This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A First Assessment of the P-SBAS DInSAR Algorithm Performances Within a Cloud Computing Environment

Ivana Zinno, Stefano Elefante, Lorenzo Mossucca, Claudio De Luca, Michele Manunta, Olivier Terzo, Riccardo Lanari, *Fellow, IEEE*, and Francesco Casu

***Abstract*—We present in this work a first performance assessment of the Parallel Small BAseline Subset (P-SBAS) algorithm, for the generation of Differential Synthetic Aperture Radar (SAR) Interferometry (DInSAR) deformation maps and time series, which has been migrated to a Cloud Computing (CC) environment. In particular, we investigate the scalable performances of the P-SBAS algorithm by processing a selected ENVISAT ASAR image time series, which we use as a benchmark, and by exploiting the Amazon Web Services (AWS) CC platform. The presented analysis shows a very good match between the theoretical and experimental P-SBAS performances achieved within the CC environment. Moreover, the obtained results demonstrate that the implemented P-SBAS Cloud migration is able to process ENVISAT SAR image time series in short times (less than 7 h) and at low costs (about USD 200). The P-SBAS Cloud scalable performances are also compared to those achieved by exploiting an in-house High Performance Computing (HPC) cluster, showing that nearly no overhead is introduced by the presented Cloud solution. As a further outcome, the performed analysis allows us to identify the major bottlenecks that can hamper the P-SBAS performances within a CC environment, in the perspective of processing very huge SAR data flows such as those coming from the existing COSMO-SkyMed or the upcoming SENTINEL-1 constellation. This work represents a relevant step toward the challenging Earth Observation scenario focused on the joint exploitation of advanced DInSAR techniques and CC environments for the massive processing of Big SAR Data.**

***Index Terms*—Big data, Cloud Computing (CC), Differential Synthetic Aperture Radar (SAR) Interferometry (DInSAR), Earth surface deformation, Parallel Small BAseline Subset (P-SBAS).**

Manuscript received November 06, 2014; revised February 13, 2015; accepted April 13, 2015. This work was supported in part by the Italian Ministry of University and Research (MIUR) under the project "Progetto Bandiera RITMARE," and in part by the Italian Civil Defence Department (DPC) of the Prime Minister's Office. This work has been carried out through the I-AMICA (Infrastructure of High Technology for Environmental and Climate Monitoring—PONa3_00363) project of Structural improvement financed under the National Operational Programme (NOP) for "Research and Competitiveness 2007–2013," cofunded with European Regional Development Fund (ERDF) and National resources. The ENVISAT SAR data have been provided by the European Space Agency through the Virtual Archive 4. The DEM of the investigated zone was acquired through the SRTM archive.

I. Zinno, S. Elefante, C. De Luca, M. Manunta, R. Lanari, and F. Casu are with the Istituto per il Rilevamento Elettromagnetico dell'Ambiente (IREA), Consiglio Nazionale delle Ricerche, Napoli 80124, Italy (e-mail: casu.f@irea.cnr.it).

C. De Luca is also with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Napoli, Italy.

L. Mossucca and O. Terzo are with the Advanced Computing and Electromagnetics (ACE), Istituto Superiore Mario Boella, Torino 10138, Italy.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2015.2426054

I. Introduction

ADVANCED differential synthetic aperture radar (SAR) interferometry (DInSAR) usually identifies a set of algorithms, tools, and methodologies for the generation of Earth's surface deformation maps and time series computed from a sequence of multitemporal differential SAR interferograms [1]. A widely used advanced DInSAR approach is the technique named Small BAseline Subset (SBAS) [2] which allows to produce line-of-sight (LOS)-projected mean deformation velocity maps and corresponding displacement time series by exploiting interferograms characterized by a small temporal and/or spatial separation (baseline) between the acquisition orbits. The SBAS algorithm has proven its effectiveness to detect ground displacements with millimeters accuracy [3] in different scenarios, such as volcanoes, tectonics, landslides, anthropogenic-induced land motions [4]–[7] and it is capable to perform analyses at different spatial scales [8] and with multisensor data [9], [10].

The SBAS algorithm, and more generally the advanced DInSAR techniques, found their success on the large availability of SAR data archives acquired over time by several satellite systems. Indeed, the current radar Earth Observation (EO) scenario takes advantage of the widely diffused long-term C-band ESA (e.g., ERS-1, ERS-2, and ENVISAT) and Canadian (RADARSAT-1/2) SAR data archives, which have been acquired during the last 20 years, as well as of data sequences provided by the X-band generation SAR sensors, such as the COSMO-SkyMed (CSK) and TerraSAR-X (TSX) constellations. Moreover, a massive and ever increasing data flow will be further supplied by the recently launched (April 2014) Copernicus (European Union) SENTINEL-1A SAR satellite, which will also be paired during 2016 with the SENTINEL-1B twin system that will allow halving the constellation revisit time (from 12 to 6 days) [11]. With the SENTINEL-1 era, new SAR data relevant to extended areas on Earth will be soon publically available, thanks to the free and open access data policy adopted by the Copernicus program. Moreover, the SENTINEL-1 data will be collected on land by using the TOPS SAR mode that has been specifically designed for DInSAR and advanced DInSAR applications [12], [13].

In this context, the massive exploitation of these Big SAR data archives for the generation of advanced DInSAR products will open new research perspectives to understand Earth's surface deformation dynamics at global scale.

However, the accomplishment of this task requires not only large and very high computing resources to process the existing and upcoming huge SAR data amounts within short time frames, but also efficient algorithms able to effectively exploit the available computing facilities.

To provide a contribution toward this direction, a parallel version of the SBAS algorithm, namely parallel SBAS (P-SBAS), has been recently proposed [14]. P-SBAS permits to generate, in an automatic and unsupervised manner, advanced DInSAR products by taking full benefit from parallel computing architectures, such as cluster and GRID infrastructures. P-SBAS has been extensively tested by exploiting in-house processing facilities achieving promising results in terms of both scalability and efficiency [14]–[16]. However, it is worth noting that, even if in-house solutions can provide high computing performances, they can represent a bottleneck due to their intrinsic limited resource availability. Therefore, they cannot be suited to properly face the massive processing that will be inevitably required by the expected huge SAR data flow, particularly when global scale analyses are concerned. Moreover, in-house High Performance Computing (HPC) infrastructures can be very expensive in terms of procurement, maintenance, and upgrading.

The use of Cloud Computing (CC) environments represents a promising solution to overcome the above-mentioned limitations and this is the rationale why they are becoming more and more diffused in EO scenarios [17]–[19]. Indeed, CC provides highly scalable and flexible architectures that are, in general, computationally very efficient and less expensive with respect to in-house solutions. In addition, CC can be extremely helpful for both resources optimization and performance improvements due to its possibility to build up customized computing infrastructures. Moreover, the increasing availability of public Cloud environments [20]–[22], and their relative simplicity to use, thanks to advanced application programming interfaces (API) and web-based tools, is further pushing toward the use of such a technology also in scientific applications [19], [23], [24]. In this context, the migration of scientific applications to CC environments is, therefore, a key issue, with particular reference to advanced DInSAR algorithms, because a significant initial effort and a deep preliminary analysis of the specific algorithm which has to be "cloudified" could be required.

We present in this work a first performance assessment of the P-SBAS algorithm that has been migrated to a public CC environment. In particular, the goal of our study is to evaluate the P-SBAS scalable performances achieved within a CC environment as well as to identify the major inefficiency sources that can hamper such performances. To this aim, we select the proper Cloud migration approach that allows the full exploitation of the P-SBAS parallelization strategy. Moreover, due to the high P-SBAS computational requirements, we evaluate the most appropriate Cloud resources configuration, in terms of both instances and storage. An extensive analysis is carried out by processing a selected SAR images time series acquired (from ascending orbits) by the ENVISAT ASAR sensor over the Napoli bay area and by exploiting the Amazon Web Services (AWS) Cloud platform. The obtained P-SBAS scalable performances are also compared to those achieved by exploiting an in-house, dedicated, HPC cluster.

The paper is organized as follows. In Section II, a concise description of the P-SBAS processing chain is provided, which aims at recalling the main processing steps. Section III describes how the entire P-SBAS processing chain has been migrated to the selected CC environment. Section IV is dedicated to the experimental framework that includes the scalable performance study, which has been performed by exploiting both a dedicated cluster and the AWS Cloud, as well as the analysis of the processing times and costs. Finally, in Section V, some concluding remarks and further developments are thoroughly discussed.

II. P-SBAS Description

This section focuses on providing a concise but informative description of the P-SBAS processing chain which has already been thoroughly discussed in [14]. It aims at recalling the P-SBAS major processing steps in terms of main tasks, implemented procedures and computational challenges that will be addressed by showing CPU usage, RAM occupation, and Input/Output (I/O) transfer requirements.

The P-SBAS solution has been designed by carefully taking into account several different conceptual aspects, such as data dependencies, task partitioning, inherent granularity, scheduling policy, load unbalancing, in order to optimize the usage of CPU, RAM, and I/O resources. Moreover, the heterogeneous nature of the computational algorithms that are comprised within the SBAS processing chain has strived the P-SBAS design toward the employment of proper parallelization strategies that depend on the algorithmic structure of the considered specific processing step [14], [18]. Furthermore, P-SBAS has been designed in a manner that allows us to take advantage of both multinode and multicore architectures and therefore two-parallelization levels have been employed: process and thread. The former considers a coarse/medium granularity-based approach and has been mainly applied to the whole processing chain, while the latter relies on a fine-grained parallelization. This strategy has been implemented both for the most computing-intensive operations, to optimize CPU usage through multithreading programming, as well as for highly RAM demanding algorithms, to reduce memory occupation by applying a data parallelism strategy [14].

The block diagram of the P-SBAS processing chain is shown in Fig. 1; in this scheme, the steps depicted by blue blocks represent the jobs that are parallel executed by simultaneously running on different nodes, while black blocks represent steps that are intrinsically sequential. Moreover, dashed line blocks describe the steps that are multithreaded programmed. In Table I, instead, the main characteristics of each step of Fig. 1, in terms of CPU and RAM usage as well as I/O operations, are briefly summarized in Table I.

In the following, a conceptual description of the P-SBAS processing chain will be briefly addressed. It is worth noting that such a processing chain has been designed to analyze the majority of SAR data available through the different spaceborne systems (ERS-1/2, ENVISAT, COSMO-SkyMed, TerraSAR-X, ALOS-1/2, and RADARSAT-1/2). Moreover, it is also robust with respect to possible system failures (e.g., ERS-2 gyroscope

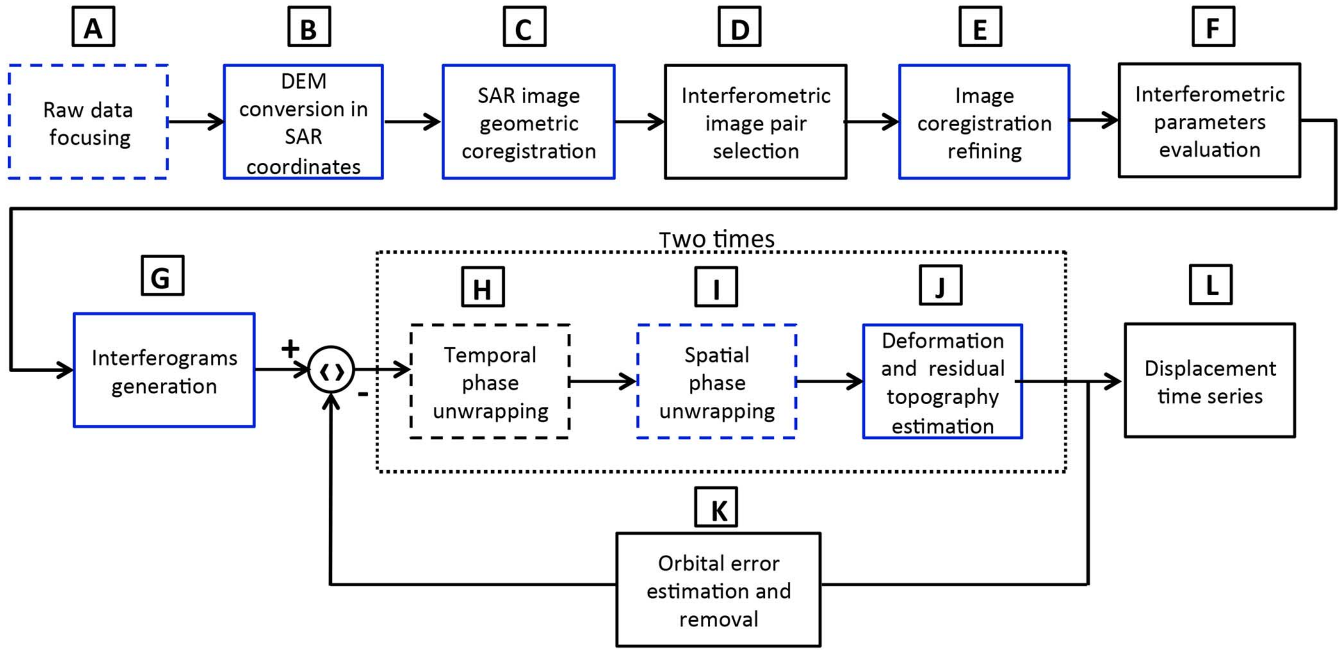


Fig. 1. P-SBAS workflow. Black and blue blocks represent sequential and parallel (from a process-level perspective) processing steps, respectively. Dashed line blocks represent multithreading programmed processing steps.

failure event) and ancillary data inaccuracies (e.g., inaccuracy of orbital information). Step A implements the SAR data focusing operation and consists in transforming the radar raw data into microwave images, often referred to as single look complex (SLC) images. This step has not only a high computational burden because it performs two-dimensional FFTs on large matrices [25] but has also a remarkably significant data flow as it involves many I/O operations (see Table I). Step B performs the digital elevation model (DEM) conversion that consists in referring the elevation profile of the area under consideration with respect to the SAR coordinate reference system [26]. In this step, data matrices that are typically of the order of several GBytes are processed and therefore this step requires a significant high amount of memory and I/O operations, as highlighted in Table I.

Within step C, the SAR image coregistration operation is carried out to refer all the SLCs to the radar geometry of a selected reference "master," via an interpolation procedure [27]. The main limitation of this step is represented by the intensive I/O access and large amount of available memory that are required.

Subsequently, step D performs the identification of interferometric data pairs that are required for the subsequent coregistration refinement performed within the following step E, in which first possible residual subpixel rigid shifts are evaluated, which are, afterward, inverted and used to resample the whole image stack. This step is computationally high demanding because of the applied resampling method that is based on FFTs, which are calculated on large complex matrices.

Within step F, some parameters related to the interferometric data pairs, which needs to be included in the following of the processing chain are selected [25], [26], [28]. Having already both the coregistered images (step C) and the DEM referred to a common radar geometry (as output of step B), the differential interferograms and the corresponding spatial coherence map are generated through step G [28]. These operations require a large amount of RAM memory usage as they are carried out at the SAR images full spatial resolution and in the complex domain. However, the final interferograms are stored in low resolution mode through a complex spatial average (multilook) operation [25]. This procedure, undertaken to mitigate the decorrelation noise affecting the DInSAR interferograms, also drastically reduces the sizes of the final outputs, while the intermediate products remains at full resolution.

The modulo-2π restricted phase of each computed multilook interferogram needs afterward to be "unwrapped" to retrieve the original phase [25]. This procedure is carried out in step H and I by applying the extended minimum cost flow (EMCF) phase unwrapping (PhU) algorithm [29].

The phase unwrapping step is one of the most demanding in terms of memory and CPU usage. Indeed, it deals with wrapped and unwrapped interferogram stacks that are represented by three-dimensional matrices.

A pixel-based inversion of the unwrapped phase system of equations is, afterward, carried out (step J) to retrieve the final deformation time-series. Moreover, in step K, the estimation of possible residual phase artifacts often referred to as "orbital phase ramps" is undertaken by exploiting interferograms. Such phase ramps, that are due to possible orbital inaccuracies, are removed from the wrapped interferograms, implying that another PhU step on the "orbital error free" interferograms has to be performed (second run of steps H, I, and J of Fig. 1). After executing temporal coherence estimation [29] used for the coherent pixel selection, block L provides the final deformation time-series.

TABLE I
P-SBAS ALGORITHM RESOURCE REQUIREMENTS

Step	CPU usage	Maximum RAM usage	I/O operations
A) Raw data focusing	Very high (multithreading)	Medium	Very high
B) DEM conversion	High	Medium	Medium
C) SAR image coregistration	Medium	Medium	Very high
D) Interferometric pair selection	Medium	Low	Medium
E) Image coregistration refining	High	Medium	Very high
F) Interferometric parameters evaluation	High	Medium	Medium
G) Interferograms generation	High	High	Very high
H) Temporal phase unwrapping	Very high (multithreading)	Medium	High
I) Spatial phase unwrapping	Very high (multithreading)	High	High
J) Deformation and residual topography estimation	Medium	Medium	Medium
K) Orbital error estimation and removal	High	Medium	Medium
L) Displacement time series	Medium	Medium	High

Reference values for a standard dataset (64 ENVISAT SAR images, see Section IV). CPU: medium (<100%), high (100%–300%), very high (>300%); RAM: low (<1 GB), medium (1–10 GB), high (>10 GB); I/O: medium (<10 GB), high (tens of GB), very high (hundreds of GB).

In the following, the main critical issues concerning the porting of the P-SBAS algorithm within CC environments are summarized.

First, P-SBAS requires that the execution of the different steps needs to be performed following a well-defined specific order. For instance, SAR image registration can occur only once the focusing operation has been accomplished, while the PhU can be executed only after the differential interferograms sequence generation has been achieved. This rationale motivates why the generated data and products of the SBAS processing chain are highly interconnected and, as a consequence of this data dependency, the joint processing of different step outputs needs at some point to be performed.

Second, many steps of the chain, such as the raw data focusing (step A), image coregistration (steps C and E), interferogram generation (step G) and the unwrapping steps (H and I), are characterized by a large amount of RAM usage, thus implying a proper resources allocation design. Finally, very large amount of data are also involved during the SBAS processing and, therefore, data transfer and I/O issues are two critical factors to be carefully taken into account. The usage of *ad hoc* storage capability and network facility is envisaged to be a viable solution for a proper P-SBAS algorithm exploitation.

III. CLOUD IMPLEMENTATION

Among the different possibilities for integrating an existing application in a CC environment that comprise either adaptation or redesign of the application components [30], we followed the approach of migrating the whole P-SBAS processing chain to the Cloud. This choice has been founded on the criterion of applying a minimum invasiveness policy to the existing application implementation. This is a basic example of migration to the Cloud, where the application is encapsulated in virtual machines (VMs) which run within the CC environment [30].

Thanks to the use of VMs, several benefits can be exploited such as: 1) security and isolation, as services can run in Cloud environment totally independent from each other; 2) ease administration and management, due to the common virtualization layer; 3) disaster recovery, as VMs can be launched in few minutes and furthermore can be cloned and migrated to different locations; and 4) high reliability and load balancing optimization.

The Cloud architecture, depicted in Fig. 2, has been designed taking into account the P-SBAS algorithm analysis carried out in the previous section and consists of several worker nodes (WNs) connected through the network to a master node (MN), which is also used as WN.

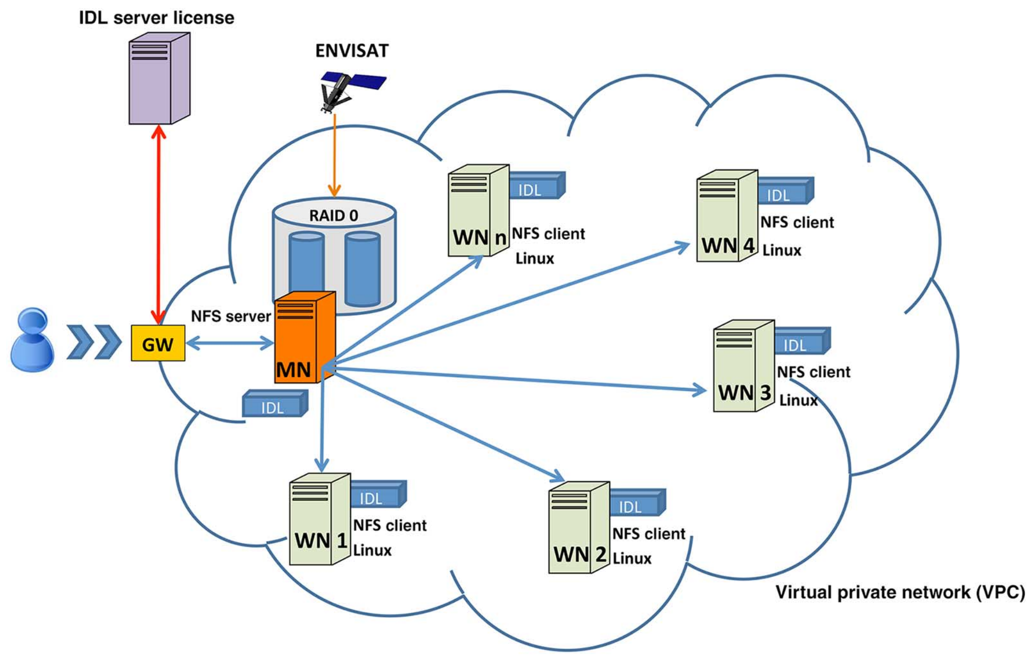


Fig. 2. Cloud architecture for P-SBAS analysis constituted of several WNs which include a MN and a common storage volume in a RAID 0 configuration. Each component is located in the Amazon Web Services Cloud.

As mentioned in the previous section, P-SBAS rationale provides that in many steps of the chain the joint processing of the outputs generated by previous steps needs to be performed. Hence, a common storage is required and therefore a network file system (NFS) has been adopted [31].

Moreover, the majority of the P-SBAS algorithms are developed in the Exelis Interactive Data Language (IDL) [32]: a programming language that is widely used by the scientists who develop algorithms for SAR and DInSAR data processing [14], [33]. IDL is a commercial software and, therefore, each VM running the application requires a license. Hence, an interconnection layer between the IDL License Server (located at the site of CNR-IREA institution) and VMs (located on Cloud) has been implemented. This layer allows us to connect end-points and satisfies the firewall policies adopted by the CNR-IREA institution and the Cloud provider.

The used CC platform is hosted in the Amazon Elastic Compute Cloud (EC2), an infrastructure as a service (IaaS) that is part of AWS Cloud; EC2 has been chosen because it is currently a feature-rich, stable and commercial public Cloud [20]. Moreover, a web service through which users can configure and instantiate a VM image is also available. AWS adopts a virtualization technology that permits to flexibly configure VM instances allowing users to fully set up features such as the number of CPU's cores, the processor type, the memory, the I/O performance, etc. In addition, the operating system and the software that runs on the VM can be customized by the user.

In particular, a Virtual Private Cloud (VPC) has been implemented that is a logically isolated section of the AWS in which resources can be launched in a completely defined virtual network. The easy customization of the network configuration allows users to fully control the virtual networking environment, through IP address range selection, subnet creation, routing tables configuration, network gateways, and multiple layers of security (security groups and network access control lists). Thanks to this customization, the interconnection layer for IDL License Server linking and SAR data uploading has been configured with predefined rules to ensure a secure connection between end points and allow only authorized users to share data. Furthermore, the VM images have been configured by exploiting Linux operating system, as required by the P-SBAS algorithm and all the software and libraries needed for the processing (IDL, C++, etc.) have been installed.

Finally, it is worth noting that the Cloud resources provisioning and configuration phases are automatically performed through dedicated scripts written in Linux Bash. Accordingly, the P-SBAS deployment in a different Cloud environment should be rather easy and quick to be carried out because it requires only slight modifications to such scripts in order to exploit the specific Cloud API.

IV. EXPERIMENTAL FRAMEWORK

The aim of this section is to evaluate the scalable performances of the P-SBAS processing chain within a CC environment and compare them to the corresponding results achieved on an in-house cluster located at CNR-IREA institute premises. Such an analysis has been performed by migrating the existing overall P-SBAS application to the selected AWS CC platform and, besides, it has been aimed at identifying the major inefficiency sources that can hamper its performances. It is worth noting that a one-to-one performance comparison is not fully achievable due to the different resources and architectures underlying the Cloud and in-house computing infrastructures. However, the specific solution implemented within the Cloud environment has been targeted to provide the computing

TABLE II
CNR-IREA CLUSTER CONFIGURATION

	CNR-IREA cluster node
Operative system	GNU/Linux 2.6.32
Processor	8-core CPU — 2.6 GHz Intel Xeon E5-2670
RAM per node	384 GB
Network	56 Gb/s infiniband

TABLE III
AWS INSTANCE TYPE CONFIGURATION

	c3.8.xlarge
Architecture	64 bit
Processor	High frequency Intel Xeon E5-2680
vCPU	32
RAM	60 GB
Network	Very high (nominal 10 Gb/s)

resources as much comparable as possible between the two used infrastructures, as discussed in the following.

A. Computational Platform

The experimental analysis, as above mentioned, has been carried out by exploiting both an in-house HPC cluster which is located at CNR-IREA premises and the public AWS Cloud.

The CNR-IREA cluster consists of 16 nodes, each one is equipped with two CPUs (eight-core 2.6 GHz Intel Xeon E5-2670) and 384 GB of RAM (see Table II). The cluster has a storage shared among different nodes that is implemented through NFS and employs a 56-Gb/s InfiniBand interconnection. In particular, each processing node is equipped with a direct attached storage (DAS) system in a RAID 5 configuration which ensures a disk access bandwidth of approximately 300 MB/s.

Concerning the AWS Cloud architecture described in the previous section, it has been implemented by selecting, among the available ones, the c3.8xlarge EC2 instance that has been used for both master and working nodes (see Table III). Indeed, such an instance is equipped with a processor similar to the one that is present within the in-house cluster nodes (high-frequency Intel Xeon E5-2680 v2 processor). It is also worth noting that AWS describes the EC2 instance computing performances in terms of virtual CPUs that are not easily referable to physical CPUs and cores. To make as much as possible comparable the cluster and Cloud analyses, we treated a single AWS instance as a single node and we forced the algorithm multithreading parts to run using the same fixed number of cores in both the cluster and the Cloud cases. Moreover, the selected c3.8xlarge instance has got an amount of RAM sufficient to run the P-SBAS application without incurring in page faulting and is connected with a 10-Gb/s network bandwidth which is the largest available within AWS. The choice of an instance with such a very high network bandwidth has been driven to guarantee high data transfer and I/O performances as much as possible similar to those of the in-house cluster. As a matter of fact, one of the major critical issues of the P-SBAS algorithm is the very large amount of data (in terms of inputs, intermediate products and final results) that are read/written during the overall processing into the common NFS storage; therefore, particular attention has been addressed to the choice of the storage volume type as well as of the network which allows taking full advantage of the storage I/O performance. AWS gives users the possibility to select among different types of storage according to the specific processing needs; in this case the Amazon Elastic Block Store (EBS) has been employed which provides persistent block-level storage volumes to use together with EC2 instances. In particular, the "provisioned IOPS" (SSD) EBS volume type has been selected since it is designed to meet the needs of I/O intensive workloads and can provision up to 4000 IOPS (I/O operation per second) which corresponds to about 128 MB/s of throughput [20]. Moreover, to improve this latter parameter, two EBS volumes have been selected and configured in RAID 0 (Striping) that allows summing the throughput of the two volumes within it, thus achieving in our case a total bandwidth of about 256 MB/s. The 10-Gb/s network is able to support both such a read/write bandwidth and the communications among different nodes and storage (up to a certain number of exploited nodes as widely discussed in the following).

B. Parallel Computing Metrics and Exploited Dataset

In order to quantitatively evaluate the scalable performances of the P-SBAS processing chain, appropriate metrics were adopted. Let N be the number of computing nodes used to solve a problem and T_1 be the execution time of the sequential implementation to solve the same problem, the speedup S_N of a parallel program with parallel execution time T_N is defined as [34]

$$S_N = \frac{T_1}{T_N}. \quad (1)$$

Accordingly, the speedup is a metric that compares the parallel and sequential execution times.

An alternative performance measure for a parallel program is the efficiency [34], [35]

$$\varepsilon = \frac{S_N}{N} \quad (2)$$

which is a measure of the speedup achieved per computing node. It should be noted that an ideal speedup corresponds to a unitary efficiency.

It is worth highlighting that the parallel performance of the P-SBAS algorithm has already been discussed in [14]; it was shown that the major source of efficiency loss is given by the amount of sequential part within the P-SBAS processing

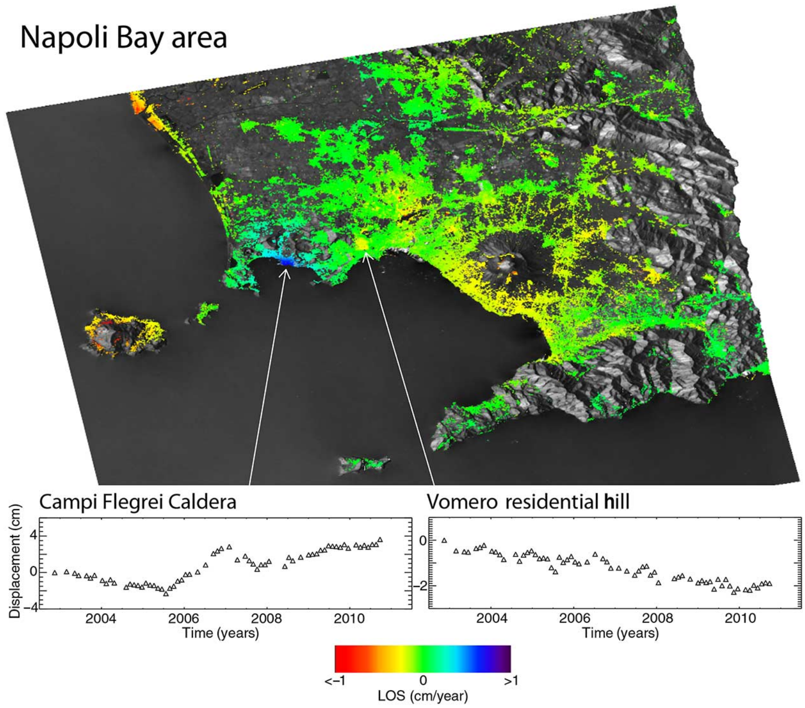


Fig. 3. Mean deformation velocity map of the Napoli Bay area, generated via the P-SBAS processing on AWS Cloud. The graph of the displacement time series relevant to two pixels located in the area of maximum deformation of Campi Flegrei and in the Vomero residential hill are also shown.

chain which, even if liable to a remarkable further reduction, remains essentially noneliminable due to the complex nature of the algorithm.

To quantitatively assess the effect of the serial parts of the algorithm on the attainable speedup, the well-known Amdahl's law is hereafter introduced [35]

$$S_N^A = \frac{1}{f_s + \frac{1-f_s}{N}}, \quad 0 \leq f_s \leq 1 \tag{3}$$

where f_s is the parallel program fraction that has been sequentially executed (sequential fraction) [35]. It is also worth mentioning that the formulation (3) of Amdahl's law does not take into account either the load unbalancing or the data transfer overhead.

The experimental analysis has been performed by processing an interferometric dataset acquired over the Napoli bay, a volcanic and densely urbanized area in Southern Italy that includes the active caldera of Campi Flegrei, the Vesuvius volcano, and the city of Napoli. In particular, we consider the overall time series of ENVISAT acquisitions collected by the ASAR sensor from ascending orbits, which is composed by 64 SAR data. This dataset, which spans the 2003–2010 time interval and covers an area of about 100×100 km^2 that corresponds to an ENVISAT frame, is often used as a benchmark dataset for DInSAR analyses [8], [14], [15], [18]. The selected dataset has been processed by using the P-SBAS algorithm to generate the DInSAR products. In particular, in Fig. 3, the obtained mean deformation velocity map is shown. This map has been geocoded and afterward superimposed on a multilook SAR image of the investigated area. The estimated mean deformation velocity has been only computed in coherent areas; accordingly, areas in which the measurement accuracy is affected by decorrelation noise have been excluded from the false-color map. In particular, it is worth noting that in Fig. 3, a significant deformation pattern corresponding to the area of the Campi Flegrei caldera is clearly shown. Moreover, the computation of the temporal evolution of the detected deformation has also been carried out for each coherent point of the scene. For instance, the chronological sequences of the computed displacement of two specific points (the first located in the maximum deforming area of the Campi Flegrei caldera while the second in the Vomero residential hill which shows a slow down lift movement) are plotted in the insets of Fig. 3. These results are in accordance with ground truth measurements [8], [10].

C. Experimental Results

As previously mentioned, a scalability analysis with respect to the number of exploited computing nodes has been carried out both on the CNR-IREA cluster and on AWS Cloud. Concerning the test performed at the CNR-IREA premises, the speedup depicted as a function of the number of engaged nodes is represented in Fig. 4. Such a plot shows the speedup ideal linear behavior (blue/diamonds), the Amdahl's law (red/squares) and the experimental results achieved on the above mentioned cluster (green/triangles). The Amdahl's law has been evaluated by computing the processing sequential fraction as the ratio between the sum of elapsed times of the P-SBAS sequential parts and the total processing time on a single computing node; it has turned out to be approximately 9% ($f_s = 0.09$). It is evident from Fig. 4 that the achieved speedup is definitely

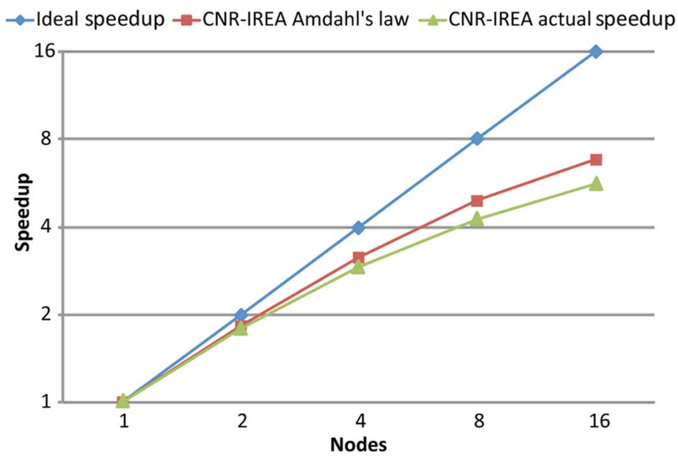


Fig. 4. P-SBAS performances on CNR-IREA cluster: Speedup as a function of the number of nodes N (green/triangles). The ideal achievable speedup (blue/diamonds) and the Amdahl's law behavior (red/squares) are also shown.

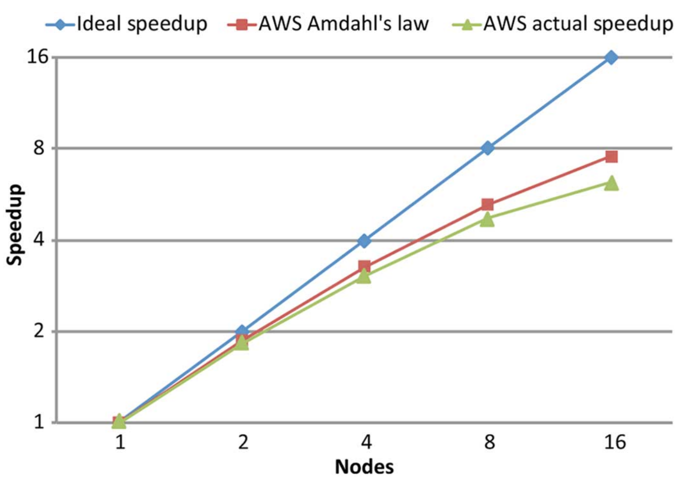


Fig. 5. P-SBAS performances on AWS Cloud: Speedup as a function of the number of nodes N (green/triangles). The ideal achievable speedup (blue/diamonds) and the Amdahl's law behavior (red/squares) are also shown. Note that we refer to a node as to an AWS EC2 instance.

satisfactory; indeed the speedup curve is very close to the Amdahl's law with a slight deviation as approaching 16 nodes. This result reveals that, by exploiting a HPC cluster, all the factors that can hamper the efficiency, such as load unbalancing, communication times and I/O overhead [36], are basically negligible at least up to 16 nodes. Hence the only significant inefficiency source is the P-SBAS sequential fraction that is taken into account by the Amdahl's law and is still subject of further improvements, being some sequential parts of the P-SBAS processing chain currently on progress to be turned into parallel ones.

Fig. 5, instead, shows the speedup that has been evaluated through the scalable analysis carried out within AWS Cloud by exploiting up to 16 instances. In this case, the processing inherent sequential fraction has been estimated to be approximately 7.5% ($f_s = 0.075$), thus being even slightly smaller than in the cluster case. Such a difference is ascribable to the fact that the processing elapsed times strongly depend on the exploited specific computational environments that in the cluster and Cloud cases are different; in particular, the processors of the Cloud nodes are slightly more powerful than those of the cluster nodes (see Tables II and III). As Fig. 5 clearly shows, also in this case, the speedup behavior is very close to the Amdahl's law and it begins to diverge as approaching 16 nodes. In Table IV, the Amdahl's law and actual speedup values evaluated in both the CNR-IREA cluster and AWS Cloud cases are shown. Furthermore, the percentage deviations between the theoretical (Amdahl's law) and experimental behavior are provided, quantitatively confirming the good match between the P-SBAS performance both on cluster and Cloud.

To provide an idea of the times as well as the economical expenses that P-SBAS took to complete the processing, in Table V, the elapsed times relevant to the P-SBAS running tests performed on both CNR-IREA cluster and AWS Cloud by exploiting 1, 2, 4, 8, and 16 nodes are shown together with the corresponding costs relative to the AWS usage. Note that such costs consider both those relevant to the EC2 exploited instances as well as those of the selected storage volumes. On the contrary, the cost of the IDL licenses has not been included because we used those available on the CNR-IREA server with no additional expenses. By exploiting the AWS Cloud, the P-SBAS processing times passed from 41 to less than 7 h when moving from 1 to 16 nodes with a cost of USD 113 and 213, respectively. Table V shows that the P-SBAS parallel performances on Cloud are deemed satisfactory, as they are absolutely comparable with those achieved on the dedicated cluster.

The elapsed times and associated costs shown in Table V, which are related to a maximum of 16 nodes, are undoubtedly adequate when the processing of ENVISAT ASAR datasets on the Cloud is concerned. However, in the perspective of dealing with archives bigger than ENVISAT ones, as in the case of CSK and Sentinel-1 data, the need of exploiting a larger number of nodes is envisaged in order to keep the processing time of the same order of magnitude. In this case, according to the performance behavior represented by the speedup curves of Fig. 5, the discrepancy between the actual and the Amdahl's law speedup curves is expected to increase, thus significantly lowering the efficiency.

In order to identify which is the performance bottleneck when the number of parallel processes increases, some useful metrics, such as the read/write bandwidth and average queue length, provided by the AWS CloudWatch monitoring system [20] have been analyzed in detail.

It turned out that the loss of efficiency relevant to the 16-nodes processing is ascribable to the I/O workload as it understandably increases with the number of parallel processes which concurrently read or write on the common storage volume. Hence, the factor that mainly lowers the P-SBAS scalable performance in our case is essentially the storage volume access bandwidth that is smaller than the network bandwidth (256 MB/s vs. 10 Gb/s).

In the following the performances of two steps of the P-SBAS algorithm, the image coregistration and the spatial phase unwrapping (blocks C and I of Fig. 1), which are characterized by very different I/O workloads, are thoroughly analyzed. To this aim, we considered two metrics: the read/write bandwidth

TABLE IV
VALUES OF THE AMDAHL'S LAW AND EXPERIMENTAL SPEEDUP ON BOTH CNR-IREA CLUSTER AND AWS CLOUD; THEIR PERCENTAGE DEVIATION IS ALSO SHOWN

Number of nodes	1	2	4	8	16
CNR-IREA cluster Amdahl's law	1	1.84	3.15	4.92	6.84
CNR-IREA cluster actual speedup	1	1.80	2.90	4.26	5.60
AWS cloud Amdahl's law	1	1.86	3.27	5.25	7.55
AWS cloud actual speedup	1	1.83	3.07	4.72	6.20
Deviation between Amdahl's law and actual speedup (%) CNR-IREA	0	1.97	7.90	13.46	18.23
Deviation between Amdahl's law and actual speedup (%) AWS	0	1.67	6.16	10.19	17.82

TABLE V
P-SBAS PROCESSING TIMES AND COSTS

Number of exploited nodes	Total elapsed times on CNR-IREA cluster (min)	Total elapsed times on AWS cloud (min)	AWS costs* (USD)
1	2456	2426	113
2	1364	1325	102
4	826	791	119
8	554	514	146
16	415	391	213

*Note that the reported costs include both the instances and the "provisioned IOPS" (SSD) volume usage.

as well as the average length queue, both measured with respect to the EBS storage volume. The former metric is provided by AWS in KiB/s and has been converted in MB/s to be consistent with the known storage volume access bandwidth, while the latter metric represents the number of pending I/O requests and, therefore, it is a latency measure. For the sake of simplicity in Tables VI and VII, only the metrics relevant to the processing carried out with 4, 8, and 16 nodes are reported, which are the most significant to understand how the I/O workloads affects the scalable performances when the number of concurrent processes increases.

The image coregistration has been selected as it is one of the most demanding step of P-SBAS in terms of I/O workload; moreover, it shows the poorest scalable performances among all the P-SBAS algorithm steps. Indeed, accordingly to Table VI, the image coregistration presents an efficiency that strongly degrades while the number of nodes increases. Even if this step could be liable to an improvement from a programming viewpoint in order to reduce its I/O workload, its behavior is helpful to correlate the inadequate scalable performances to the read/write bandwidth saturation. As a matter of fact, such a bandwidth is practically saturated for eight nodes as it already reaches the amount of 230 MB/s of data transfer; this value increases and becomes even 250 MB/s for 16 nodes. The critical value of the average queue length for the employed storage configuration is around 40 counts, indeed this number depends on the I/O capacity of the selected EBS volume [20]. Once again eight nodes are already enough to reach the maximum allowed latency threshold and a greater number of employed nodes would not bring a significant reduction in the elapsed time for this step.

On the contrary, the phase unwrapping step, even if very burdensome from a computational viewpoint, is less heavy for what concerns I/O operations. This step presents satisfactory scalable performances as shown in Table VII, with a 70% efficiency in correspondence with 16 nodes and it would therefore further scale if a greater number of nodes were used. Indeed, in this case, both the read/write bandwidth and the queue length values are evidently below the saturation threshold.

It is worth noting that the scalable performances of the P-SBAS processing chain in the presented Cloud configuration can be further improved by increasing the storage volume access bandwidth by configuring a RAID 0 striping with a greater number of "provisioned IOPS" volumes (up to 12). This would allow us to exploit a larger number of nodes without saturating the storage volume access bandwidth, as long as it is supported by an adequate network bandwidth, with the performance limit given by the Amdahl's law.

TABLE VI
IMAGE COREGISTRATION STEP: EFFICIENCY AND I/O METRICS (RETRIEVED FROM THE CLOUDWATCH MONITORING SYSTEM) AS A FUNCTION OF THE NUMBER OF EXPLOITED NODES ON AWS CLOUD

Image coregistration (nodes)	Elapsed time (min)	Efficiency (%)	Average read/write bandwidth (MB/s)	Average queue length (count)
4	37	81	125	20
8	23	65	230	40
16	19	39	250	50

TABLE VII
PHASE UNWRAPPING STEP: EFFICIENCY AND I/O METRICS (RETRIEVED FROM THE CLOUDWATCH MONITORING SYSTEM) AS A FUNCTION OF THE NUMBER OF EXPLOITED NODES ON AWS CLOUD

Phase unwrapping (nodes)	Elapsed time (min)	Efficiency (%)	Average read/write bandwidth (MB/s)	Average queue length (count)
4	38	89	40	7
8	21	80	65	12
16	12	70	75	20

V. CONCLUSION AND FURTHER DEVELOPMENTS

In this paper, a first assessment of the scalable performances of the P-SBAS DInSAR algorithm migrated to a CC environment has been presented. Such performances have been evaluated, through the processing of a selected dataset acquired by the ENVISAT ASAR system, by investigating the P-SBAS algorithm experimental speedup on an increasing number of nodes (up to 16) and by comparing it with the corresponding theoretical speedup limit given by the Amdahl's law. This analysis has shown a very good match between the Amdahl's law and the experimental speedup behavior that, in the worst case, i.e., in correspondence with the 16-nodes test case, it presents a discrepancy of about 17%. Moreover, the speedup obtained within the AWS CC environment has been evaluated versus the one achieved by exploiting an in-house HPC cluster, showing comparable results and thus proving that the proposed Cloud implementation does not introduce any significant overhead.

In summary, the P-SBAS scalable performances achieved within a CC environment are definitely satisfactory for the considered number of employed nodes. Moreover, the evaluated processing elapsed times show that it is possible to carry out a P-SBAS elaboration of a typical ENVISAT SAR image time series within 7 h with a cost of about USD 200. Hence, the presented P-SBAS Cloud implementation is highly suitable to process image time series acquired by the widely diffused long-term C-band SAR sensors generation, as for the ENVISAT case. This result offers the possibility to simultaneously process many of such ENVISAT SAR datasets by exploiting a sufficiently large number of nodes, which can be only available within CC environments, thus allowing us to process very large portions of the ENVISAT archive in a short time.

However, in the perspective of the need to process massive SAR data amounts, as those already acquired by the COSMO-SkyMed constellation as well as the ones expected from the SENTINEL-1 satellites, the presented P-SBAS Cloud implementation can present a drawback. In this case, indeed, the use of a common NFS-based storage architecture (characterized by the concurrent reading and writing of all the parallel processes on a single shared volume) is expected to become a bottleneck. In fact, when a very large number of computing nodes and, therefore, of parallel processes will be required by the processing, both the disk access and the network bandwidth would become saturated because of inevitable technological limitations. In this regard, a solution based on a distributed file system, properly designed to allocate data to be read or written on local servers (thus minimizing the I/O bottlenecks and the network workload) is a very promising answer to the need of processing very large SAR data flows by scaling on a huge number of nodes. In this context, also the fault tolerance issues need to be addressed in order to provide a P-SBAS implementation which is robust with respect to nodes failures.

It is also worth noting that when a steady data stream is expected, as in the Sentinel-1A case, the convenience to add new scenes without reprocessing the whole stack (append mode) should be assessed. Indeed, a deep analysis aimed at evaluating the cost relevant to store, for an extended time frame, the large additional amount of intermediate products, which are needed to update the P-SBAS deformation time-series with the new acquisitions, is required.

We further remark that, although the presented analysis is fully focused on the AWS Cloud exploitation, the proposed solution is also suitable to be deployed on other Cloud environments. Such a task is already under development; indeed, the porting of the P-SBAS algorithm within the Helix Nebula Cloud environment, which aims at provisioning a sustainable European scientific Cloud [37], is a topic well worthy of further investigation.

This work represents a relevant step toward the challenging EO scenario focused on the joint exploitation of advanced DInSAR techniques and CC platforms for the massive processing of Big SAR Data. This will give the opportunity of generating value added interferometric products on very large scale and in short times, thus broadening the path to a comprehensive understanding of Earth's surface deformation dynamics.

ACKNOWLEDGMENT

The authors would like to thank S. Guarino, F. Parisi, and M.C. Rasulo for their technical support.

REFERENCES

[1] E. Sansosti, F. Casu, M. Manzo, and R. Lanari, "Space-borne radar interferometry techniques for the generation of deformation time series: An advanced tool for Earth's surface displacement analysis," *Geophys. Res. Lett.*, vol. 37, 2010, doi: 10.1029/2010GL044379.

[2] P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti, "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2375–2383, Nov. 2002.

[3] F. Casu, M. Manzo, and R. Lanari, "A quantitative assessment of the SBAS algorithm performance for surface deformation retrieval from DInSAR data," *Remote Sens. Environ.*, vol. 102, pp. 195–210, 2006.

[4] A. Manconi *et al.*, "On the effects of 3-D mechanical heterogeneities at Campi Flegrei caldera, southern Italy," *J. Geophys. Res. Solid Earth*, vol. 115, 2010, doi: 10.1029/2009JB007099.

[5] R. Lanari, F. Casu, M. Manzo, and P. Lundgren, "Application of the SBAS-DInSAR technique to fault creep: A case study of the Hayward fault, California," *Remote Sens. Environ.*, vol. 109, pp. 20–28, 2007.

[6] F. Calò *et al.*, "Enhanced landslide investigations through advanced DInSAR techniques: The Ivancich case study, Assisi, Italy," *Remote Sens. Environ.*, vol. 142, pp. 69–82, 2014.

[7] R. Lanari, P. Lundgren, M. Manzo, and F. Casu, "Satellite radar interferometry time series analysis of surface deformation for Los Angeles, California," *Geophys. Res. Lett.*, vol. 31, 2004, doi: 10.1029/2004GL021294.

[8] M. Bonano, M. Manunta, A. Pepe, L. Paglia, and R. Lanari, "From previous C-band to new X-band SAR systems: Assessment of the DInSAR mapping improvement for deformation time-series retrieval in urban areas," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 1973–1984, Apr. 2013.

[9] A. Pepe, E. Sansosti, P. Berardino, and R. Lanari, "On the generation of ERS/ENVISAT DInSAR time-series via the SBAS technique," *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 3, pp. 265–269, Jul. 2005.

[10] M. Bonano, M. Manunta, M. Marsella, and R. Lanari, "Long-term ERS/ENVISAT deformation time-series generation at full spatial resolution via the extended SBAS technique," *Int. J. Remote Sens.*, vol. 33, pp. 4756–4783, 2012.

[11] S. Salvi *et al.*, "The Sentinel-1 mission for the improvement of the scientific understanding and the operational monitoring of the seismic cycle," *Remote Sens. Environ.*, vol. 120, pp. 164–174, May 2012.

[12] A. Rucci, A. Ferretti, A. M. Guarnieri, and F. Rocca, "Sentinel 1 SAR interferometry applications: The outlook for sub millimeter measurements," *Remote Sens. Environ.*, vol. 120, pp. 156–163, May 2012.

[13] F. De Zan and A. Monti Guarnieri, "TOPSAR: Terrain observation by progressive scans," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2352–2360, Sep. 2006.

[14] F. Casu *et al.*, "SBAS-DInSAR parallel processing for deformation time-series computation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 8, pp. 3285–3296, Aug. 2014.

[15] P. Imperatore *et al.*, "Scalable performance analysis of the parallel SBAS-DInSAR algorithm," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS'14)*, Quebec City, Canada, pp. 350–353, 2014.

[16] S. Elefante *et al.*, "G-POD implementation of the P-SBAS DInSAR algorithm to process big volumes of SAR data," in *Proc. Conf. Big Data from Space (BiDS'14)*, 2014.

[17] P. Rosen, K. Shams, E. Gurrola, B. George, and D. Knight, "InSAR scientific computing environment on the cloud," in *Proc. AGU Conf.*, San Francisco, CA, USA, 2012 [Online]. Available: http://fallmeeting.agu.org/2012/eposters/eposter/in31c-1508/

[18] S. Elefante *et al.*, "SBAS-DInSAR time series generation on cloud computing platforms," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS'13)*, Melbourne, Australia, pp. 274–277, 2013.

[19] M. Hansen *et al.*, "High-resolution global maps of 21st-century forest cover change," *Science*, vol. 342, pp. 850–853, 2013.

[20] Amazon. (2014). *Amazon Web Services* [Online]. Available: http://aws.amazon.com/

[21] Google. (2014). *Google Earth Engine* [Online]. Available: http://www.google.it/intl/it/earth/outreach/tools/earthengine.html

[22] HNX. (2015). *Helix Nebula Marketplace* [Online]. Available: http://hnx.helix-nebula.eu/

[23] K. Yelick, S. Coghlan, B. Draney, and R. S. Canon, "The Magellan report on cloud computing for science," U.S. Dept. Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR), Washington, DC, USA, Dec. 2011.

[24] O. Terzo, L. Mossucca, A. Acquaviva, F. Abate, and R. Provenzano, "A cloud infrastructure for optimization of a massive parallel sequencing workflow," presented at the 12th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput. (CCGrid'12), Ottawa, Canada, 2012, pp. 678–679.

[25] G. Franceschetti and R. Lanari, *Synthetic Aperture Radar Processing*. Boca Raton, FL, USA: CRC Press, 1999.

[26] J. C. Curlander and R. McDonough, *Synthetic Aperture Radar-System and Signal Processing*. Hoboken, NJ, USA: Wiley, 1991.

[27] E. Sansosti, P. Berardino, M. Manunta, F. Serafino, and G. Fornaro, "Geometrical SAR image registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2861–2870, Oct. 2006.

[28] F. Gatelli *et al.*, "The wave-number shift in SAR interferometry," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 855–865, Jul. 1994.

[29] A. Pepe and R. Lanari, "On the extension of the minimum cost flow algorithm for phase unwrapping of multitemporal differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2374–2383, Sep. 2006.

[30] V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch, "How to adapt applications for the cloud environment," *Computing*, vol. 95, pp. 493–535, 2013.

[31] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and implementation of the SUN network filesystem," in *Proc. USENIX Conf.*, pp. 119–130, 1985.

[32] Exelis. (2014). *Exelis Visual Information Solution—Exelis Interactive Data Language (IDL)* [Online]. Available: http://www.exelisvis.com

[33] Globesar AS. (2015). *Globesar—Discovering Global Changes* [Online]. Available: http://www.globesar.com/

[34] H. El-Rewini and M. Abd-El-Barr, *Advanced Computer Architecture and Parallel Processing*. Hoboken, NJ, USA: Wiley, 2005.

[35] G. Hager and G. Wellein, *Introduction to High Performance Computing for Scientists and Engineers*. 2010.

[36] I. Foster, *Designing and Building Parallel Programs*. USA: Addison Wesley, 1995.

[37] Helix Nebula. (2014). *Helix Nebula—The Science Cloud* [Online]. Available: http://www.helix-nebula.eu

Ivana Zinno was born in Naples, Italy, on July 13, 1980. She received the Laurea (*summa cum laude*) degree in telecommunication engineering and the Ph.D. degree in electronic and telecommunication engineering from the University of Naples Federico II, Naples, Italy, in 2008 and 2011, respectively.

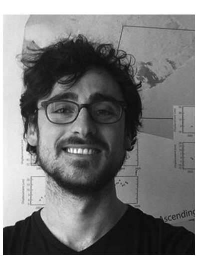
In 2011, she received a grant from the University of Naples to be spent at the Department of Electronic and Telecommunication Engineering for research in the field of remote sensing. Since January 2012, she has been with IREA-CNR, Napoli, Italy, where she is currently a Post-doc Researcher, mainly working on the development of advanced differential SAR interferometry (DInSAR) techniques for deformation time series generation by also exploiting parallel computing platforms. Her research interests include microwave remote sensing, differential SAR interferometry applications for the monitoring of surface displacements and information retrieval from SAR data by exploiting fractal models and techniques.

Stefano Elefante received the Laurea degree (*summa cum laude*) from the University of Naples, Napoli, Italy and the Ph.D. degree in aerospace engineering from the University of Glasgow, Scotland, U.K., in 1997 and 2001, respectively. He holds a PgCert in financial engineering from the Columbia University, New York, NY, USA, and in applied statistics from the Birkbeck College, University of London, London, U.K., since 2008 and 2011, respectively.

He has more than 10 years of diverse professional experience in academia and industry, during which he has been involved in different scientific fields from statistical genetics and financial mathematics to aerospace technologies. From 2005 to 2009, he was a System Analyst and Research Engineer with Boeing Research & Technology Europe, Madrid, Spain, where he conducted extensive research in the field of air traffic management developing and implementing aerospace systems for improving airport and airspace efficiency. Since 2011, he has been holding a research position with IREA-CNR, Napoli, Italy. His research interests include investigating innovative mathematical methodologies for remote sensing applications. He is currently developing novel parallel algorithms for synthetic aperture radar interferometry (InSAR) within cluster, grid and cloud computing environments.

Lorenzo Mossucca was born in Torino, Italy, on February 1982. He received the Laurea degree in computer engineering from the Polytechnic of Turin, Turin, Italy, in 2014.

From 2007, he works as a Researcher with the Istituto Superiore Mario Boella (ISMB) in the Infrastructures and Systems for Advanced Computing (IS4AC) research unit. He has authored and coauthored about 30 international journal, book chapter, and conference papers. He joins in Technical Program Committee and as a Reviewer for many international conferences and editor of the book *Cloud Computing With e-Science Applications* (CRC Press, 2015). His research interests include studies on distributed databases, distributed infrastructures, grid and Cloud Computing, migration of scientific applications to cloud, bioinformatics, and earth sciences fields.



Claudio De Luca was born in Naples, Italy, on July 16, 1987. He received the Laurea degree (110/110) in telecommunication engineering from the University of Naples “Federico II,” Naples, Italy, in 2012. He is currently pursuing the Ph.D. degree in computer and automatic engineering at the same university.

His research interests include cloud computing solution for intensive processing of remote sensing data, development of advanced algorithm for Sentinel-1 SAR and InSAR data processing.

Michele Manunta was born in Cagliari, Italy, in 1975. He received the Laurea degree in electronic engineering and the Ph.D. degree in informatics and electronic engineering from the University of Cagliari, Cagliari, Italy, in 2001 and 2009, respectively.

From 2002, he has been with the Istituto per il Rilevamento Elettromagnetico dell’Ambiente (IREA), an Institute of the Italian National Research Council (CNR), where he currently holds a Researcher position. He was a Visiting Scientist at the Institut Cartografic de Catalunya, Barcelona, Spain, in 2004, and the Rosenstiel School of Marine and Atmospheric Science of the University of Miami in 2006. He has been collaborating in various national and international initiatives for the exploitation of satellite technologies, and in particular of SAR techniques. His research interests include high resolution SAR and DInSAR data processing and application, Cloud and GRID computing exploitation for SAR interferometry applications.

Olivier Terzo was born in Dijon, France, on September 1969. He received the Degree in electrical engineering technology and industrial informatics from the University Institute of Nancy, Nancy, France, the M.Sc. degree in computer engineering and the Ph.D. degree in electronic engineering and communications from the Polytechnic of Turin, Turin, Italy.

He is a Senior Researcher with the Istituto Superiore Mario Boella (ISMB), Torino, Italy, from 2004 to 2009. He worked in the e-security laboratory, mainly with a focus on P2P protocols, encryption on embedded devices, security of routing protocols and activities on grid computing infrastructures, and from 2010 to 2013, he was the Head of the Research Unit Infrastructure Systems for Advanced Computing (IS4AC) at ISMB. Since 2013, he is the Head of Research Area: Advanced Computing and Electromagnetics (ACE), dedicated to the study and implementation of computing infrastructure based on virtual grid and Cloud computing and to the realization of theoretical and experimental activities of antennas, electromagnetic compatibility, and applied electromagnetics. He has authored about 60 papers in conferences, journals and book chapters and he is the editor of the book *Cloud Computing With e-Science Applications* (CRC Press, 2015). His research interests include hybrid private and public cloud distributed infrastructure, grid and virtual grid, mainly activities are on applications integration in cloud environments.

Riccardo Lanari (M’91–SM’01–F’13) graduated in electronic engineering (*summa cum laude*) from the University of Napoli, Federico II, Napoli, in 1989.

In the same year, following a short experience with ITALTEL SISTEMI SPA, he joined IRECE and after Istituto per il Rilevamento Elettromagnetico dell’Ambiente (IREA), a Research Institute of the Italian National Research Council (CNR), Napoli, where, since November 2011, he is the Institute Director. He has lectured in several national and foreign universities and research centers. He was an Adjunct Professor of electrical communication with the l’Università del Sannio, Benevento, Italy, from 2000 to 2003, and from 2000 to 2008, he was the Lecturer of the Synthetic Aperture Radar (SAR) module course of the International Master in Airbone Photogrammetry and Remote Sensing offered by the Institute of Geomatics, Barcelona, Spain. He was a Visiting Scientist at different foreign research institutes, including the Institute of Space and Astronautical Science, Japan, in 1993; German Aerospace Research Establishment (DLR), Germany, in 1991 and 1994; and Jet Propulsion Laboratory, Pasadena, CA, in 1997, 2004, and 2008. He is the holder of two patents, and he has authored or coauthored 80 international journal papers and the book *Synthetic Aperture Radar Processing* (CRC Press, 1999). His research interests include SAR data processing field as well as in SAR interferometry techniques.

Mr. Lanari is a Distinguished Speaker of the Geoscience and Remote Sensing Society of IEEE, and he has served as the Chairman and as a Technical Program Committee Member at several international conferences. Moreover, he acts as a Reviewer of several peer-reviewed international journals. He received a NASA Recognition and a Group Award for the technical developments related to the Shuttle Radar Topography Mission.

Francesco Casu received the Laurea (*summa cum laude*) and Ph.D. degrees in electronic engineering from the University of Cagliari, Cagliari, Italy, in 2003 and 2009, respectively.

Since 2003, he has been with the IREA-CNR, Napoli, Italy, where he currently holds a permanent Researcher position. He was a Visiting Scientist at the University of Texas at Austin, Austin, TX, USA, in 2004, the Jet Propulsion Laboratory, Pasadena, CA, USA, in 2005, and the Department of Geophysics, Stanford University, Stanford, CA, USA, 2009. Moreover, he acts as a Reviewer of several peer-reviewed international journals. His research interests include DInSAR field, multipass interferometry (particularly concerning the improvement of the SBAS-DInSAR algorithm), SBAS-DInSAR measurement assessment, with particular emphasis on novel generation satellite constellations such as COSMO-SkyMed, TerraSAR-X and Sentinel-1, development of DInSAR algorithms for unsupervised processing of huge SAR data archives by exploiting high-performance computing platforms, such as GRID and Cloud computing ones.