endmember extraction using commodity graphics hardware," *IEEE Geosci. Remote Sens. Lett.*, vol. 43, no. 3, pp. 441–445, 2007.

[31] A. Plaza, J. Plaza, and H. Vegas, "Improving the performance of hyperspectral image and signal processing algorithms using parallel, distributed and specialized hardware-based systems," *J. Signal Process. Syst.*, vol. 61, pp. 293–315, 2010.

[32] B. Huang, J. Mielikainen, H. Oh, and H.-L. Huang, "Development of a GPU-based high-performance radiative transfer model for the infrared atmospheric sounding interferometer (IASI)," *J. Comput. Phys.*, vol. 230, pp. 2207–2221, 2011.

[33] J. Mielikainen, B. Huang, and A. H.-L. Huang, "GPU-accelerated multi-profile radiative transfer model for the infrared atmospheric sounding interferometer," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. (JSTARS)*, vol. 4, no. 3, pp. 691–700, Sep. 2011.

[34] C. Song, Y. Li, and B. Huang, "A GPU-accelerated wavelet decompression system with SPIHT and Reed-Solomon decoding for satellite images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. (JSTARS)*, vol. 4, no. 3, pp. 683–690, Sep. 2011.

[35] S.-C. Wei and B. Huang, "GPU acceleration of predictive partitioned vector quantization for ultraspectral sounder data compression," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. (JSTARS)*, vol. 4, no. 3, pp. 677–682, Sep. 2011.

[36] C.-C. Chang, Y.-L. Chang, M.-Y. Huang, and B. Huang, "Accelerating regular LDPC code decoders on GPUs," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. (JSTARS)*, vol. 4, no. 3, pp. 653–659, Sep. 2011.

[37] Y. Tarabalka, T. V. Haavardsholm, I. Kasen, and T. Skauli, "Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and gpu processing," *J. Real-Time Image Process.*, vol. 4, pp. 1–14, 2009.

[38] D. A. Buell, T. A. El-Ghazawi, K. Gaj, and V. V. Kindratenko, "Guest editors' introduction: High-performance reconfigurable computing," *IEEE Computer*, vol. 40, pp. 23–27, 2007.

[39] U. Thomas, D. Rosenbaum, F. Kurz, S. Suri, and P. Reinartz, "A new software/hardware architecture for real time image processing of wide area airborne camera images," *J. Real-Time Image Process.*, vol. 5, pp. 229–244, 2009.

[40] A. Plaza and C.-I Chang, "Clusters versus FPGA for parallel processing of hyperspectral imagery," *Int. J. High Performance Computing Applications*, vol. 22, no. 4, pp. 366–385, 2008.

[41] M. Hsueh and C.-I Chang, "Field programmable gate arrays (FPGA) for pixel purity index using blocks of skewers for endmember extraction in hyperspectral imagery," *Int. J. High Performance Computing Applications*, vol. 22, pp. 408–423, 2008.

[42] E. El-Araby, T. El-Ghazawi, J. L. Moigne, and R. Irish, "Reconfigurable processing for satellite on-board automatic cloud cover assessment," *J. Real-Time Image Process.*, vol. 5, pp. 245–259, 2009.

[43] C. Gonzalez, J. Resano, D. Mozos, A. Plaza, and D. Valencia, "FPGA implementation of the pixel purity index algorithm for remotely sensed hyperspectral image analysis," *EURASIP J. Advances in Signal Processing*, vol. 969806, pp. 1–13, 2010.

[44] A. Paz and A. Plaza, "Clusters versus GPUs for parallel automatic target detection in remotely sensed hyperspectral images," *EURASIP J. Advances in Signal Processing*, vol. 915639, pp. 1–18, 2010.

[45] J. Resano, J. A. Clemente, C. González, D. Mozos, and F. Catthoor, "Efficiently scheduling runtime reconfigurations," *ACM Trans. Design Automation of Electronic Systems*, vol. 13, pp. 58–69, 2008.

[46] A. Plaza, J. A. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, J. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. 110–122, 2009.

[47] J. C. Tilton, W. T. Lawrence, and A. Plaza, "Utilizing hierarchical segmentation to generate water and snow masks to facilitate monitoring of change with remotely sensed image data," *GIScience & Remote Sens.*, vol. 43, pp. 39–66, 2006.

[48] J. Plaza, A. Plaza, D. Valencia, and A. Paz, "Massively parallel processing of hyperspectral images," in *Proc. SPIE*, 2009, vol. 7455, pp. 1–11.

[49] A. Remon, S. Sanchez, A. Paz, E. S. Quintana-Orti, and A. Plaza, "Real-time endmember extraction on multi-core processors," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 924–928, 2011.

[50] *"Systems Engineering Handbook, A Guide For System Life Cycle Processes and Activities,"* ver. 3.1, C. Haskins, Ed., INCOSE, 2007, IN-COSE-TP-2003-002-03.1.

[51] Apache Hadoop Wiki. 2011 [Online]. Available: http://wiki.apache.org/hadoop/

[52] S. R. Arnon, S. A. Renner, A. S. Rosenthal, and J. G. Scarano, "Data interoperability: Standardization or mediation," presented at the IEEE Metadata Workshop, Silver Spring, MD, 1996.

[53] A. Merzky, K. Stamou, and S. Jha, "Application level interoperability between clouds and grids," in *Workshops at the Grid and Pervasive Computing Conf. 2009 (GPC'09)*, May 2009, pp. 143–150.

[54] C. Lee, B. Michel, E. Deelman, and J. Blythe, "From event-driven workflows towards a posteriori computing," in *Future Generation Grids*, V. Getov, D. Laforenza, and A. Reinefeld, Eds. New York: Springer-Verlag, 2006, pp. 3–28.

[55] The Grid Workflow Forum. 2011 [Online]. Available: http://www.grid-workflow.org

[56] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *J. Supercomput.* vol. 54, pp. 252–269, Nov. 2010 [Online]. Available: http://dx.doi.org/10.1007/s11227-009-0318-1

[57] M. Caramia and S. Giordani, "Resource allocation in grid computing: An economic model," *WSEAS Trans. Computer Research*, vol. 3, no. 1, pp. 19–27, 2008.

[58] M. Li, M. Chen, and J. Xie, "Cloud computing: A synthesis models for resource service management," in *2nd Int. Conf. Communication Systems, Networks and Applications (ICCSNA 2010)*, Jul. 2010, vol. 2, pp. 208–211.

[59] D. Lee, J. J. Dongarra, and R. S. Ramakrishna, "visPerf: Monitoring Tool for Grid Computing," in *ICCS 2003*, ser. Lecture Notes in Computer Science, vol. 2657. New York: Springer Verlag, 2003, pp. 233–243.

[60] Z. Balaton, P. Kacsuk, N. Podhorszki, and F. Vajda, "Comparison of representative grid monitoring tools," Lab. of Parallel and Distributed Systems, Computer and Automation Research Inst., Hungarian Academy of Sciences, Tech. Rep. LPDS-2/2000, 2000.

[61] G. Shaheen, M. Malik, Z. Ihsan, and N. Daudpota, "Grid visualizer: A monitoring tool for grid environment," in *Proc. 16th Int. Workshop on Database and Expert Systems Applications 2005*, Aug. 2005, pp. 297–301.

[62] D. Chou, "Using events in highly distributed architectures," *MSDN Library: The Architecture Journal,* no. 17, Oct. 2008 [Online]. Available: http://msdn.microsoft.com/en-us/library/dd129913.aspx

[63] Y. Huang and D. Gannon, "A comparative study of web services-based event notification specifications," in *Proc. 2006 Int. Conf. Parallel Processing (ICPP 2006) Workshops*, 2006, pp. 8–14.

[64] K. B. Alexander, National Information Assurance (IA) Glossary, Committee on National Security Systems CNSS Secretariat (I01C), National Security Agency, Ft. Meade, MD, Tech. Rep. CNSS Instruction 4009, Jun. 2006 [Online]. Available: http://www.cnss.gov/

[65] C. Tilmes and A. Fleig, "Provenance tracking in an earth science data processing system," presented at the 2nd Int. Provenance and Annotation Workshop Salt Lake City, Utah, Jun. 17–18, 2008.

[66] I. Sfiligoi, G. Quinn, C. Green, and G. Thain, "Pilot job accounting and auditing in open science grid," in *Proc. 9th IEEE/ACM Int. Conf. Grid Computing 2008*, Oct. 2008, pp. 112–117.

[67] E. Gallopoulos, E. Houstis, and J. Rice, "Computer as thinker/doer: Problem-solving environments for computational science," *IEEE Comput. Sci. Eng.*, vol. 1, no. 2, pp. 11–23, Summer, 1994.

[68] *High Performance Computing in Remote Sensing*, ser. Chapman & Hall/CRC Computer & Information Science Series, A. J. Plaza and C.-I Chang, Eds. Boca Raton, FL: Chapman & Hall/CRC, 2007, vol. 16.

[69] N. DeBardeleben, I. Ligon, W. B. J. Stanzione, and D. C. , "The component-based environment for remote sensing," in *2002 IEEE Aerospace Conf. Proc.*, 2002, vol. 6, pp. 6-2661–6-2670.

[70] G. Aloisio, M. Cafaro, I. Epicoco, and G. Quarta, "A problem solving environment for remote sensing data processing," in *Proc. Int. Conf. Information Technology: Coding and Computing (ITCC 2004)*, Apr. 2004, vol. 2, pp. 56–61.

[71] C. Lee and G. Percivall, "Standards-based computing capabilities for distributed geospatial applications," *IEEE Computer*, pp. 50–57, Nov. 2008.

[72] D. Mandl, "Matsu: An elastic cloud connected to a sensorweb for disaster response," in *Ground System Architectures Workshop (GSAW), Workshop on Cloud Computing for Spacecraft Operations*, Mar. 2, 2011.

[73] The Open Cloud Consortium [Online]. Available: http://www.open-cloudconsortium.org.

[74] The Global Disaster Alert and Coordination System [Online]. Available: http://www.gdacs.org

[75] R. Cossu and L. Fusco, "GENESI-DR . . . and its future," in *GEO Alliances and Harmonization Workshop*, Nov. 11–12, 2009.

[76] GENESI-DEC [Online]. Available: http://www.genesi-dec.eu

[77] The EC INSPIRE Directive [Online]. Available: http://inspire.jrc.ec.europa.eu

[78] OpenSearch [Online]. Available: http://www.opensearch.org

[79] Grid Processing on Demand [Online]. Available: http://gpod.eo.esa.int

[80] Globus [Online]. Available: http://globus.org

[81] F. Brito, "Cloud computing in ground segments: Earth observation processing campaigns," in *Ground System Architectures Workshop (GSAW), Workshop on Data Center Migration for Ground Systems: Geospatial Clouds*, Mar. 3, 2010.

[82] The GSFC Mission Services Evolution Center [Online]. Available: http://gmsec.gsfc.nasa.gov

[83] S. Sekiguchi *et al.*, "Design principles and IT overviews of the GEO grid," *IEEE Syst. J.*, vol. 2, no. 3, pp. 374–389, Sep. 2008.

[84] The Open Geospatial Consortium [Online]. Available: http://www.opengeospatial.org

[85] H. Yamamoto *et al.*, "Field sensor virtual organization integrated with satellite data on a GEO grid," *Data Science J.*, vol. 8, pp. 461–476, Feb. 7, 2010.

[86] The Global Earth Observation System of Systems [Online]. Available: http://www.earthobservations.org/geoss.shtml

[87] The Group on Earth Observations [Online]. Available: http://www.earthobservations.org

[88] The Committee on Earth Observation Satellites [Online]. Available: http://www.ceos.org

[89] L. Fusco and R. Cossu, "Past and future of ESA earth observation grid," (in English) *Memorie della Societá Astronomica Italiana*, vol. 80, pp. 461–476, 2009.

[90] The US Group Earth Observations [Online]. Available: http://www.usgeo.gov

[91] P. Yue *et al.*, "Sharing geospatial provenance in a service-oriented environment," *Computers, Environment and Urban Systems*, 2011, in press.

[92] ISO 14863 System Independent Data Format. Originally available as ECMA-208. [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-208.pdf

[93] Long Term Preservation of Earth Observation Space Data: European LTDP Common Guidelines, V2, Jun. 4, 2009 [Online]. Available: http://earth.esa.int/gscb/ltdp/EuropeanLTDPCommonGuidelines_DraftV2.pdf

[94] The OGF HPC Basic Profile, GFD.114 [Online]. Available: http://www.ogf.org/documents/GFD.114.pdf

[95] The OASIS Web Services Business Process Execution Language, V2 [Online]. Available: http://docs.oasis-open.org/wsbpel/2.0/OS/ws-bpel-v2.0-OS.html

[96] C. A. Lee, "A perspective on scientific cloud computing," in *Science Cloud Workshop, HPDC*, Jun. 2010.

[97] K. Jackson *et al.*, "Seeking supernovae in the clouds: A performance study," in *Science Cloud Workshop, HPDC*, Jun. 2010.

[98] OGF WS-Agreement Negotiation [Online]. Available: https://forge.gridforum.org/sf/go/doc16194?nav=1

[99] The Deadliest, Costliest, and Most Intense United States Tropical Cyclones From 1851 to 2006 (and Other Frequently Requested Hurricane Facts), 2007 [Online]. Available: http://www.nhc.noaa.gov/pdf/NWS-TPC-5.pdf, Retrieved 2011-03-03

[100] P. Bogden *et al.*, "Architecture of a community infrastructure for predicting and analyzing coastal inundation," *Marine Tech. Soc. J.*, vol. 41, no. 1, pp. 53–61, Jun. 2007.

[101] The International Grid Trust Federation, 2011 [Online]. Available: http://www.igtf.net

[102] The Open Grid Forum [Online]. Available: http://www.ogf.org

[103] The OGF Open Cloud Computing Interface [Online]. Available: http://www.occi-wg.org/doku.php

[104] The DMTF Open Virtualization Format [Online]. Available: www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf

[105] The SNIA Cloud Data Management Interface [Online]. Available: http://www.snia.org/cloud

[106] C. A. Lee and G. Percivall, "Emerging standards for geospatial distributed computing infrastructures," *Data Flow from Space to Earth: Applications and Interoperability,* Mar. 21–23, 2011 [Online]. Available: http://www.space.corila.it

Craig A. Lee (M'94) received the Ph.D. in computer science from the University of California, Irvine.

He is a Senior Scientist in the Computer Systems Research Department of The Aerospace Corporation, a non-profit, federally funded, research and development center. He has conducted DARPA and NSF sponsored research in the areas of grid computing, optimistic models of computation, active networks, and distributed simulations, in collaboration with USC, UCLA, Caltech, Argonne National Lab, and the College of William and Mary. Dr. Lee served as the President of the Open Grid Forum (OGF) from 2007 to 2010, and facilitated the Open Cloud Computing Interface (OCCI), an open standard API for infrastructure clouds. He currently sits on OGF's Board of Directors and NIST's Cloud Computing Standards Roadmap Working Group. He is active in the Large Data Working Group for the Open Cloud Consortium facilitating on-demand, satellite imagery analysis for disaster assessment, in collaboration with NASA Goddard. He has been also instrumental in the NCOIC proposal to the NGA for a GEOINT Community Cloud pathfinder project. He has published over 67 technical works, is on the steering committee for the Grid XY and CCGrid conference series, and sits on the Editorial Boards of *Future Generation Computing Systems* (Elsevier) and the *Journal of Cloud Computing* (Inderscience).
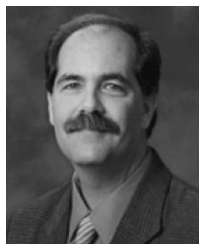
Samuel D. Gasster (M'85–SM'89) received the Ph.D. and M.A. in physics from the University of California, Berkeley, and the S.B. in mathematics from the Massachusetts Institute of Technology (MIT), Cambridge, MA.

He is a Senior Scientist in the Computer Systems Research Department of The Aerospace Corporation. He specializes in high performance computing for scientific applications. His research interests include the application of high performance computing technology for scientific and remote-sensing applications, in quantum information science and technology, data-modeling and data-management system development, and systems and software engineering. He has worked at Aerospace for over 20 years and has supported a wide range of defense and civilian programs and agencies, including DMSP, NPOESS, DARPA, NASA, and NOAA. He has taught remote sensing and computer science at UCLA Extension. He has also been a judge at the California State Science Fair in the Software and Mathematics Section.

Dr. Gasster is a life member of the APS, a Senior Member of the IEEE, a member of the AGU and INCOSE.

Antonio Plaza (M'05–SM'07) received the M.S. and Ph.D. degrees in computer engineering from the University of Extremadura, Caceres, Spain.

He was a Visiting Researcher with the Remote Sensing Signal and Image Processing Laboratory, University of Maryland Baltimore County, Baltimore, with the Applied Information Sciences Branch, Goddard Space Flight Center, Greenbelt, MD, and with the AVIRIS Data Facility, Jet Propulsion Laboratory, Pasadena, CA. He is currently an Associate Professor with the Department of Technology of Computers and Communications, University of Extremadura, Caceres, Spain, where he is the Head of the Hyperspectral Computing Laboratory (HyperComp). He was the Coordinator of the Hyperspectral Imaging Network (Hyper-I-Net), a European project designed to build an interdisciplinary research community focused on hyperspectral imaging activities. He has been a Proposal Reviewer with the European Commission, the European Space Agency, and the Spanish Government. He is the author or coauthor of more than 280 publications on remotely sensed hyperspectral imaging, including more than 50 journal citation report papers, book chapters, and conference proceeding papers. His research interests include remotely sensed hyperspectral imaging, pattern recognition, signal and image processing, and efficient implementation of large-scale scientific problems on parallel and distributed computer architectures.

Dr. Plaza has coedited a book on high-performance computing in remote sensing and guest edited four special issues on remotely sensed hyperspectral imaging for different journals, including the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (for which he serves as Associate Editor on hyperspectral image analysis and signal processing since 2007), the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATION AND REMOTE SENSING, the *International Journal of High Performance Computing Applications*, and the *Journal of Real-Time Image Processing*. He has served as a reviewer for more than 240 manuscripts submitted to more than 40 different journals, including more than 120 manuscripts reviewed for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He has served as a Chair for the IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing in 2011. He has also been serving as a Chair for the SPIE Conference on Satellite Data Compression, Communications, and Processing since 2009, and for the SPIE Europe Conference on High Performance Computing in Remote Sensing since 2011. He is a recipient of the recognition of Best Reviewers of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2009 and a recipient of the recognition of Best Reviewers of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2010. He is currently serving as Director of Education activities for the IEEE Geoscience and Remote Sensing Society.

Chein-I Chang (F'10) received the B.S. degree from Soochow University, Taipei, Taiwan, the M.S. degree from the Institute of Mathematics, National Tsing Hua University, Hsinchu, Taiwan, and the M.A. degree from the State University of New York at Stony Brook, all in mathematics. He also received the M.S. and M.S.E.E. degrees from the University of Illinois at Urbana-Champaign and the Ph.D. degree in electrical engineering from the University of Maryland, College Park.

He has been with the University of Maryland, Baltimore County (UMBC), since 1987 and is currently a Professor in the Department of Computer Science and Electrical Engineering. He was a visiting research specialist in the Institute of Information Engineering at the National Cheng Kung University, Tainan, Taiwan, from 1994 to 1995. He received an NRC (National Research Council) senior research associateship award from 2002–2003 sponsored by the US Army Soldier and Biological Chemical Command, Edgewood Chemical and Biological Center, Aberdeen Proving Ground, Maryland. Additionally, Dr. Chang was a distinguished lecturer chair at the National Chung Hsing University sponsored by the Ministry of Education in Taiwan, ROC from 2005–2006. He was a chair professor of the Environmental Restoration and Disaster Reduction Research Center and Department of Electrical Engineering, National Chung Hsing Universuty, Taichung, Taiwan, ROC and has been chair professor of remote sensing technology at the same institute since 2009. He was also a distinguished visiting fellow/fellow professor sponsored by the National Science Council in Taiwan, ROC from 2009 to 2010.

Dr. Chang was a plenary speaker for SPIE Optics+Applications, Remote Sensing Symposium, 2009. In addition, he was also a keynote speaker at the User Conference of Hyperspectral Imaging 2010, 30 December 2010, Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan; 2009 Annual Meeting of the Radiological Society of the Republic of China, 29 March, Taichung, Taiwan, 2008 International Symposium on Spectral Sensing Research (ISSSR) in 2008; and Conference on Computer Vision, Graphics, and Image Processing 2003 (CVGIP 2003), Kimen, Taiwan. Dr. Chang has four patents and several pending on hyperspectral image processing. He was the guest editor of a special issue of the *Journal of High Speed Networks on Telemedicine and Applications* (April 2000) and co-guest editor of another special issue of the same journal on Broadband Multimedia Sensor Networks in Healthcare Applications, April 2007. He was also co-guest editor of a special issue on High Performance Computing of Hyperspectral Imaging for the *International Journal of High Performance Computing Applications*, December 2007 and special issue on Signal Processing and System Design in Health Care Applications for

EURASIP Journal on Advanced in Signal Processing, 2009. Dr. Chang has authored a book, "Hyperspectral Imaging: Techniques for Spectral Detection and Classification" (Kluwer Academic Publishers, 2003) and edited two books, "Recent Advances in Hyperspectral Signal and Image Processing" (Transworld Research Network, India, 2006) and "Hyperspectral Data Exploitation: Theory and Applications" (John Wiley & Sons, 2007) and co-edited with A. Plaza a book "High Performance Computing in Remote Sensing" (CRC Press, 2007). He is currently working on a second book, "Hyperspectral Data Processing: Signal Processing Algorithm Design and Analysis" (John Wiley & Sons, 2011) and a third book, "Real Time Hyperspectral Image Processing: Algorithm Architecture and Implementation" (Springer-Verlag, 2012). Dr. Chang was an Associate Editor in the area of hyperspectral signal processing for IEEE Transaction on Geoscience and Remote Sensing 2001–2007. He is currently on the editorial boards of the *Journal of High Speed Networks*, *Recent Patents on Mechanical Engineering*, *International Journal of Computational Sciences and Engineering*, and *Open Remote Sensing Journal*. Dr. Chang is a Fellow of SPIE. His research interests include multispectral/hyperspectral image processing, automatic target recognition, medical imaging.
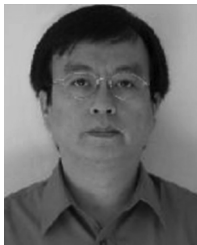
Bormin Huang received the M.S.E. degree in aerospace engineering from the University of Michigan, Ann Arbor, and the Ph.D. in the area of satellite remote sensing from the University of Wisconsin- Madison.

He was with NASA Langley Research Center during 1998–2001 for the NASA New Millennium Program's Geosynchronous Imaging Fourier Transform Spectrometer (GIFTS). He is currently a research scientist and principal investigator at the Space Science and Engineering Center, University of Wisconsin-Madison, where he advises and supports both national and international graduate students and visiting scientists. He was the principal investigator of the NOAA-funded satellite data compression research project for the Hyperspectral Environmental Suite (HES), the next-generation operational geostationary sounder. This project led to a 2006 NOAA bronze medal, the highest honor award that can be granted by the Under Secretary of Commerce for Oceans and Atmosphere. He has authored or coauthored over 100 scientific and technical publications, including the book "Satellite Data Compression" (Springer, 2011). He has broad interests and experiences in remote sensing science and technology, including satellite data compression and communications, remote sensing image processing, remote sensing forward modeling and inverse problems, and high-performance computing in remote sensing.

Dr. Huang has been serving as a Chair for the SPIE Conference on Satellite Data Compression, Communications, and Processing since 2005, and a Chair for the SPIE Europe Conference on High Performance Computing in Remote Sensing since 2011. He currently serves as an Associate Editor for the *Journal of Applied Remote Sensing*, the Guest Editor for the special section on High-Performance Computing in the *Journal of Applied Remote Sensing*, a Guest Editor for the special issue on Advances in Compression of Optical Image Data from Space in the *Journal of Electrical and Computer Engineering*, and the Guest Editor for the special section on Satellite Data Compression in the *Journal of Applied Remote Sensing*. He has also served as a a Program Committee member for several IEEE and SPIE conferences.