

A Cloud Computing Solution for the Efficient Implementation of the P-SBAS DInSAR Approach

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

Abstract—We present an efficient Cloud Computing (CC) implementation of the Parallel Small Baseline Subset (P-SBAS) algorithm, which is an advanced Differential Interferometric Synthetic Aperture Radar (DInSAR) technique for the generation of Earth surface displacement time series through distributed computing infrastructures. The rationale of our approach consists in properly distributing the large data volumes and the processing tasks involved in the P-SBAS chain among the available (virtual and/or physical) computing nodes of the CC infrastructure, so that each one of these elements can concurrently work on data that are physically stored on its own local volume. To do this, both an ad hoc management of the data flow and an appropriate scheduling of the parallel jobs have been also implemented to properly handle the high complexity of the P-SBAS workflow. The proposed solution allows minimizing the overall data transfer and network load, thus improving the P-SBAS efficiency and scalability within the exploited CC environments. The presented P-SBAS implementation has been extensively validated through two experimental analyses, which have been carried out by exploiting the Amazon Web Services (AWS) Elastic Cloud Compute (EC2) resources. The former analysis involves the processing of a large (128 SAR images) COSMO-SkyMed dataset, which has been performed by exploiting up to 64 computing nodes, and is aimed at demonstrating the P-SBAS scalable performances. The latter allows us to show the P-SBAS capability to generate DInSAR results at a regional scale (150 000 km² in Southern California) in a very short time (about 9 h), by simultaneously processing 18 ENVISAT frames that correspond to a total of 741 SAR images, exploiting in parallel 144 AWS computing

nodes. The presented results confirm the effectiveness of the proposed P-SBAS CC solution, which may contribute to further extend the frontiers of the DInSAR investigation at a very large scale.

Index Terms—Cloud Computing (CC), DInSAR, Earth surface deformation, Parallel Small Baseline Subset (P-SBAS).

I. INTRODUCTION

IN the last decades, the Differential Interferometric Synthetic Aperture Radar (DInSAR) techniques have continuously been developed becoming a very important tool for Earth's surface deformation investigation [1], [2], thanks to both their capability to observe very large areas of the globe and their cost effectiveness.

The DInSAR techniques present a very wide range of applications. In particular, they are largely exploited within the natural hazards scenario to study the geophysical processes related to tectonics, volcanoes, and landslides [3]–[8]. Moreover, the DInSAR products are very useful to monitor surface displacements due to anthropogenic actions, as in the cases of groundwater exploitation [9], oil and gas extraction [10], gas capture and storage [11], [12], mining activities [13], [14] tunneling, building and management of dams and flood defenses, and transportation [15], [16].

Basically, the DInSAR technique allows generating spatially dense deformation maps with centimeter to millimeter accuracy by exploiting the phase difference (interferogram) between pairs of complex SAR images, usually referred to as Single Look Complex (SLC) images, relevant to acquisitions gathered at different times, but with nearly the same illumination geometry and from sufficiently close flight tracks, whose separation is typically referred to as baseline [1], [2], [17]. More specifically, the DInSAR methodology analyzes the so-called differential interferograms; they are generated through the difference between an interferogram and its topography-related phase component [17], the latter being calculated by exploiting the sensor orbital information and an external digital elevation model (DEM) of the illuminated scene, properly converted in the SAR coordinates system [17]. For the sake of simplicity, in the following of this paper, the terms interferogram and differential interferogram are considered as synonyms. Note also that the DInSAR techniques permit to estimate the surface displacements component along the radar line of sight (LOS) and they properly work in areas where the computed interferograms are characterized by high coherence, i.e., in zones that are not significantly affected by phase noise effects, usually referred to as decorrelation phenomena [18].

Manuscript received March 22, 2016; revised June 27, 2016; accepted July 23, 2016. Date of publication September 15, 2016; date of current version February 13, 2017. This work was supported in part by MIUR under the project “Progetto Bandiera RITMARE,” in part by the Italian Department of Civil Protection (DPC) (manuscript contents reflect authors' positions that could be different from the DPC official statements), in part by the EPOS-IP project under the H2020 R&I programme, GA 676564, and in part by the I-AMICA (Infrastructure of High Technology for Environmental and Climate Monitoring-PONa3_00363) project of Structural improvement financed under the National Operational Programme for “Research and Competitiveness 2007–2013,” co-funded with European Regional Development Fund and National resources. (Corresponding author: Riccardo Lanari.)

I. Zinno, F. Casu, R. Lanari, and M. Manunta are with the Istituto Per il Rilevamento Elettromagnetico Dell'ambiente-Consiglio Nazionale delle Ricerche (National Research Council), Napoli 80124, Italy (e-mail: zinno.i@irea.cnr.it; casu.f@irea.cnr.it; lanari.r@irea.cnr.it; manunta.m@irea.cnr.it).

C. De Luca is with the Department of Electrical Engineering and Information Technology (DIETI), University of Naples Federico II, Napoli 80138, Italy and also with the Istituto Per il Rilevamento Elettromagnetico Dell'ambiente-Consiglio Nazionale delle Ricerche (National Research Council), Napoli 80124, Italy (e-mail: deluca.c@irea.cnr.it).

S. Elefante is with the Department of Geodesy and Geoinformation, Vienna University of Technology, Wien 1040, Austria (e-mail: stefano.elefante@geo.tuwien.ac.at).

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.2016.2598397

TABLE I
CHARACTERISTICS OF THE SAR RAW DATA ARCHIVES RELEVANT TO
DIFFERENT SPACEBORNE SAR SENSORS (COURTESY OF ASI AND ESA)

Mission	Description	Size	n° products	Time interval	Data supplier
ERS-1/2	L0 (Level 0)	540 TB	180 000	1992–2011	ESA
CSK	L0 (Level 0)	600 TB	620 000	2011–2014	ASI
S1-A	L0 (Level 0)	> 40 TB	42 000	Oct. 2014–Jul. 2015	EU

Originally, the DInSAR methodology has been applied to analyze single deformation episodes such as earthquakes and volcanic unrests [19]–[21]; however, especially thanks to the availability of long SAR data time-series collected in the last decades, the interest of the scientific community has significantly moved toward the study of the temporal evolution of the detected deformations. This is possible through the exploitation of the advanced DInSAR techniques, which properly combine the information available from a set of multitemporal differential interferograms relevant to the area of interest, in order to compute the deformation time series [22].

Among several advanced DInSAR algorithms, a well-known approach is the technique referred to as Small BAseline Subset (SBAS) [23] that allows the computation of displacement time series and the corresponding mean deformation velocity maps in different scenarios (tectonics, volcanoes, landslides, anthropogenic-induced land motions) from a temporal sequence of SAR acquisitions. The SBAS algorithm is, besides, also capable to perform analyses at different spatial scales and with multisensor data [24]–[26].

Currently, the remote sensing scenario is characterized by the huge availability of SAR data acquired during the last 20 years, comprising the long-term C-band European Space Agency (ESA) archives (e.g., ERS-1, ERS-2, and ENVISAT), the RADARSAT-1 and RADARSAT-2 C-band data sequences, those provided by the L-band ALOS-1 and ALOS-2 sensors and by the X-band generation of SAR sensors, such as the COSMO-SkyMed (CSK) and TerraSAR-X constellations. Moreover, a massive and ever increasing data flow is going to be further supplied by the SENTINEL-1A and 1B satellites (launched in April 2014 [27] and April 2016, respectively) that operate within the framework of the COPERNICUS program of the European Union [28]. These sensors acquire with the Terrain Observation with Progressive Scans (TOPS) mode, specifically designed for interferometric application [29], which guarantees a very large spatial coverage. In addition, when fully operative¹, the Sentinel-1 constellation will show a revisit time of 6 days, thus continuously providing new large data archives of extended areas on Earth; remarkably, these data are available with a free and open access policy.

In Table I, a synthetic description of the amount of data provided by some of the above-mentioned sensors is given, in order to highlight how fast the SAR data volumes are growing over time. Considering the depicted fast evolution that is

¹Sentinel-1B is expected to complete the commissioning phase by mid-September 2016.

characterizing the DInSAR technology, the development of effective solutions able to properly deal with the transfer, the storage, and, above all, the processing of such a huge SAR data flow is strongly needed.

Within the framework of the advanced DInSAR processing, a parallel algorithmic solution for the SBAS approach, referred to as Parallel Small BAseline Subset (P-SBAS), which implements the complete advanced DInSAR processing chain (from the SAR raw data focusing up to the displacement time-series generation) and is able to exploit distributed computing architectures, has been recently developed [30]. This solution has also been implemented within the European Space Agency (ESA) Grid Processing on Demand (G-POD) environment [31] to make available the SBAS technique for on-demand processing [32].

However, in house or grid resources can be a bottleneck due to their limitedness. This is particularly the case if the provisioning of multiuser services, allowing scientists to process SAR data through advanced DInSAR techniques and to access the achieved results, is envisaged. In this scenario, a key role can be played by the cloud computing (CC) technologies that are well established within the Information and Communication Technologies, with important spin-offs also in the scientific application context [33]–[36]. Note that the exploitation of customized computing infrastructures built up within CC environments can be crucial in many respects; first, because of the practically unlimited data storage and computing facilities they make available; second, for the flexibility they provide, which allows extensive resources optimization. In such a context, some studies on the P-SBAS algorithm migration to the public CC environment Amazon Web Services (AWS) have already been carried out [37], [38]. These works concerned the evaluation of the P-SBAS parallel performances achievable on a CC infrastructure and allowed highlighting the main issues affecting the P-SBAS scalability, i.e., the capability to efficiently use increased processing resources, which are related to the bottlenecks that occur when huge data volumes, which are processed and generated by the algorithm, need to be transferred and shared among the different computing nodes. In particular, very high network and Input/Output (I/O) capabilities are required to sustain good scalable performances, but, when the number of computing nodes exploited for the parallel computation increases, these resources inevitably tend to saturate [37], [38].

In this paper, we present an efficient CC implementation of the P-SBAS algorithm that has been designed to overcome the above-mentioned scalability limitations encountered with the previous Centralized Storage (hereinafter referred to as CS) implementation [37], [38], thus permitting to achieve good scalable performances without requiring very high-performing resources in terms of I/O throughput. The rationale of this novel implementation lies in distributing the data storage involved in the P-SBAS processing among different nodes connected through the Network File System (NFS) protocol [39], instead of centralizing it, thus parallel handling the overall I/O workload. However, such a strategy, which in the following is referred to as P-SBAS NFS Distributed Storage (DS) implementation, requires the design and the development of an ad hoc management of the data-flow

dependencies, as well as a proper job scheduling to handle the high complexity of the P-SBAS workflow.

In order to investigate the scalable capability of the proposed solution, we carried out a thorough experimental study by exploiting the AWS Elastic Cloud Compute (EC2) public platform [40]. In particular, we performed two kinds of analyses. The former, aimed at evaluating the scalable performances achieved with the P-SBAS NFS DS implementation, has been developed in continuation of the outcomes presented in [37] and [38]; in this case, the obtained results show how the new implementation leads to performance improvements even without exploiting machines with very high-performing features and capabilities, thus also permitting a significant reduction of the corresponding costs relevant to the P-SBAS processing. The latter analysis, besides, has been targeted to show the results of a large-scale experiment involving 741 ENVISAT-ASAR acquisitions that have been processed through the presented P-SBAS NFS DS implementation by exploiting a large number of computing nodes (144 instances) provided by the AWS environment.

II. P-SBAS WORKFLOW AND DATA DEPENDENCY

The SBAS algorithm allows us to generate Earth's surface displacement time series and the corresponding mean velocity maps, and, as previously mentioned, an efficient implementation of the SBAS approach, referred to as P-SBAS, has been recently presented [30]. In particular, the P-SBAS algorithm is capable to exploit distributed computing infrastructures by taking advantage of both multinode and multicore architectures to process huge amount of data in an efficient way.

More specifically, the P-SBAS technique is an advanced DInSAR processing chain consisting of several steps starting with the raw SAR data (Level 0) extraction and ending with the generation of displacement time series and mean deformation velocity maps. A thorough description of the P-SBAS workflow has been provided in [30] and it is clearly outside the scope of this work. In this section, instead, we concentrate on the flow of data and information through the P-SBAS workflow, particularly focusing on their amount and volumes, with the main scope of highlighting the data dependencies among the different P-SBAS steps. Such an analysis permits us to identify the main criticisms and possible bottlenecks throughout the processing chain that are strictly linked to the data transfer and sharing.

Let us start our discussion by considering the input of the P-SBAS algorithm: the N SAR raw data acquisitions (level 0) and the orbital information associated to each image, together with the DEM information of the investigated area, as depicted in Fig. 1. The SAR raw data represent the main input to run the DInSAR processing since they contain the radar echoes received by the satellite sensor expressed as complex values. Typical SAR raw data file sizes range from hundreds of MB (e.g., for the ENVISAT-ASAR standard frames) up to several GB (as for the case of the CSK and Sentinel acquisitions). Orbital information consists in the satellite state vectors and attitudes that are usually provided as acquisition metadata, thus not representing a large data volume. Finally, the DEM of the investigated area is usually the result of the mosaicking of different DEM patches

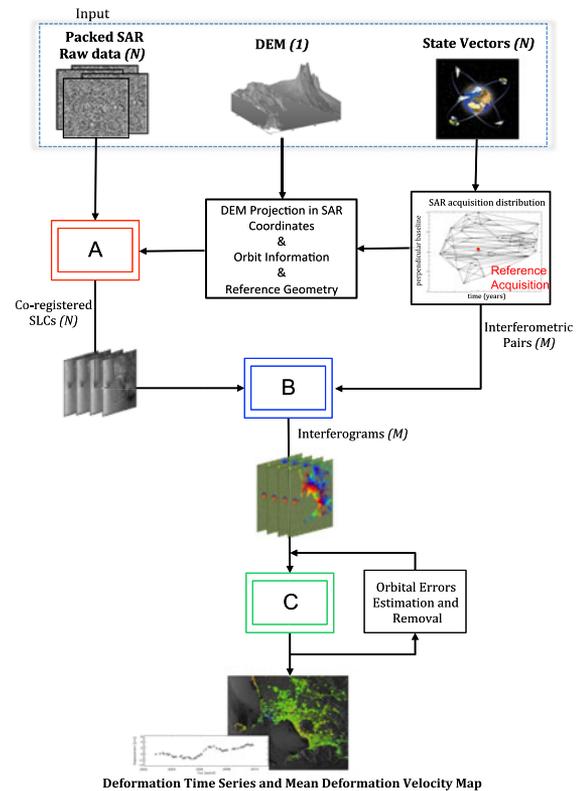


Fig. 1. P-SBAS high-level workflow; the blocks A (SLCs Co-registration), B (Interferogram Generation) and C (Deformation Time Series and Mean Deformation Velocity Maps generation) are detailed in the following Figs. 2–4, respectively. Note that within the brackets (\cdot) the cardinality [I , N , M] of the data exploited as input for the P-SBAS processing steps is specified, with $M > N > I$.

that have to be collected together to cover the zone of interest, thus resulting in a single input file. The DEM size is typically below 100 MB, even if it strongly depends on the DEM resolution and area extent; for instance, the file size of a 3-arcsec SRTM patch, which covers an area of 1 by 1 degree, is of about 20 MB.

According to Fig. 1, the orbital information is exploited to identify both the reference SAR geometry (master image acquisition) and the interferometric pairs that, in the SBAS case, are characterized by small spatial and temporal baselines [23]. Moreover, the orbital information together with the DEM needs to be processed in order to transform the latter from the cartographic (or geographic) projection geometry into the radar master image coordinate system. Once converted, the DEM, together with the SAR raw data, enters a processing block (see block A of Fig. 1) aimed at generating the SLC images and co-registering them with respect to the master image geometry. The block A is run several times since it has to be performed for all the input SAR raw data acquired over the same area along time. Once the coregistered SLCs have been generated, they are collected and coupled in the previously identified interferometric pairs in order to generate the differential interferograms (see block B of Fig. 1). Finally, the deformation time series and the mean deformation velocity maps are generated through block C (see Fig. 1).

TABLE II
I/O DATA VOLUMES FOR EACH STEP OF THE P-SBAS WORKFLOW
(SEE FIGS. 3–5)

Step	Reading [bytes]	Writing [bytes]	# of Runs
a1	$S/2$	$S/2$	N
a2	$S/2$	S	N
a3	$6S$	$4S$	N
a4	$9S$	S	N
TOT A	$16 S*N$	$6.5 S*N$	
b1	$2S$	–	$3N$
b2	S	S	N
b3	$6S$	$S/100$	$3N$
TOT B	$25 S*N$	$\sim S*N$	
c1	$3SN/100$	SN	1
c2	$3SN/100$	$3SN/100$	30
c3	SN	$3SN/100$	1
c4	$3SN/100$	$SN/100$	1
TOT C	$\sim S*N$	$\sim S*N$	

Let us now focus more in details on each block of the P-SBAS processing chain to better understand the I/O data flow, the intermediate products generation and the volume of data transferred among the different steps of the chain (see Table II). To make such an analysis independent from the specific SAR sensor, let us consider as a reference the SLC data matrix size, say S (expressed by bytes), and relate to S the volume of the I/O operations of each block. Let us also assume that N is the number of the input SAR raw data, whereas M represents the number of the interferograms. In this analysis, we focus on the main data that are read and written throughout the different steps of the processing chain and on their dependencies: hence, we neglect the exchanges of ancillary information and metadata, whose sizes are several orders of magnitude smaller than S .

Let us start our analysis by considering the data flow concerning the block A of Fig. 1, which is aimed at generating the coregistered SAR images (see its detailed block diagram in Fig. 2). In this case, the block a1 (SAR raw data ingestion) works on a single packed raw dataset (level 0); following the unpacking operation, the SAR raw data are stored and made available to the subsequent block a2 that performs the image focusing. The block a1 typically reads a matrix of $S/2$ bytes² and provides as output a matrix of $S/2$ bytes. In our application, the focusing operation is carried out according to [30] and [42] and generates the SLC SAR image of the current data, together with the information needed by the following steps, such as the Doppler centroid information [17]. Considering only the main I/O operations, block a2 reads a complex matrix of size $S/2$ (raw SAR data) and stores on the disk the SLC matrix of size S . Step a3 receives as input the DEM converted in the SAR reference geometry, corresponding to the coordinates, in a geocentric Cartesian system, of each SAR pixel of the reference master image (3 matrices each of size $2S$ because they are provided as double precision floating point) [43]. In particular, this block computes matrices containing, for each SAR pixel, the

information of the distance between the pixel on the ground and the SAR sensor position (range distance) in the plane perpendicular to sensor flight path (zero Doppler plane), as well as the azimuth distance (in time) between each pixel and a selected reference time along the orbit [43]. These two matrices, referred to as range and azimuth files, have both a size of $2S$. The SLC of the current data (S) as well as the range ($2S$) and azimuth ($2S$) files of both the current and the master acquisitions are the inputs for the co-registration step (see block a4 of Fig. 2). The output of this last step of the block A is the current SLC image (S) resampled with respect to the master acquisition geometry (coregistered SLC). As already said, the block A of Fig. 1 runs N times, i.e., for each one of the N SAR acquisitions included in the considered dataset. Therefore, the total I/O data flow of the block A involves $16S*N$ and $6.5S*N$ bytes of reading/writing operations, respectively.

For what concerns the block B of Fig. 1 (detailed in Fig. 3), it carries out the generation of the differential interferograms. Note that to obtain accurate interferograms, the SLC co-registration step needs to be refined, on a subpixel basis, for each interferometric pair. This operation is carried out within the blocks b1 and b2 of Fig. 3: the former ingests as inputs two coregistered SLCs ($S+S$ bytes) and provides, as output, the residual subpixel rigid shifts between such data, which are saved in a text file of negligible size. The block b1 operates on all the interferometric pairs (M) so that M subpixel shifts are generated. All these outputs are inverted in the block b2 to obtain the residual subpixel shift for each image, which are applied to the coarsely coregistered SLC images generated in block A in order to retrieve the fine coregistered ones. To do this, the block b2 reads S bytes and writes S bytes for each of the N SAR scenes composing the considered temporal sequence. Finally, the block b3 of Fig. 3 computes the corresponding differential interferogram from the fine coregistered SLCs (S bytes for each acquisition) and the corresponding range files generated within block A ($2S$ bytes for each file). It is worth noting that the interferogram is generated after a spatial average (multilook) operation and, therefore, its size is significantly smaller than that of the SLC input data (typically one order of magnitude or more). Accordingly, the input data of b3 correspond to $6S$ bytes, whereas the output data are significantly less than $S/10$ bytes and they can be neglected.

We also remark that the blocks b1 and b3 of Fig. 3 are carried out as many times as the number of interferometric pairs (M) which, in our SBAS implementation, is on average about $3N$ [23], whereas block b2 is executed N times. Therefore, the total I/O data flow related to the block B of Fig. 1 is of about $25S*N$ and $S*N$ for the reading and writing operations, respectively.

Let us now focus on the block C of Fig. 1 that deals with the phase unwrapping (PhU) and the deformation time-series generation steps, as detailed in Fig. 4. In particular, the PhU operation is carried out by applying the Extended Minimum Cost Flow (EMCF) algorithm [44], which consists of the solution of two networks that are defined in the temporal/perpendicular baseline and azimuth/range domains, respectively. We discuss in the following the implementation of the EMCF PhU approach that is described in details in [44]. From the algorithmic point of view, the PhU operation is carried in the blocks c1, c2, and c3,

²The raw data size can vary depending on the specific SAR sensor.

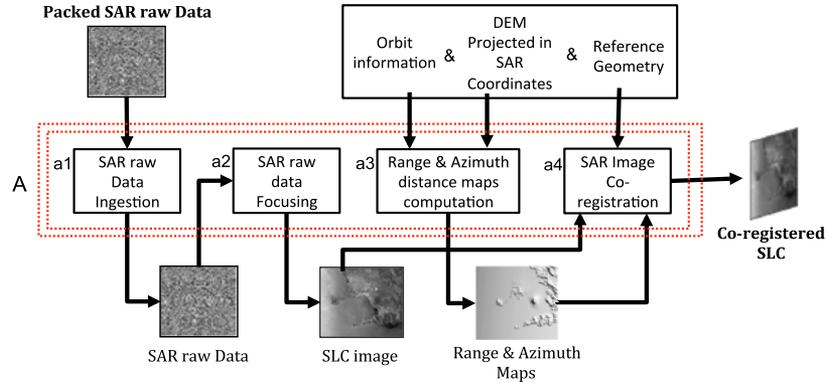


Fig. 2. Detailed workflow of the Block A (SLC Co-registration) of Fig. 1. All the operations performed in blocks a1–a4 are carried out for each SAR raw input data (N times). The dashed red box includes the processing steps with cardinality equal to N .

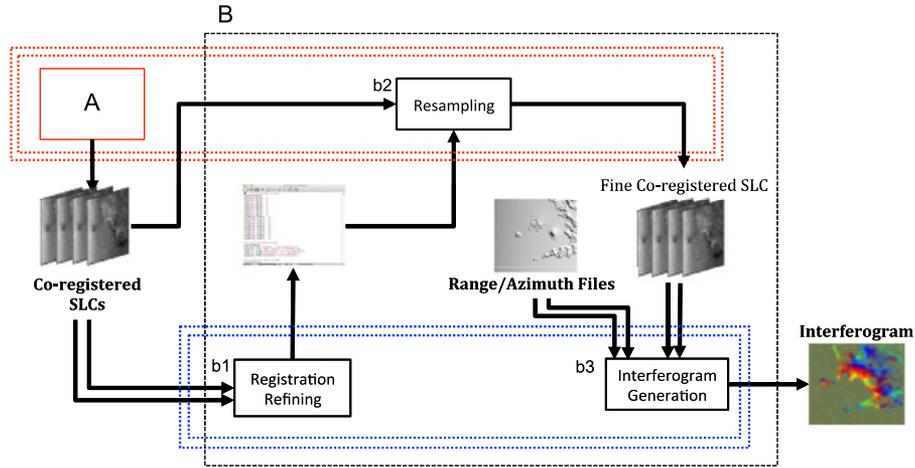


Fig. 3. Detailed workflow of the Block B (Interferogram generation) of Fig. 1. The operations in blocks b1 and b3 are performed for each interferometric pair (M times), whereas those of block b2 are carried out for each SAR acquisitions (N times). The dashed red box encloses the processing steps with cardinality equal to N , whereas the dashed blue box includes the processing steps with cardinality equal to M .

which deal with the temporal and spatial PhU steps (TPhU and SPhU, respectively), both exploiting the basic MCF approach [45], and a weighted combination of the previous results, which is performed by maximizing the temporal coherence [44]. In particular, in order to perform the TPhU step, all the interferograms are simultaneously exploited in block c1, which generates as output a number of three-dimensional (3-D) matrices containing the information needed for the subsequent SPhU operation implemented by the block c2; the size of each one of such 3-D matrices is comparable to the stack of all the interferograms ($\sim 3S*N/100$ bytes) [46]. The number of these 3-D matrices rarely exceeds 30 [46] and, therefore, the output of block c1 can be considered approximately of the order of $S*N$ bytes. Block c2 executes the SPhU and operates on each one of the 3-D matrices resulting from block c1. Since the output of each block c2 is an unwrapped interferogram stack, both the input and the output of block c2 are of the order of $S*N$ bytes. Finally, the block c3 performs the weighted combination of the spatial unwrapped interferometric stacks provided by c2, to achieve the final unwrapped interferograms through the maximization of the temporal coherence. Accordingly, block c3 receives as input $S*N$ data and provides as output the stack of unwrapped

interferograms of size $\sim 3S*N/100$ bytes. Once the unwrapped interferometric stack is generated, the deformation time series are retrieved in block c4 through the SBAS inversion, including the detection and filtering of possible errors related to DEM inaccuracies or atmospheric artifacts [23]; therefore the last block c4 reads a stack of size $\sim 3S*N/100$ bytes and provides as output $\sim S*N/100$ bytes. It is also worth noting that the block C of Fig. 1 is entirely performed twice because it is repeated after the possible orbital phase errors are estimated and filtered out [30].

The overall I/O data flow of block C is about $S*N$ bytes for both reading and writing operations, respectively. We remark that this data flow is mainly due to the blocks c2 (SPhU) that are parallel executed on a 3-D matrix of size $\sim 3S*N/100$ bytes.

III. P-SBAS NFS-BASED DS IMPLEMENTATION

A. Rationale

In this section, we discuss a CC implementation for the P-SBAS algorithm allowing the advanced DInSAR processing of very large SAR datasets, such as those provided by the CSK sensors and, in the near future, by the Sentinel-1 satellites. We remark that, although CC environments offer the possibility of

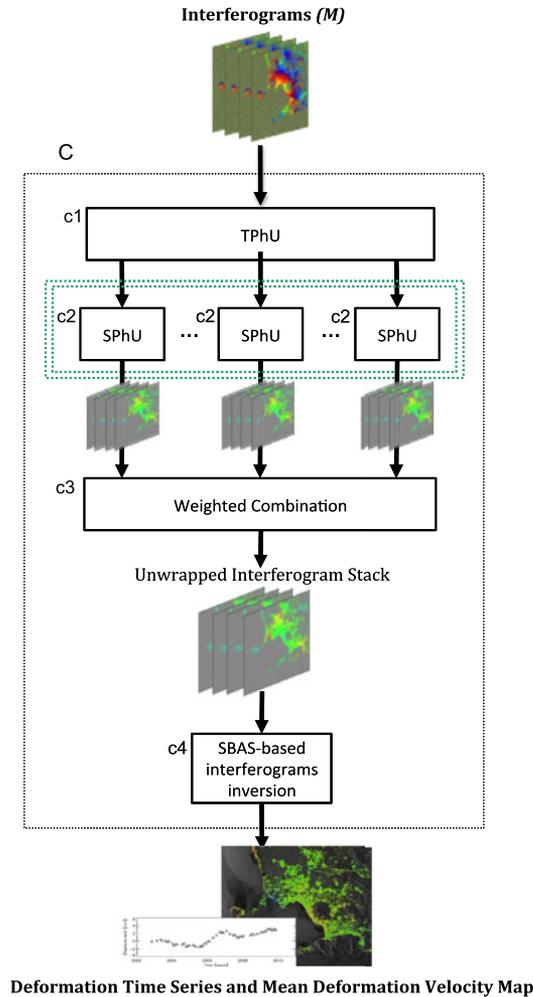


Fig. 4. Detailed workflow of the Block C (deformation time-series generation and mean deformation velocity map generation) of Fig. 1. The operations of blocks $c2$ are performed for each interferogram stack generated by block $c1$, whose cardinality is represented by the dashed green box.

using a practically infinite number of computing elements, a well-working and cost-effective solution can be achieved only if the resources utilization is maximized. Consequently, an efficient CC implementation should exhibit a good scalability in the achieved computing performances, when the number of employed computing nodes increases. This is not a trivial issue because, as also demonstrated by the available literature on this subject [33], [47], [48], the achievement of high scalable performances on CC platforms is a challenging task, especially for scientific applications involving huge data volumes.

The P-SBAS algorithm, as described in the previous sections, deals with a continuous processing of large data time series that are shared among different computing nodes. A P-SBAS scalable performance analysis, carried out exploiting the AWS EC2 and concerning a 64 ENVISAT images dataset, has been thoroughly discussed in [37], [38]. This analysis allowed us also to highlight the main issues affecting the P-SBAS scalability, which are related to the exploitation of a centralized common storage disk, shared via an NFS among the different computing nodes. Indeed, the concurrent access to such a disk from the par-

allel jobs of the P-SBAS algorithm, running on different nodes, makes both the data transfer and the I/O workload to increase proportionally with the number of exploited nodes. This means that, although the network and the disk access bandwidth can be highly performing, there will always be a number of parallel jobs leading to saturation. Therefore, the centralized NFS storage solution, even if easily solves the P-SBAS data dependency issue, represents a bottleneck. Indeed, it sets a restriction to the maximum number of nodes that can be exploited and, therefore, to the overall algorithm scalability.

Accordingly, we propose an ad hoc P-SBAS CC solution based on distributing the I/O workload among the different computing nodes, instead of centralizing it on a unique storage. Such a new solution has been designed to minimize the I/O and data transfer among different nodes in order to achieve good scalable performances without necessarily requiring very high performance resources (in terms of network and disk access bandwidth). This also allows us to limit the costs relevant to the resources utilization within CC environments.

To realize the above-mentioned solution, we implemented an architecture in which each node is directly attached to a local disk, which is used to store a defined portion of the data exploited and produced throughout the P-SBAS processing. Such a data distribution permits to comply the requirement of providing high data throughput fully benefiting from the I/O performances of each disk. Moreover, to ensure data transfer minimization, a proper job scheduling has been designed in order to guarantee that the parallel processes of the P-SBAS algorithm will mostly deal with files that are physically stored on their local disks. This strategy will ensure that the network workload and the disk access saturation are strongly reduced. Note that, for the sake of simplicity, in the following, no distinction will be made between the nodes and the corresponding storage volumes (if not differentially specified). Obviously, although significantly cut down by the proposed solution, a data transfer among different nodes is still required in some steps of the P-SBAS algorithm and, therefore, all the node storages of the implemented architecture need to see each other, in particular, we decide to mutually connect them via the NFS. The implemented strategy allows simultaneously speeding up the I/O operations and reducing the network usage, eliminating both the disk access and the network bottlenecks.

Finally, it is worth noting that the sequential steps of the P-SBAS processing store the produced data on a single defined storage volume among the exploited ones, which acts as a main common storage.

A detailed description of both the implemented computing architecture and the adopted scheduling strategy of the proposed P-SBAS CC solution is provided in the following paragraphs.

B. Computing Architecture

This paragraph is aimed at describing the computing architecture that we have designed to implement the previously mentioned P-SBAS NFS-based DS CC solution.

The overall architecture has been realized by exploiting the AWS CC services, which provide on-demand delivery of

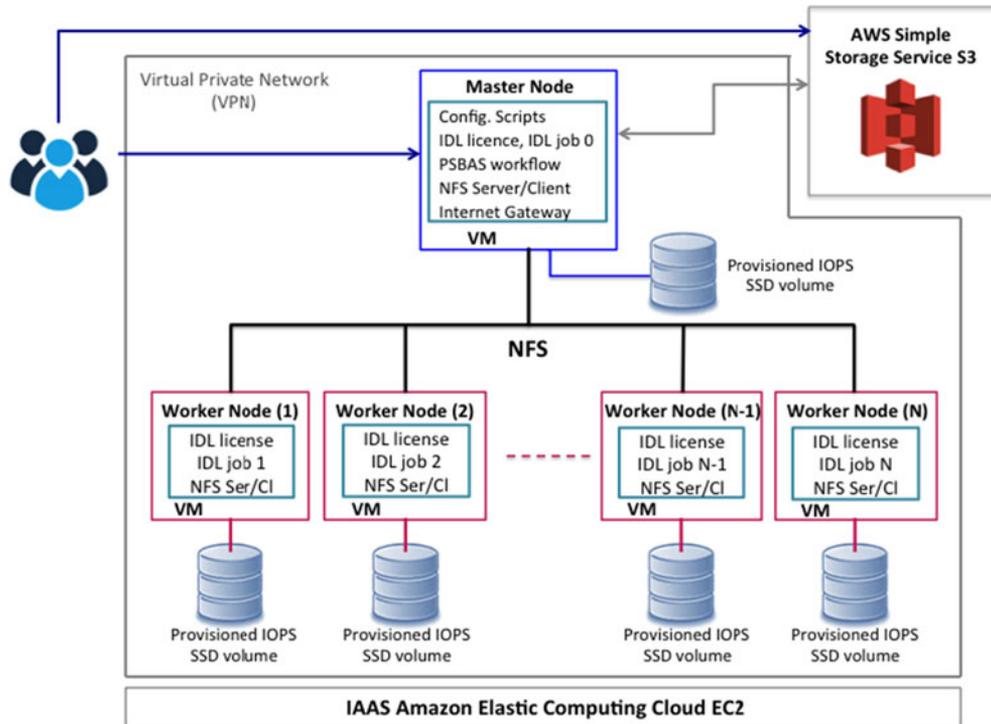


Fig. 5. Computing architecture implemented within the AWS public cloud for the NFS-based distributed storage implementation of the P-SBAS approach. Note that the architecture consists of one master node and several worker nodes (each of these with attached its own local disk), which are all connected via the NFS protocol.

Information Technology resources with a pricing policy based on a “pay per what you use” model [40].

We first implemented an Amazon Virtual Private Cloud (VPC) [49], which is a logically isolated section of the AWS in which resources can be launched in a completely defined virtual private network (VPN). Second, we configured a customized Amazon Machine Image (AMI) containing the operating system (we chose the Fedora distribution [50] but different selection can be carried out), the AWS Command Line Interface (CLI) tool [51], which allows controlling multiple AWS services and automate them through scripts, and the software needed for the P-SBAS processing, such as Interactive Data Language (IDL), Fortran and C codes. Moreover, since the IDL software needs a license file to run, it has been packed directly within such an AMI, which is mounted on all the computing nodes of the implemented platform.

Fig. 5 depicts the computing architecture that we built up, which consists of one master node and N worker nodes. The master node runs the entire P-SBAS workflow by managing the scheduling of the parallel jobs among the worker nodes and it also performs computation for the interferometric processing, equivalently to all the other worker nodes. Moreover, the master node is configured as a network gateway to interface the implemented VPN outward.

Finally, each computing node has got a local disk (or, more generally, a dedicated storage volume) that is directly attached to it, thus ensuring high I/O performances. Such storage volumes are connected to each other via NFS, therefore they are all at the same time NFS server and client. These disks are maintained

active only during the computation lifetime; indeed, at the end of the P-SBAS processing, the final results are transferred to a permanent storage (i.e., AWS Simple Storage Service S3 [52]) and the local disks are deleted when the computing nodes are turned off.

It is worth noting that the creation of the overall computing architecture is automatically carried out through some scripts that we developed by exploiting both the Bash programming language and the AWS CLI tool. The whole computing architecture creation phase lasts few minutes independently of the number of machines that are launched, because the resources are allocated in parallel.

The time needed for transferring the final results, i.e., the deformation time series and the mean velocity maps, depends on the size of the considered dataset and ranges from few up to tens of minutes.

C. Scheduling Strategy and Data Distribution

The design of the scheduling strategy of the P-SBAS NFS-based DS CC solution is not a trivial task. Indeed, as described in Section III, the P-SBAS algorithm is characterized by the succession of parallel and intrinsic sequential steps that, in some points of the processing chain, need to use all the output data resulting from the previous operations. Moreover, as pictorially shown in Fig. 1, blocks A, B, and C operate on input datasets having different cardinalities, which are the number of SAR acquisitions, the number of produced interferograms and the number of 3-D matrices defined by the phase model

used to carry out the TPhU, respectively. Accordingly, different strategies concerning the jobs scheduling and the storage policies have been, therefore, implemented to successfully port the P-SBAS application in a CC environment using the proposed novel approach of partitioning data among different computing nodes and managing the processing in accordance with this rationale.

In particular, for what concerns the job scheduling, two different strategies have been implemented:

- 1) *Static scheduling*: The jobs are assigned to a given node at the beginning of each step. This policy is implemented when only one specific node hosts the data required to perform the step's job. This solution is implemented in the following steps: a1, a2, a3, a4, b2, c2 (see Figs. 2–4).
- 2) *Semidynamic scheduling*: The jobs are assigned during the run execution. This strategy is implemented when the data that are required to perform the step's job are held by more than one node (typically by two nodes such as for the steps dealing with the interferometric data pairs). This solution is implemented in the following steps: b1, b3 (see Fig. 3).

Let us now consider the storage policy that is organized by following three different modalities:

- 1) The data are saved in a unique volume, which is the master's local disk. This solution is implemented when small size files of the order of few MB are generated and this condition usually occurs during the sequential steps, as files containing data parameters and information useful for the whole processing are produced. This solution is implemented in the following steps: the step of the DEM projection in SAR coordinates (see Fig. 1), and within b1, c3, and c4 (see Figs. 3 and 4).
- 2) The data are stored in specific storage volumes mutually connected via NFS, which have been predetermined by the job scheduler. This modality is employed when the data need to be processed by a single node and therefore the data will be stored in its corresponding hard-disk. This solution is implemented in the following steps: a2, a3, a4, b2, b3, and c1 (see Figs. 2–4).
- 3) The data are stored evenly across the nodes' volumes. This solution is implemented when the specific data need to be initially distributed among the computing nodes. Therefore it is only carried out at the beginning of the processing, within step a1.

Following the above discussion on the implemented scheduling and storage solutions, let us now focus on the full description of the overall processing chain.

First of all, the input raw data are stored in the EC2 S3 persistent storage. At this stage, the job scheduler will evenly copy such data to the computing nodes storage volumes. The raw data ingestion step (see block a1 of Fig. 2) is realized in a parallel mode and is fully performed locally in the sense that each computing node will extract only the data that have been physically copied on its hard-disk.

The focusing step (see block a2 of Fig. 2), which generates the SLC SAR images, is executed in a parallel mode as depicted in Fig. 2. As previously explained, the raw data have already been

extracted and stored in the different storage volumes. Hence, each computing node will implement the focusing operation on the data that are physically present on its volume as, in this way, a minimal data transfer through the network is guaranteed.

The DEM conversion in SAR coordinates step (see Fig. 1) occurs in parallel mode and the overall data are currently stored in a single volume.

The coregistration step (see block a4 of Fig. 2) can be executed only after the reference Master acquisition has been chosen. The NFS mutual sharing ensures that every single computing node can access the disk containing all the information relevant to the master image to be able to execute the coregistration.

The image registration refinement and the interferogram generation steps (blocks b1 and b3 of Fig. 3) involve the processing of an interferometric images pair. In this case, the job scheduling has been designed to ensure that a minimal data transfer would occur. To achieve this task, if the two images of an InSAR pair are both stored in the same disk, the node of this volume will process the images. Conversely, in the case in which the images are saved in different volumes, the job scheduling policy will assign the processing pair to one of the node in which one image is stored. This job scheduling occurs in a semidynamic manner in the sense that during the processing, as soon as a node becomes free, the image pair is assigned.

The operation that follows is the PhU one that is very demanding in terms of CPU usage and I/O operations. The sequential step of the temporal PhU (see block c1 of Fig. 4) is designed to write the output data (which are large 3-D matrices containing phase information) directly on the disks on which the spatial PhU (see block c2 of Fig. 4) will be, afterwards, executed. This solution will ensure that the spatial PhU will make use of a minimal network bandwidth and common disk access. The weighted combination step (see block c3 of Fig. 4) is performed in a sequential manner and its output data are saved in the master node storage.

The final step concerning the displacement time series generation (see block c4 of Fig. 4) is executed in parallel but it stores the final matrices containing the displacement time series and the corresponding mean deformation velocity information on the master node storage volume.

The described scheduling strategy has been implemented through Bash scripts launched by the master node that orchestrate the entire P-SBAS workflow. Depending on the specific P-SBAS step to be executed, such scripts launch the sequential job execution on the master node or the jobs to be performed in parallel across the AWS computing nodes. Moreover, for the steps where it is needed, these scripts check the data distribution among the computing nodes and, as a consequence, assign the corresponding parallel jobs; they check the status of the processing by periodically querying all the computing nodes and performing error control, and, finally, they properly manage the synchronization of the different steps of the P-SBAS workflow.

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the scalable performances of the P-SBAS DS implementation showing the improvement that is

TABLE III
AWS INSTANCE TYPE CONFIGURATION

m4.4.xlarge	
Architecture	64 bit
Processor	Intel Xeon E5-2676 v3
vCPU	16
RAM	64 GB
Network	High (around 1 Gb/s)

achieved with respect to the CS solution [37], [38], particularly when the number of nodes that are exploited for the computation increases. Moreover, the following analysis is aimed at demonstrating that the presented P-SBAS implementation shows good parallel performances without needing to exploit very high network and disk access bandwidth performances, thus allowing the reduction of both processing times and costs.

A. Computational Platforms

The computing architectures that we exploited for the comparison between the P-SBAS CC centralized [37], [38] and the presented DS implementations have been built up by exploiting the AWS EC2 resources located in the Ireland region. Among the available EC2 instances, we selected those allowing us to demonstrate how the new implementation overcomes the network bottleneck problem, which characterized the CS implementation discussed in [37] and [38], leading also to a significant cost reduction. In particular, instead of exploiting instances with the maximum network bandwidth available within EC2 (i.e., 10 Gb/s), which were needed in [37] and [38] to reach a satisfactory scalability, we chose machines with a network performance referred to as “high,” which is around 1 Gb/s. This reduction for the requested network bandwidth permitted us to save about 42% of the costs related to the instance payment.

More precisely, for our experimental analysis, both as master and worker nodes, we selected the m4.4xlarge machines because they have got characteristics satisfying the P-SBAS requirements in terms of RAM capacity and processor (number of cores), equivalently to [37] and [38], but with the above-mentioned 1 Gb/s network capability and with a configuration optimized for the Amazon storage volumes (Elastic Block Store, EBS). This means that these instances have dedicated connections to the storage disks to which are attached. In Table III, the characteristics of the m4.4xlarge instances are shown in details.

Note that, throughout our analysis, we refer to an instance as to a single computing node. Moreover, for the CS implementation, as a shared common storage volume, we selected an IOPS (Input Output Operation per Second) provisioned SSD storage volume, ensuring very high I/O performances. In particular, we configured the volume with 16 000 IOPS, which correspond to 250 MB/s disk access bandwidth, which is the maximum throughput achievable by the m4.4xlarge instances. The volume size depends on the overall amount of data produced by the P-SBAS processing and this storage volume has been attached to the master node. Clearly, the storage configuration cannot disregard the network capability because, during the parallel steps

of the P-SBAS processing, the data to be read and written on the common storage are transferred via network; consequently, the actual I/O bandwidth is given by the minimum between the disk access and the network bandwidth.

Concerning the DS implementation, instead, each computing node of the implemented platform had attached its own storage volume. In this case we exploited an IOPS provisioned SSD storage volume with 1000 IOPS for each worker node, and 4000 IOPS for the master node, as this latter one, in several steps of the P-SBAS processing, performs more I/O operations than the other nodes. In these experiments, we carefully set the I/O performances strictly necessary to sustain the I/O workload of each computing node. Moreover, the disk sizes have been selected so that the total amount of data exploited and used by the P-SBAS algorithm could be uniformly distributed among them.

B. Parallel Computing Metrics and Exploited SAR Dataset

In order to quantitatively evaluate the scalable performances of the implemented P-SBAS processing chain, appropriate metrics have been adopted. Let N be the number of computing nodes used to solve a problem and T_1 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 follows [53], [54]:

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

Accordingly, the speedup is a metric that compares the parallel and sequential execution times. Moreover, to quantitatively assess the effect of the serial parts of the algorithm on the attainable speedup, the well-known Amdahl’s law is hereafter considered [54]:

$$S_N = \frac{1}{f_S + \frac{1-f_S}{N}} \quad (2)$$

where $f_S \in [0, 1]$ is the parallel program fraction that has been sequentially executed (sequential fraction) [54]. It is also worth mentioning that the formulation (2) of Amdahl’s law does not take into account either the load unbalancing or the data transfer overhead.

For our experimental analysis, we exploited a CSK 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. The choice of using this CSK dataset is due to the fact that, currently, both in terms of image size and temporal sampling of the sensor acquisition, it is a good example of large InSAR dataset within the remote sensing scenario. In particular, the exploited dataset is composed of 128 acquisitions (for a total amount of about 130 GBytes, concerning the raw data only) relevant to ascending orbits, covering an area of about $40 \times 40 \text{ km}^2$ and spanning the 2012–2014 time interval.

The selected dataset has been processed by using the implemented P-SBAS algorithm in order to generate the DInSAR products. Note that the amount of intermediate and final results produced during our processing is of about 4.5 TBytes.

TABLE IV
P-SBAS PROCESSING TIMES

Number of nodes	CS Implem. parallel fraction processing time [h]	DS Implem. parallel fraction processing time [h]	CS Implem. total processing time [h]	DS Implem. total processing time [h]
8	60.8	57.6	79	75
16	37.2	33.1	55	51
32	27.5	17.6	45	35.3
64	25	13.7	42	30.8

C. Scalable Performance Analysis

This section is aimed at evaluating and comparing the scalable performances of the developed P-SBAS NFS-based DS implementation with those based on the CS implementation [37], [38]. In particular, the goal is to show how the DS solution presented in this paper preserves the scalability when the number of computing nodes increases, differently to the CS one, thus ensuring that the P-SBAS processing times that are achieved within the considered CC environment are significantly improved.

We carried out the scalability analysis by running the P-SBAS algorithm within the AWS CC environment with 8, 16, 32, and 64 m4.4xlarge instances (as said, one instance corresponds to one computing node) both with the centralized and the DS implementations. Moreover, we performed the sequential run (with only one computing node), which essentially coincides for the two implementations, in order to evaluate the Amdahl law (see (2)). The total processing time of the P-SBAS sequential execution is 478.8 h and, starting from this time, we computed the sequential fraction of the processing, which represents the 3.7%. We did not execute the 2 and 4 nodes runs because they would have not been significant for the objective of our analysis, which is the investigation of the P-SBAS performances when the number of computing nodes increases, i.e., when maintaining the scalability becomes a critical issue [37], [38].

In Table IV, the P-SBAS processing times relevant to the above-described experiments are summarized. In particular, we reported both the elapsed times relevant to the parallel fraction of the P-SBAS processing, which actually contain the information about the achieved scalability, and the times of the overall P-SBAS processing, which include also those relevant to the sequential part of the algorithm. It is worth noting that the fourth and the fifth columns of Table IV show the total elapsed times computed from the data extraction until the deformation map generation, including the time to transfer the obtained deformation map to the EC2 S3 persistent storage. Note also that the time to transfer the initial raw data to the master node is not included in Table IV; it depends, indeed, on the external network that is used and in our case it was of about half an hour. When only the parallel part is considered, Table IV shows that the new P-SBAS implementation leads to a noticeable reduction of the elapsed time when it is compared to the use of the centralized solution. A reduction of 5% is achieved when 8 nodes are considered but this amount significantly grows reaching a value of 45% when 64 nodes are taken into account. Such a performance improvement increases when the number of nodes working in

TABLE V
P-SBAS SPEEDUP ANALYSIS WITH 8,16, 32, AND 64 COMPUTING NODES

Number of nodes	Amdahl's law behavior	NFS P-SBAS distributed storage Speedup	NFS P-SBAS centralized storage Speedup
8	6.37	6.35	6.1
16	10.32	9.39	8.71
32	14.90	13.37	10.65
64	18.22	15.53	11.38

parallel increases, because of the progressive saturation of the overall system in terms of bandwidth and disk access relevant to the CS implementation, which does not occur in the DS case. It is important to remark that an elapsed time reduction can be observed only during the parallel steps execution, as the new implementation has no effect on the sequential steps.

Moreover, Fig. 6 shows the Amdahl law behavior (black line), which is the reference of the maximum achievable theoretical speedup performance, and the experimentally measured speedups relevant to the P-SBAS centralized and DS implementations, shown in green and red colors, respectively, for the P-SBAS runs performed with 8, 16, 32, and 64 computing nodes. It is worth noting that, as mentioned in Section III, the step c2 of the P-SBAS processing chain, due to the nature of the implemented algorithm, is realized to scale up to a maximum of 30 nodes. Consequently, the corresponding processing fraction, which is also repeated twice and therefore represents a nonnegligible portion of the processing chain, cannot benefit from the 64 nodes exploitation. This worsens the overall scalability achieved with 64 nodes, reducing the relative improvement with respect to the 32 nodes experiment. Future developments to overcome this limitation are foreseen. At present, we took into account this issue when we evaluated the 64 node Amdahl's law by considering, in addition to the sequential part of the processing, also the fraction with the limited scalability according to the following formulation:

$$S_{64}^A = \frac{1}{f_s + \frac{f_{c2}}{30} + \frac{1-f_s-f_{c2}}{64}}. \quad (3)$$

In Table V, the values of the speedups represented in Fig. 6 are shown. Comparing the two graphs and the relevant values in Table V, it is evident that, in the case of the CS implementation (green line), the actual speedup begins to slightly diverge from the Amdahl law already in correspondence with the 8 nodes run and it progressively gets worse moving to 64 nodes, as the network bandwidth saturation grows when the number of exploited computing nodes increases. On the contrary, in the case of the DS implementation (red line), the actual speedup follows quite well the Amdahl law behavior, thus exhibiting the good scalability that is fulfilled. It is also worth emphasizing that this new implementation is specifically designed to minimize the P-SBAS algorithm bottleneck.

In Fig. 7, the generated mean deformation velocity map is shown; note that it has been geocoded and afterward superimposed on a multilook SAR image of the investigated area.

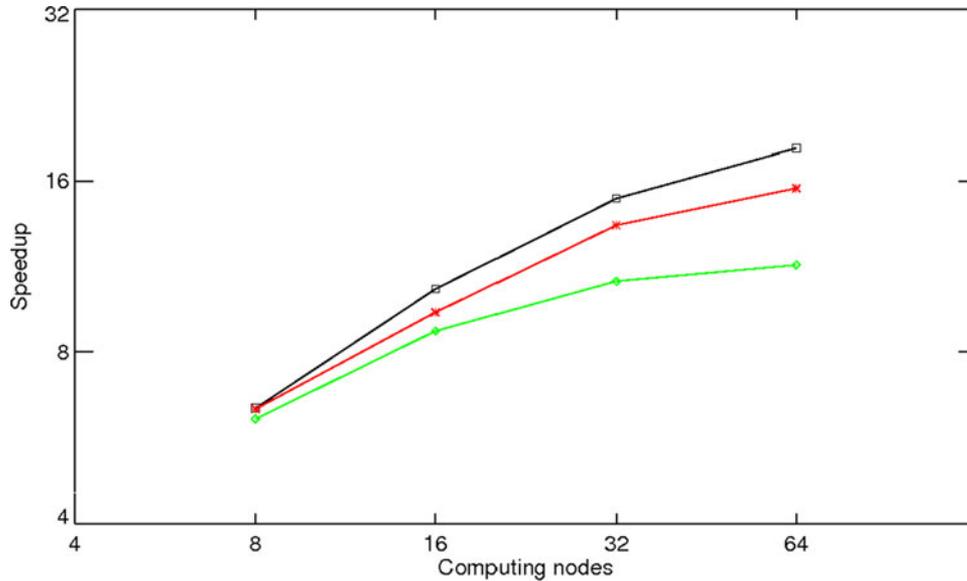


Fig. 6. Speedup analysis with 8, 16, 32, and 64 computing nodes. Amdahl's law behavior (black line) and the speedups relevant to the NFS P-SBAS centralized storage (green line) and distributed storage (red line) implementations are depicted.

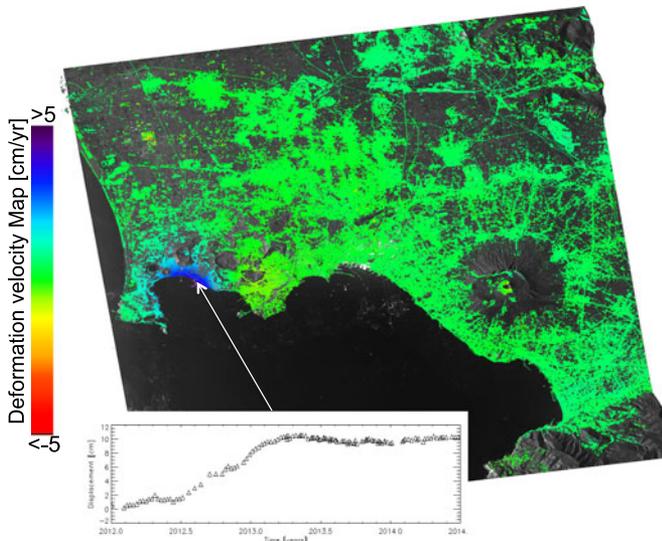


Fig. 7. Mean deformation velocity map of the Napoli Bay (Italy) area generated via the P-SBAS algorithm implemented on the AWS cloud. The inset shows the time series of the computed displacement of a point located in the maximum deformation area of the Campi Flegrei caldera.

We remember that 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 map. It is worth noting in Fig. 7 a significant deformation pattern corresponding to the area of the Campi Flegrei caldera. Moreover, the computation of the temporal evolution of the detected deformation has also been carried out for each coherent point of the scene. We represent in the inset of Fig. 7 the chronological sequence of the computed displacement of a point located in the maximum deformation area of the Campi Flegrei caldera, emphasizing the uplift

phenomena that have characterized this area during the 2012–2014 time period [55]. This area has been extensively analyzed in the past through DInSAR techniques in order to investigate its surface deformation [56]–[60].

Finally, it is worth noting that the presented P-SBAS DS implementation is able to run within any distributing computing infrastructure, in particular, cluster and GRID environments, achieving similar scalable performances if the same hardware requirements of the AWS instances exploited for the analysis discussed in this paragraph are available.

V. A REGION-SCALE P-SBAS PROCESSING CASE STUDY: THE ENVISAT ANALYSIS OVER CALIFORNIA

We show in this section the results of a region-scale DInSAR analysis performed through the presented P-SBAS NFS-based DS implementation. Such an analysis has been carried out by processing in parallel 18 ENVISAT frames acquired over an area of the California region including both the San Diego and Los Angeles metropolitan areas, as well as the San Joaquin Valley region. This region is characterized by a large variety of phenomena inducing ground displacements, such as active seismic fault motions, oil extraction, aquifer water exploitation, etc. In particular, the analyzed dataset is composed of 741 ENVISAT-ASAR scenes acquired from ascending orbits, spanning the 2003–2010 time interval and covering an area of approximately 150 000 km²; the overall input datasets (raw data) size is of about 560 GBytes.

To perform such an extended processing we used 144 AWS instances running concurrently. In particular, we exploited eight instances for processing each ENVISAT frame, i.e., we carried out in parallel the processing of each frame on a subset of eight computing nodes. During the raw data ingestion step of the P-SBAS chain (see block a1 of Fig. 2), some operations of selection and merging of the input raw data are performed

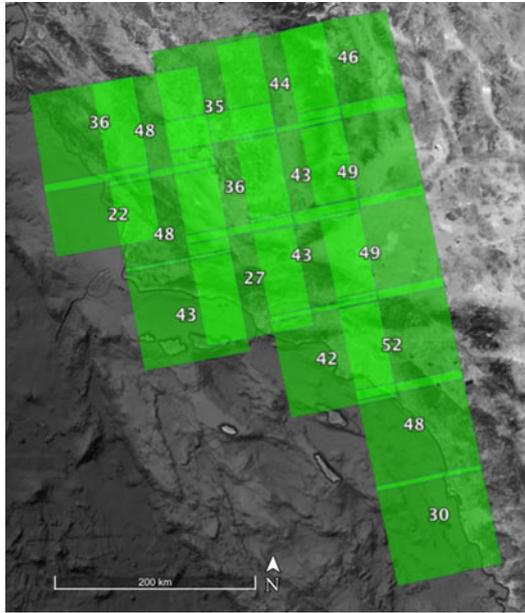


Fig. 8. Spatial coverage of the 18 ENVISAT frames exploited for the regional-scale DInSAR analysis relevant to the investigated California (US) area. For each frame, the number of images that have been effectively processed through the P-SBAS CC implementation is also represented.

TABLE VI
SIZES AND PROCESSING TIMES OF THE PROCESSED ENVISAT ACQUISITIONS FOR THE EXPERIMENT SUMMARIZED IN TABLE IX

Frame number	Number of acquisition	P-SBAS processing time [min/h]
1	30	228 min \approx 3.8 h
2	48	399 min \approx 6.6 h
3	52	483 min \approx 8 h
4	49	484 min \approx 8 h
5	49	473 min \approx 7.9 h
6	46	355 min \approx 5.9 h
7	42	366 min \approx 6.1 h
8	43	413 min \approx 6.9 h
9	43	389 min \approx 6.5 h
10	44	356 min \approx 5.9 h
11	27	242 min \approx 4.0 h
12	36	417 min \approx 6.9 h
13	35	322 min \approx 5.4 h
14	43	271 min \approx 4.5 h
15	48	416 min \approx 6.9 h
16	48	407 min \approx 6.8 h
17	22	191 min \approx 3.2 h
18	36	310 min \approx 5.2 h

in order to extract the correct initial datasets for the following processing. In Fig. 8, we show the spatial coverage of the 18 exploited ENVISAT frames; for each of them, we also indicated the number of images that have been effectively used for the processing after the raw data ingestion step. Note that each of the considered ENVISAT frames is composed on average of about 40 images, the smaller one comprising 22 images, the larger one 52; in any case, in the first column of Table VI the number of acquisitions for each frame is detailed.

We ran all the 144 instances as a part of a unique VPN, connecting them through the NFS protocol. One of the instances

TABLE VII
AWS INSTANCES TYPE CONFIGURATION

	m4.2xlarge	c4.8xlarge
Processor	Intel Xeon E5-2676 v3	Intel Xeon E5-2666 v3
vCPU	8	36
RAM	32 GB	60 GB
Network	High (around 1Gb/s)	10 Gb/s

TABLE VIII
OVERALL TIMES AND COSTS OF THE 18 ENVISAT FRAMES (DETAILED IN TABLE VII) P-SBAS PROCESSING

	Time
Overall Experiment Runtime	\approx 9 h
Overall architecture creation and configuration	10 min
Data transfer from S3 storage to the EC2 instances	\approx 40 min
Overall P-SBAS processing time (18 ENVISAT frames)	\approx 8 h
Average P-SBAS processing time per ENVISAT frame	\approx 6 h
Time to transfer main results to the S3 storage	5 min
Overall Experiment Cost	< 1000 USD
Average cost per ENVISAT frame	\approx 55 USD

was configured as a global master node and managed both the scheduling and the supervision of the 18 concurrent ENVISAT frames processing among all the other instances. Moreover, each one of the subsets of eight computing nodes used to process a single frame had its own local master to manage the parallel jobs of the P-SBAS chain among its eight instances. The global master node periodically checked the state of completion of all the frames processing and, as soon as one of them finished, moved the relevant results to the AWS S3 storage and turned off the corresponding instances.

Since an ENVISAT image is smaller (about four times, in terms of pixels) than a CSK one, we could exploit in this case study instances less performing with respect to those used for the experiments discussed in the previous section, both in terms of RAM capacity and of CPUs number. More specifically, we selected the EC2 m4.2xlarge [40] instance for all the computing nodes except for the global master node, for which a more powerful instance has been chosen (i.e., the c4.8xlarge). Indeed, the global master not only had to periodically access all the other computing nodes to schedule the overall processing jobs as well as monitor its progress status, thus needing a larger network bandwidth, but it also performed computation as a local master of its subset of computing nodes. Note also that the characteristics of the m4.2xlarge and c4.8xlarge instances are compared in Table VII. Finally, for the local storage disks attached to the exploited instances we used 144 Provisioned IOPS SSD volumes each one equipped with 1000 IOPS, for a total of 144 000 Provisioned IOPS.

In Table VIII, we summarize the computing times of the described experiment: note that the achieved result is particularly relevant since the overall experiment required less than 9 h. The creation, i.e., the parallel launch of the 144 EC2 instances, and the whole architecture configuration phases took about 10 min to complete. Then, the input data were moved from the AWS

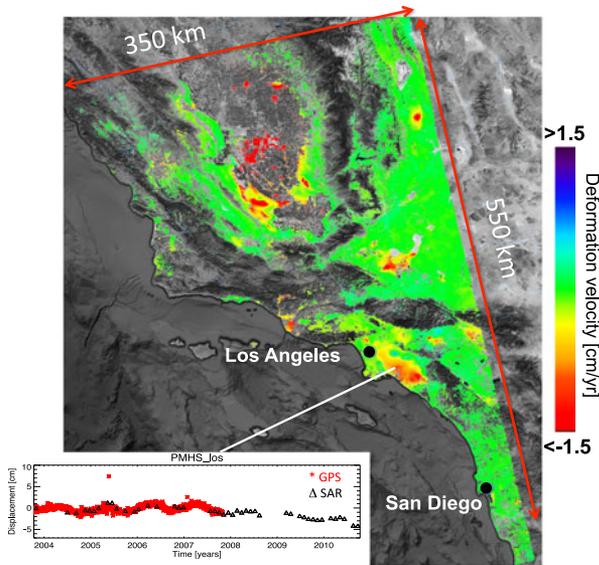


Fig. 9. Mean deformation velocity map relevant to the investigated California (US) region that has been analyzed. The inset represents the comparison between the DInSAR time series (black triangles) computed for a pixel located in the Santa Ana aquifer and the corresponding LOS projected GPS measurements (red stars).

S3 archive to the EC2 instances. In particular, the 18 ENVISAT frames datasets were transferred in parallel to the computing nodes designated as local masters for the corresponding processing. Such a transfer lasted for about 40 min. These actions were automatically carried out through the mentioned Linux Bash scripts that exploit the AWS CLI commands. The P-SBAS processing of all the 18 frames, i.e., for the 741 SAR scenes, lasted 8 h. The average processing time per frame was about 6 h; moreover, the shortest runtime was slightly more than 3 h, the longest one 8 h. The overall cost of the experiment was less than 1000 USD, with an average cost per frame of approximately 55 USD.

Clearly, the different P-SBAS runtimes depend on the size of the considered frame, in terms of the number of acquired images, as well as on the electromagnetic characteristics of the ground scene illuminated by the sensor that determine the number of coherent pixels to be processed in several steps of the P-SBAS processing chain.

In Fig. 9, the obtained final mean deformation velocity map is shown. This map has been achieved by joining the 18 mean deformation velocity maps obtained through the P-SBAS processing of the corresponding ENVISAT frames with a simple postprocessing phase that has been accomplished offline. Note also that we present in Fig. 9 the comparison, for a pixel located in the Santa Ana aquifer area, between the computed DInSAR time series (black triangles) and the corresponding LOS projected GPS measurements (red stars); a very good agreement between the DInSAR and the GPS data is evident.

VI. CONCLUSION

We presented, in this paper, an efficient implementation of the P-SBAS DInSAR algorithm within a CC environment, for the generation of Earth's surface deformation time series and the

corresponding mean deformation velocity maps, starting from SAR raw data sequences. Such a novel implementation has been conceived to ensure good scalable performances within a CC environment, where a very large number of computing nodes and storage volumes can be employed for the parallel SBAS processing. This is, indeed, a time-sensitive issue within the current remote sensing scenario characterized by the availability of very huge SAR data archives whose full exploitation through advanced DInSAR methods requires appropriate solutions in terms of data transfer, storage and, above all, processing.

In particular, the proposed P-SBAS CC implementation has been designed to overcome the scalability bottlenecks, related to the original centralized NFS storage solution [37], [38], which are due to the intensive I/O workload of the P-SBAS algorithm when very large input datasets are concerned. The adopted rationale consists of distributing the data generated and exploited throughout the P-SBAS processing chain among the different storage volumes attached to the computing nodes working concurrently, which are connected via the NFS protocol. Accordingly, we have minimized the data transfer between different nodes, so that the network does not saturate. Such a strategy, however, because of the high complexity of the P-SBAS processing chain, required the design of an ad hoc management of the data-flow dependencies, as well as a proper job scheduling to handle the entire P-SBAS workflow.

A thorough experimental survey has been carried out to show the effectiveness of the novel P-SBAS CC implementation, by exploiting the AWS EC2 public cloud platform. In particular, we carried out two kinds of analyses. The former has been aimed at evaluating the scalable performances achieved by the novel P-SBAS NFS DS implementation and has been conducted by exploiting a large CSK interferometric dataset composed of 128 SAR images. The obtained results show that the presented P-SBAS CC solution exhibits a very good scalability by exploiting up to 64 AWS instances (for a total of 1024 virtual CPUs) with a noticeable improvement with respect to the previous solution based on an NFS CS. Moreover, the proposed implementation allows us to exploit computing resources with moderate network and I/O capabilities in order to fulfill such a good scalability, thus significantly reducing the required costs.

In the latter analysis, we presented the DInSAR processing results relevant to an extended region in California achieved through the presented P-SBAS CC solution. In this case, we processed in parallel 18 ENVISAT frames, corresponding to 741 SAR raw data acquisitions, acquired over an area of 150 000 km² that includes both the Los Angeles and San Diego metropolitan areas, as well as the San Joaquin Valley region, which are interested by several phenomena inducing ground deformation. For this processing, we exploited 144 AWS EC2 instances, each one with its own storage volume attached, for a total of 1152 CPUs and approximately 11 TBytes of storage. The overall experiment lasted approximately 9 h with a cost of less than 1000 USD.

The presented results clearly show that the synergetic exploitation of CC environments and of the existing very large Earth Observation archives opens new intriguing perspectives towards the full exploitation of DInSAR products. Indeed, the

availability of such large-scale data, which are of interest to a large community of researchers working across different scientific disciplines and to many other potential users, is envisaged to be a key issue to improve the comprehension of several geophysical processes affecting the Earth's surface.

ACKNOWLEDGMENT

The authors would like to thank EXELIS VIS for the support given in providing the licenses of the software Interactive Data Language, which has been exploited for the P-SBAS processing. The digital elevation models of the investigated zone were acquired through the SRTM archive.

REFERENCES

- [1] R. Burgmann, P. A. Rosen, and E. J. Fielding, "Synthetic aperture radar interferometry to measure Earth's surface topography and its deformation," *Annu. Rev. Earth Planet. Sci.*, vol. 28, pp. 169–209, May 2000.
- [2] D. Massonnet and K. L. Feigl, "Radar interferometry and its application to changes in the Earth's surface," *Rev. Geophys.*, vol. 36, pp. 441–500, 1998.
- [3] D. Massonnet *et al.*, "The displacement field of the Landers earthquake mapped by radar interferometry," *Nature*, vol. 364, no. 6433, pp. 138–142, Jul. 1993.
- [4] K. Mogi, "Relations between the eruptions of various volcanoes and the deformations of the ground surfaces around them," *Bull. Earthq. Res. Inst.*, vol. 36, pp. 99–134, 1958.
- [5] M. A. Chinnery, "The deformation of the ground around surface faults," *Bull. Seismol. Soc. Amer.*, vol. 51, pp. 355–372, 1961.
- [6] W. E. Farrell, "Deformation of the Earth by surface loads," *Rev. Geophys.*, vol. 10, no. 3, pp. 761–797, Aug. 1972.
- [7] Y. Okada, "Surface deformation due to shear and tensile faults in a half-space," *Bull. Seismol. Soc. Amer.*, vol. 75, no. 4, pp. 1135–1154, Aug. 1985.
- [8] R. Lanari *et al.*, "Evidence for a peculiar style of ground deformation inferred at vesuvius volcano," *Geophys. Res. Lett.*, vol. 29, no. 9, pp. 6-1-6-4, 2002. doi:10.1029/2001GL014571.
- [9] P. A. Hsieh, "Deformation-induced changes in hydraulic head during ground-water withdrawal," *Groundwater*, vol. 34, pp. 1082–1089, 1996.
- [10] R. F. Yerkes and R. O. Castle, "Surface deformation associated with oil and gas field operations in the United States," *Land Subsidence*, pp. 55–65, Internat. Assoc. Sci. Hydrology, Gentbrugge, Belgium, 1969, pubs.rsc.org.
- [11] F. M. Orr, Jr., "CO₂ capture and storage: Are we ready?," *Energy Environ. Sci.*, vol. 2, pp. 449–458, Mar. 2009.
- [12] S. Rackley, *Carbon Capture and Storage*. Oxford, U.K.: Butterworth-Heinemann, 2009.
- [13] C. Qingyun, C. Sijing, M. Shixiang, and H. Dade, "Numerical simulation study of ground deformation in underground mining," *Metal Mine*, vol. 6, pp. 19–24, 2004.
- [14] N. Gourmelen, F. Amelung, F. Casu, M. Manzo, and R. Lanari, "Mining-related ground deformation in Crescent Valley, Nevada: Implications for sparse GPS networks," *Geophys. Res. Lett.*, vol. 34, no. 9, May 2007, Art. no. L09309. doi:10.1029/2007GL029427.
- [15] M. P. O' Reilly and B. M. New, "Settlements above tunnels in the United Kingdom—their magnitude and prediction," *Inst. Mining Metallurgy*, 1982, pp. 173–181, trid.trb.org.
- [16] R. W. Clough and R. J. Woodward, "Analysis of embankment stresses and deformations," *J. Soil Mech. Found. Div.*, vol. 93, pp. 529–549, 1967.
- [17] G. Franceschetti and R. Lanari, *Synthetic Aperture Radar Processing*. Boca Raton, FL, USA: CRC Press, Mar. 1999.
- [18] H. A. Zebker and J. Villasenor, "Decorrelation in interferometric radar echoes," *IEEE Trans. Geosci. Remote Sens.*, vol. 30, no. 5, pp. 950–959, Sep. 1992.
- [19] D. Massonnet, P. Briole, and A. Arnaud, "Deflation of Mount Etna monitored by spaceborne radar interferometry," *Nature*, vol. 375, no. 6532, pp. 567–570, Jun. 1995.
- [20] G. Peltzer and P. A. Rosen, "Surface displacement of the 17 May 1993 Eureka Valley earthquake observed by SAR interferometry," *Science*, vol. 268, pp. 1333–1336, Jun. 1995.
- [21] Y. Fialko, M. Simons, and D. Agnew, "The complete (3-D) surface displacement field in the epicentral area of the 1999 Mw 7.1 Hector Mine earthquake, California, from space geodetic observations," *Geophys. Res. Lett.*, vol. 28, pp. 3063–3066, Aug. 2001.
- [22] 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, no. L20305, 2010. <http://onlinelibrary.wiley.com/doi/10.1029/2010GL044379/abstract>
- [23] 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.
- [24] R. Lanari, O. Mora, M. Manunta, J. J. Mallorqui, P. Berardino, and E. Sansosti, "A small-baseline approach for investigating deformations on full-resolution differential SAR interferograms," in *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 7, pp. 1377–1386, Jul. 2004. doi: 10.1109/TGRS.2004.828196
- [25] 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.
- [26] 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.
- [27] S. Salvi, S. Stramondo, G. J. Funning, A. Ferretti, F. Sarti, and A. Mouratidis, "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.
- [28] Copernicus—The European Earth Observation Programme. [Online]. Available: <http://www.copernicus.eu>
- [29] F. De Zan and A. M. Monti Guarnieri, "TOPSAR: Terrain observation by progressive scans," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2352–2360, Sep. 2006.
- [30] F. Casu *et al.*, "SBAS-DInSAR parallel processing for deformation time-series computation," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 7, no. 8, pp. 3285–3296, Aug. 2014.
- [31] ESA Grid Processing on Demand. [Online]. Available: <https://gpod.eo.esa.int>
- [32] C. De Luca *et al.*, "An on-demand web tool for the unsupervised retrieval of earth's surface deformation from SAR data: The P-SBAS service within the ESA G-POD environment," *Remote Sens.*, vol. 7, pp. 15630–15650, 2015.
- [33] A. Gupta, P. Faraboschi, F. Gioachin, L. V. Kale, R. Kaufmann, B. S. Lee, V. March, D. Milojicic, and C. H. Suen, "Evaluating and improving the performance and scheduling of HPC applications in cloud," in *IEEE Trans. Cloud Comput.*, vol. PP, no. 99, pp. 1–1, doi: 10.1109/TCC.2014.2339858.
- [34] J. J. Rehr, F. D. Vila, J. P. Gardner, L. Svec, and M. Prange "Scientific computing in the cloud," *IEEE Comput. Now*, vol. 12, no. 3, pp. 34–43, May/Jun. 2010.
- [35] S. Hardman *et al.*, "Enabling earth science through cloud computing," in *Proc. IEEE Aerospace Conf.*, Mar. 3–10, 2012, pp. 1–8. doi: 10.1109/AERO.2012.6187356.
- [36] P. Rosen, K. Shams, E. Gurrola, B. George, and D. Knight, "InSAR scientific computing environment on the cloud," presented at the *American Geophysical Union Conf.*, San Francisco, CA, USA, Dec. 3–7, 2012.
- [37] I. Zinno *et al.*, "Cloud computing for earth surface deformation analysis via spaceborne radar imaging: A case study," *IEEE Trans. Cloud Comput.*, vol. 4, no. 1, pp. 104–118, Jan./Mar. 2015. doi: 10.1109/TCC.2015.2440267.
- [38] I. Zinno *et al.*, "A first assessment of the P-SBAS DInSAR algorithm performances within a cloud computing environment," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 8, no. 10, pp. 4675–4686, Oct. 2015. doi:10.1109/JSTARS.2015.2426054.
- [39] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and implementation of the SUN network filesystem," in *Proc. USENIX Conf.*, 1985, pp. 119–130.
- [40] Amazon EC2. [Online]. Available: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
- [41] ESA—Sentinel Online. [Online]. Available: <https://earth.esa.int/web/sentinel/technical-guides/sentinel-1-sar/>
- [42] R. Lanari, "A new method for the compensation of the SAR range cell migration based on the chirp z-transform," in *IEEE Trans. Geosci. Remote Sens.*, vol. 33, no. 5, pp. 1296–1299, Sep. 1995. doi: 10.1109/36.469496.
- [43] 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.

- [44] 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.
- [45] M. Costantini, "A novel phase unwrapping method based on network programming," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 3, pp. 813–821, May 1998.
- [46] P. Imperatore, A. Pepe, and R. Lanari, "Multichannel phase unwrapping: Problem topology and dual-level parallel computational model," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5774–5793, Oct. 2015.
- [47] A. Gupta and Dejan Milojicic, "Evaluation of HPC applications on cloud," *6th Open Cirrus Summit*, 2011, pp. 22–26.
- [48] I. Sadooghi, J. H. Martin, T. Li, K. Brandstatter, Y. Zhao, K. Maheshwari, T. Pais Pitta de Lacerda Ruivo, S. Timm, G. Garzoglio, and I. Raicu, "Understanding the performance and potential of cloud computing for scientific applications," in *IEEE Trans. Cloud Comput.*, vol. PP, no. 99, pp. 1–1, doi:10.1109/TCC.2015.2404821.
- [49] Amazon Virtual Private Cloud VPC. [Online]. Available: <https://aws.amazon.com/it/documentation/vpc/>
- [50] Fedora. [Online]. Available: <https://getfedora.org>
- [51] AWS Command Line Interface CLI. [Online]. Available: <http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html>
- [52] AWS Simple Storage Service S3. [Online]. Available: <https://aws.amazon.com/s3/details/>
- [53] H. El-Rewini and M. Abd-El-Barr, *Advanced Computer Architecture and Parallel Processing*. Hoboken, NJ, USA: Wiley, 2005.
- [54] G. Hager and W. Gerhard, *Introduction to High Performance Computing for Scientists and Engineers*. Boca Raton, FL, USA: CRC Press, 2010.
- [55] L. D'Auria *et al.*, "Magma injection beneath the urban area of Naples: A new mechanism for the 2012–2013 volcanic unrest at Campi Flegrei caldera," *Sci. Rep.*, vol. 5, Apr. 2015, Art. no. 13100. doi:10.1038/srep13100.
- [56] P. Lundgren *et al.*, "Modeling surface deformation observed with synthetic aperture radar interferometry at Campi Flegrei caldera," *J. Geophys. Res. Space Phys.*, vol. 106, pp. 19355–19366, 2001. doi:10.1029/2001JB000194.
- [57] R. Lanari *et al.*, "The use of IFSAR and classical geodetic techniques for caldera unrest episodes: Application to the Campi Flegrei uplift event of 2000," *J. Volcanol. Geothermal Res.*, vol. 133, pp. 247–260, 2004. doi:10.1016/S0377-0273(03)00401-3.
- [58] A. Manconi *et al.*, "The effects of 3-D mechanical heterogeneities at Campi Flegrei caldera, southern Italy," *J. Geophys. Res. Space Phys.*, vol. 115, 2010. <http://onlinelibrary.wiley.com/doi/10.1029/2009JB007099/abstract>
- [59] L. D'Auria, F. Giudicepietro, M. Martini, and R. Lanari, "The 4D imaging of the source of ground deformation at Campi Flegrei caldera (southern Italy)," *J. Geophys. Res. Solid Earth*, vol. 117, 2012, Art. no. B08209. doi: 10.1029/2012JB009181.
- [60] E. Trasatti *et al.*, "The 2004–2006 uplift episode at Campi Flegrei caldera (Italy): Constraints from SBAS-DInSAR ENVISAT data and Bayesian source inference," *Geophys. Res. Lett.*, vol. 35, 2008, Art. no. L07308. doi:10.1029/2007GL033091.



Ivana Zinno was born in Naples, Italy, on July 13, 1980. She received the Laurea degree (*summa cum laude*) in telecommunication engineering and the Ph.D. degree in electronic and telecommunication engineering both 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 the Istituto per il Rilevamento Elettromagnetico dell'Ambiente-Consiglio Nazionale delle Ricerche (National Research Council), Naples, where she currently holds a Researcher position. Her work is currently focused on the development of advanced Differential SAR Interferometry (DInSAR) techniques for surface deformation time series generation by also exploiting distributed computing architectures (GRID and Cloud Computing). Her main research interests include the field of microwave remote sensing; in particular, they concern differential SAR interferometry applications for monitoring the surface displacements, with particular emphasis on novel generation satellite constellations such as COSMO-SkyMed, TerraSAR-X, and Sentinel-1, and, more generally, the information retrieval from SAR data by exploiting fractal models and techniques.



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

Since 2003, he has been with the Istituto per il Rilevamento Elettromagnetico dell'Ambiente-CNR, Napoli, Italy, where he currently holds a permanent Researcher position. He was a Visiting Scientist with the University of Texas at Austin (2004), the Jet Propulsion Laboratory, Pasadena (2005), and the Department of Geophysics at the Stanford University (2009). His main research interests include the DInSAR field, in the multipass interferometry (particularly concerning the improvement of the SBAS-DInSAR algorithm) and in the SBAS-DInSAR measurement assessment, with particular emphasis on novel generation satellite constellations such as COSMO-SkyMed, TerraSAR-X, and Sentinel-1. More recently, he has been involved in the 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. Moreover, he is a Reviewer of several peer-reviewed international journals.



Claudio De Luca was born in Castellammare di Stabia (NA), 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, and the Ph.D. degree in computer and automatic engineering from the Department of Electrical Engineering and Information Technology, University of Naples "Federico II," in 2016.

Since January 2013, he has been a Research Fellow at Istituto per il Rilevamento Elettromagnetico dell'Ambiente-Consiglio Nazionale delle Ricerche (National Research Council), Naples, mainly working on the development of advanced differential SAR interferometry (DInSAR) techniques for deformation time-series generation by also exploiting parallel computing platforms. His main research interests include the field of cloud computing solution for intensive processing of remote sensing data, development of advanced algorithm for Sentinel-1 SAR and InSAR data processing.



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

He has more than 15 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 aerospace systems. From 2011 to 2015, he held a research position with Istituto per il Rilevamento Elettromagnetico dell'Ambiente-Consiglio Nazionale delle Ricerche, Naples to investigate novel parallel algorithms for synthetic aperture radar interferometry (InSAR). He is currently a Project Assistant in the Department of Geodesy and Geoinformation, Technical University of Vienna, Vienna, Austria. His research interests include investigating innovative mathematical methodologies for remote sensing applications and distributed computing for HPC, grid and cloud environments.



Riccardo Lanari (M'91–SM'01–FM'13) received the Graduate degree in electronic engineering (*summa cum laude*) from the University of Napoli Federico II, Napoli, Italy, in 1989.

In 1989, following a short experience at ITALTEL SISTEMI SPA, he joined IRECE and after Istituto per il Rilevamento Elettromagnetico dell'Ambiente, 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 Airborne 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, USA, in 1997, 2004, and 2008. His main research activities are in the SAR data processing field as well as in SAR interferometry techniques; on this topic, he is the holder of two patents, and he has authored or coauthored 80 international journal papers and the book *Synthetic Aperture Radar Processing* (1999, CRC Press).

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.



Michele Manunta was born in Cagliari, Italy, in 1975. He received the Laurea degree in electronic engineering, in 2001, and the Ph.D. degree in informatics and electronic engineering in 2009 from the University of Cagliari, Cagliari, Italy, with a thesis on differential SAR interferometry.

Since 2002, he has been with Istituto per il Rilevamento Elettromagnetico dell'Ambiente, an Institute of the Italian National Research Council (CNR), where he currently holds a Researcher Position. He was a Visiting Scientific with the Institut Cartografic de Catalunya, Spain, in 2004, and the Rosenstiel School of Marine and Atmospheric Science, University of Miami, in 2006. His research interests include the field of high-resolution SAR and DInSAR data processing and application. He particularly works on developing SAR/DInSAR algorithms and techniques for studying deformation affecting terrain surface (such as those produced by landslides, subsidence, volcano activity, and earthquakes) and man-made structures. More recently, his research interests concern Cloud and GRID computing exploitation for SAR interferometry applications. She has been collaborating in various national and international initiatives for the exploitation of satellite technologies, and in particular of SAR techniques.