Distributed Computing on Complex Networks

Francisco Prieto-Castrillo,† Francisco Astillero,‡ and María Botón-Fernández‡

1Research Centre for Environment, Energy and Technology, Spain
2Department of Computer Science and Technology, University of Extremadura, Spain

(Dated: November 18, 2013)

This work considers the problem of finding analytical expressions for the expected values of distributed computing performance metrics when the underlying communication network has a complex structure. Through active probing tests a real distributed computing environment is analysed. From the resulting network, ensembles of synthetic graphs with additional structure are used in Monte Carlo simulations to both validate analytical expressions and explore the performance metrics under different conditions. Computing paradigms with different hierarchical structures in computing services are gauged, fully decentralised (i.e., peer-to-peer) environments providing the best performance. Moreover, it is found that by implementing more intelligent computing services configurations (e.g., betweenness centrality based mappings) and task allocation strategies, significant improvements in the parallel efficiency can be achieved. We qualitatively reproduce results from previous works and provide closed-form solutions for the expected performance metrics linking topological, application structure and allocation parameters when job dependencies and a complex network structure are considered.

PACS numbers: 89.75.-k, 05.10.Ln, 89.20.Ff, 02.50.-r
Keywords: Complex networks, distributed computing, Grid

I. INTRODUCTION

As emphasised by László Barabási [1], once the universal properties of complex networks have been identified, the challenge is to correlate those properties with the processes taking place on the networks. In this regard, an unsolved problem is the effect of the topological characteristics of the underlying communication network on the performance of a distributed computing infrastructure.

From a top-level view of computer function and interconnection, Distributed Computing Infrastructures (DCI) can be understood as the coordination of computing and storage resources through a communication network aimed at solving a task. Depending on the network span (local or global), a business model for user-system interactions and the hierarchical structure of deployed services, different DCI solutions have emerged in the last decades: Grid, Cloud, P2P, Cluster, Utility, Volunteer, Parasitic and Jungle computing. In Ref. [2] a review and classification of these paradigms can be found.

Regarding the network scale, DCIs can be split into two main categories: Cluster Computing, where nodes are usually connected through high bandwidth and low latency (usually fibre channel) links, and the remaining solutions listed above, which involve a geographically dispersed network (e.g., the internet).

From the business model point of view three paradigms coexist: utility computing, volunteer computing and parasitic computing. In the utility model users pay for on-demand delivery of computational or storage services and the infrastructure operates in a transparent manner to the user by aggregating resources in a single system view. This includes Grid and Cloud computing. Grids can be understood as federations of computing clusters that coordinate their resources for a specific scientific community (Virtual Organisation or VO) [3]. Within this federation each cluster is autonomous in their access policies and in the amount and type of resources they offer. Furthermore, resources can be switched on and off dynamically without a centralised control. Grids are often deployed on LAN, WAN, or internet backbone networks at regional, national, or global scales. Cloud is the latest evolution from Grid (still under development) which by means of an intensive use of virtualization technology can offer different computing capacities as a service, usually in a pay-per-use model. There are many similarities among Grid and Cloud Computing (for a complete survey see Ref. [4]). In a rough way it can be stated that Cloud relies on Grid Computing as its backbone and infrastructure support.

Volunteer computing is another collaboration scheme in which users altruistically donate their computing capacities to a project. A problem is split into tasks that can be evaluated independently by computers connected to the internet. Those tasks (work units) are then collected by a central server. A celebrated example is the SETI@home project (see http://www.seti.org). At the opposite end of the spectrum lies the parasitic computing solution [5], which retrieves computing resources without the knowledge of the participating servers.

Large-scale, geographically dispersed DCIs can be classified according to the hierarchical relationships among their components. In the Grid case, the task mapping to clusters (Computing Elements or simply CEs) is coordinated by meta-schedulers (Resource Brokers or simply brokers) acting as servers. CEs are responsible for
managing the nodes (worker nodes in the Grid terminology) where tasks are finally executed. Brokers find CEs according to user specifications and CE’s performance evaluation, availability and other rankings \[3\]. In this regard, if two tasks hosted in two CEs need to communicate, a path passing through the involved brokers is created. Depending on the number of brokers the system can be more or less hierarchical; from a totally centralised solution (single broker) to a decentralised solution in which there is a broker for each CE connected to the same communication node in the network. Cloud virtualization layers introduce additional complexity and the component schemes can be very intricate as the elastic nature of clouds enables the change of resource quantities and characteristics at runtime. However, the Cloud WorkFlow Management System (WFMS) still makes use of schedulers and worker nodes in a hierarchical way \[8\]. Peer-to-peer computing (or simply P2P) lies at the opposite end. The P2P solution replaces the distinct notions of server and client nodes with the notion of peers \[7\]. Hence, P2P systems result in a fully distributed configuration where nodes (i.e., peers) are connected without any other intermediary service. P2P can be thought as the limit case of total decentralisation in Grids. Notice also that volunteer computing can also be thought as the total centralised case of Grids.

Finally, these computing paradigms can be more or less homogeneous in their resources. Recently, the trend appears to be the merging of several solutions resulting in a highly heterogeneous scheme known as jungle computing \[2\].

The understanding and design of DCIs with nontrivial communication topologies and dynamically changing conditions (network traffic, node connections, etc.) demands new methodologies and tools different from those usually applied in High Performance Computing \[3\]. In fact, modelling the interactions among users, application structures (workflows) and DCIs continues to pose a big and ongoing challenge. A promising approach is the integration of concepts and tools from the complex systems theory into DCI models. However, few contributions have tackled the task allocation problem in DCIs from that perspective \[4\]-\[12\].

In 2002 Iamnitchi et al. \[9\] showed how P2P scientific collaboration networks exhibit the small-world property. This work represents one of the first contributions of complex network theory to the DCI task allocation problem through the user-infrastructure interaction standpoint. The highly inspiring article by da Fontoura et al. in 2005 \[10\] analysed the effect of a complex network topology in the efficiency of computing Grids for the first time. There, a homogeneous network with constant latency and no background traffic was considered. In their model the authors used undirected graphs with random, scale-free and several customised topologies with nodes representing processing units and edges representing communication links. In this scheme every node was able to compute tasks and to forward tasks to its neighbour-
vides analytical relationships among infrastructure, application and allocation parameters. Although complex network based models were used, no evidence of why these models should reproduce the referred DCI’s topology is found. Furthermore, assembling jobs into clusters targeted to the same computing resource (i.e., job clustering) has a remarkable effect on the performance, as in this case communication overheads are zeroed. Moreover, well-known workflow management systems and algorithms have been designed and successfully applied to both multiprocessors and distributed computing environments [13, 16–19]. However, to the authors’ knowledge, no previous works have included intertask dependencies effects when analysing DCIs with a complex network topology.

This paper is aimed at finding functional relationships between application performance metrics and the key parameters of different DCI solutions. Inspired by the topology and job services organisation of a real Grid infrastructure (Sec. II), we develop a probabilistic model for application performance metrics in Sec. III In particular we address models for: 1) Simple application workflow schemes (but with job dependence relationships), 2) Parametrized hierarchical structure among DCI services, (3) Probabilistic job allocations based on task clustering. As a result, expressions of the first moments from the resulting order statistics are obtained. These expressions are then validated through Monte Carlo simulations for different DC solutions and configurations (Sec. IV). We conclude and make some remarks on the scope of the addressed model in Sec. V.

II. NETWORK TOMOGRAPHY OF A REAL GRID INFRASTRUCTURE

Global DCIs (Grid, Cloud, P2P and Jungle) overlay computing, storage and other software services over the internet. In this scheme the resulting network consists on nodes belonging to both services and communication layers in a multiplexed-network fashion [3]. In this work we start from the study of a real computing Grid to scan its structural properties. In particular we study the Workload Management System (WMS) of the generic VO Ibergrid in the joint Spanish and Portuguese National Grid Initiatives Grid (ES-NGI) within the European Grid Infrastructure project [20]. This DCI can be understood as a logical network with three node types: brokers, computing elements and communication nodes (routers). The resulting topology is determined by the coupling between the internet topology and the brokers and CEs mapping mechanisms. Hence, it can be expected that the resulting graph preserves some structural properties from the internet.

The internet can be viewed as an evolving network of subnetworks (Autonomous Systems (AS) or domains). Every domain (managed by an administration authority with specific policies) is a network composed by routers, switches and hosts (end computers where application programs run). Hence, the internet topology is modelled at two granularity levels; inter-domain (AS or domain level) and intra-domain (router-level). In Ref. [21] it was claimed that power-law relationships exist at both inter and intra domain levels from the BGP routing tables collected by the route server route-views.oregon-ix.net. However, this assertion has been a subject of great controversy. For instance, in Ref. [22] it is highlighted that derived AS-level topology is not representative for the internet connectivity since at least 20-50% of the physical links were missing in that work. Further studies suggest that this power-law at the AS-level remains when the AS maps are extended [22].

The alleged intra-domain power-law relationship in Ref. [21] derives from the traceroute (see http://www.caida.org/tools) tests collected in [24]. This tool allows tracking packet destinations along a path at the IP layer of the internet. By merging all these paths a reconstruction (i.e., active network tomography) of the network topology is achieved. However, this technique revealed several shortcomings for rendering a reliable topology. A common pitfall is to infer the existence of a physical node that is really a logical entity. In Refs. [25, 26] a survey of biases taking place in traceroute like probes can be found.

In the present work traceroute active probing, similar to that in Ref. [24], is used. The graph reconstruction process is as follows: 1) Retrieve IP address of every broker and CE, 2) For every CE send an agent job with the remaining CEs IP addresses and a traceroute command for those destinations, 3) The resulting set of paths is processed so that each intermediate IP is regarded as a node (it can be a router, a virtual node or even a whole domain) and a link is created for two consecutive IPs. This way local views are obtained by evaluating a certain number of paths from many sources to different destinations. The merging of these views provides a snapshot (the Grid is an evolving entity) of a partial map of the DCI underlying communication network. The obtained graph is of size 189 (161 communication nodes, 23 CEs and 5 brokers) with 477 edges. An average degree $k = 5.03$ and average clustering coefficient $CC = 0.106$ are found. The graph diameter and average shortest path computations renders $l = 4.24$ and $D = 15$ respectively.

With the procedures described in Ref. [27] for dealing with small samples, a power-law fit for an exponent $\alpha = 2.78 \pm 0.13$ for $k_{min} = 5 \pm 1$ is obtained. The goodness-of-fit is evaluated through the $plpva$ function provided in Ref. [28] resulting in a $p-value = 0.1350$. This result, despite the possible poor statistics effect due to the small size of the graph, supports the hypothesis that the graph might be scale-free. However from the discussion above we do not claim that the ES-NGI communication network has in fact a power-law structure. In particular, as we could only resolve a subset of domains (i.e., the IP alias resolution problem [26]), the resulting graph is undoubtedly incomplete, render-
ing only a logical representation of the network. This description falls somewhere in between the inter-domain and intra-domain levels [23]. Although being incomplete, this graph still serves as a seed and reference model for the simulations conducted in this work.

III. A PROBABILISTIC MODEL FOR TASK ALLOCATION IN DISTRIBUTED COMPUTING ENVIRONMENTS

The stochastic task allocation process can be thought as the coupling of three structures: application workflow, computing network and allocation process (see Fig. 1). In this section these structures are formalised. Then, the first moments of DCI performance metrics are derived from the resulting order statistics. Finally we provide analytic expressions for the limit cases considered.

A. Scientific application workflow modelling

Scientific applications are typically modelled as workflows, consisting of tasks, data elements, control sequences and data dependencies [29]. They can be formalised as weighted Directed Acyclic Graphs (DAGs), \( G_J = (J, E_J) \) where \( J \) is a set of \( n_J + 1 \) Jobs \( J = \{J_0, J_1, \ldots, J_{n_J}\} \) and \( E_J \) is a set of \( n_{E_J} \) directed edges \( E_J \subset J \times J \). Every job \( J_k \) has an associated weight computations, the special entry-job \( J_0 \) with \( M_0 = 0 \) and \( O_{0k} = 0, \forall k \) is introduced without loss of generality [17].

DAGs can be parametrized according to different metrics depending on their adjacency matrix \( A_{ij} \) and node/link weights. Roughly, the basic DAG components are the fork and join structures. Given a job \( J_i \), in a fork or join structure the out-degree \( k_i^{\text{out}} \) or the in-degree \( k_i^{\text{in}} \) are non-zero respectively. In this work we assume a minimalistic parametrizable DAG; DAGs are such that \( k_i^{\text{in}, \text{out}} \in \{0, 1\} \), \( \forall i \in I_j \) and with an entry-node \( k_0^{\text{in}} = 0 \), \( k_0^{\text{out}} = n_J \). This results in a set of linked job clusters \( c_{l_k} \) of different sizes \( \omega_k - \omega_{0k} \) (see Fig. 2). Further, we will

![FIG. 1. Stochastic allocation in a distributed computing infrastructure. The overlay of computing services (brokers and CEs represented as black and white circles) deployed on the communication network (routers represented as grey circles) causes different degrees of hierarchy. The overhead transfer costs in every DCI are parametrized through the quantity \( \Phi = 1, 2, 3 \). The network shown represents a real snapshot built from traceroute experiments on the ES-NGI DCI.](image)

![DIRECTED ACYCLIC GRAPH COMPUTING MODES AND NETWORK](image)

![FIG. 2. Simplified parametrizable DAG model for application workflows used in this work. Dependent jobs are arranged into clusters \( c_{l_i} \) of size \( \omega_i - \omega_{0i} \).](image)

![FIG. 2. Simplified parametrizable DAG model for application workflows used in this work. Dependent jobs are arranged into clusters \( c_{l_i} \) of size \( \omega_i - \omega_{0i} \).](image)

assume equally sized jobs (i.e., probe jobs) \( M_k = M, \forall k \) and transfer files \( O_{kl} = o, \forall k, l \). Although this is a non-negligible constraint for representing real DCIs, using probe jobs allows to explicitly highlight the effects of the network topology on the applications performance, which is the major aim of the present contribution.

B. Distributed computing infrastructure model

From the previous discussion in Sec. II a node can represent a router, switch or even an AS. Here we overlook that distinction for the purposes of the present model. The adopted graph \( G_R = (R, E_R) \) consists on a set \( R \) of \( n_R \) communication nodes (routers connected by \( n_{E_R} \) links \( r_{ij} \in E_R \) characterised by a latency \( l_{ij} \) (in seconds) and bandwidth \( bw_{ij} \) (in bytes per second), as it is usual in computer networks analysis [20]. Moreover, we assume bidirectionally in the network \( l_{ij} = l_{ji}, bw_{ij} = bw_{ji} \), \( \forall i, j \) and that it is always possible to find a geodesic path \( P_{ij} = (R_i, \ldots, R_j) \) of length \( d_{ij} \) between any two nodes \( R_i \) and \( R_j \). For every path we define both the path latency \( L_{ij} = \sum_{R_a, R_b \in P_{ij}} l_{ab} \) and path bandwidth \( BW_{ij} = \min_{(R_a, R_b \in P_{ij})} bw_{ab} \). By considering homogeneous latencies \( l_{ij} = LAT \) and bandwidths \( bw_{ij} = BW \) in the network, the path quantities render: \( L_{ij} = LAT \cdot d_{ij} \) and \( BW_{ij} = BW \) respectively. For every graph instance \( g \in G_R \) we use the following definition for the average shortest path

\[
l(g) = \frac{1}{\binom{n}{2}} \sum_{i \neq j} d_{ij}(g),
\]

(3.1)
and $D(g) = \max\{d_{ij}(g)\}$ for the network diameter $D(g)$.

From the inter-router distance distribution $f_d(x) = P[d \leq x]$ (with the random variable $d$ as the geodesic distance between two randomly chosen nodes the graph) and the tail distribution $\hat{f}_d = P[d > x]$, equation (3.1) can also be expressed as

$$l(g) = \sum_{x=0}^{D(g)-1} \hat{f}_d(x).$$

Regarding DCI services, a simple model for workload allocation is assumed. In particular, we do not consider other services such as storage, information, resources discovery, etc. In a task communication process between jobs $J_k$ and $J_l$ with a dependence relationship ($A_{kl} \neq 0$) the communication overhead depends on the DCI solution adopted. In the P2P case CEs transfer files directly (see Fig. 1). However, one or more meta-schedulers (brokers) can also be involved in the process. In the Grid case it is usual to find a hierarchical structure where a single broker covers a wide range of CEs. Other brokers can also be used as a support for redundancy, communication optimisation, failure, or by direct user specifications.

We then characterise distributed computing configurations through the constant $\Phi = 1, 2, 3$. These solutions depend on the number of brokers and CEs deployed (denoted as $n_B$ and $n_C$ respectively) and their connections (Fig. 1). With a slight abuse of the $\delta$ notation we define $\Phi = 3 - \delta_{n_B} - 2\delta_{P2P}$ where $\delta_{P2P} = 1$ if $n_B = n_C$ and both brokers and CEs share the same router (otherwise $\delta_{P2P} = 0$). This parametrizes DCI solutions as follows: 1) P2P with $\Phi = 1$ and $n_B > 1$ (fully decentralised solutions). 2) Total centralisation with $\Phi = 2$, where $n_B = 1$ and at least one CE is connected to a different router. 3) Partially centralised solutions with $\Phi = 3$ for $n_B > 1$ and where at least one CE is connected to a different router than those connected to the brokers.

For every DCI we also define: 1) A set $S = B \times C$ of $n_S$ computing modes composed by a broker set $B = \{B_1, \ldots, B_{n_B}\}$ and a CE set $C = \{C_1, \ldots, C_{n_C}\}$. 2) Two simple (with finite support) random variables $\beta : B \rightarrow R, B_i \mapsto \beta_i$, and $\theta : C \rightarrow R, C_j \mapsto \theta_j$, to model broker and CEs services mappings respectively. Computing modes are denoted as $s_{ij} = (B_i, C_j)$ but in order to lighten notation we eventually set a single index labelling through the bijection $(i, j) \mapsto n_C(i - 1) + j; i = 1, \ldots, n_B; j = 1, \ldots, n_C$.

In our model we consider CEs with infinite processing capacity, no queuing delays and equal processing speeds $V$—measured in Million of Instructions Per Second (MIPS). Infinite processing capacity means that if two tasks arrive at the same CE they are executed in parallel if no precedence relationship exists in the corresponding DAG. This takes our approach slightly away from the pure scheduling models where every task has an assigned order; if two tasks with no precedence relationship arrive at the same host the local scheduler establishes an additional order of execution [12, 14, 17, 22, 51]. In our approach it suffices to comply with the order imposed by the DAG. The effect is equivalent to CEs with infinite processing capacity so that no additional ordering is required. Furthermore, brokers are assumed to hold infinite storage capacity and task forwarding speed; there is no congestion and tasks are efficiently dispatched without introducing any additional delays.

### C. Allocation process and time metrics

Task allocation is modelled by the stochastic process $AP = \{X_k : k \in I_J\}$, where $I_J$ denotes the job index set and every random variable $X_k$ has support $S$ and $x_k \in S$ represents the assigned mode to job $J_k$ (Fig. 1). We also define the function: $\Delta : S \times S \rightarrow \{1, 0\}, (x_k, x_l) \mapsto \Delta_{kl}$, where $\Delta_{kl} = 1$ if $x_k = x_l$ and $\Delta_{kl} = 0$ if $x_k \neq x_l$. Mass probabilities $P[X_k = s_n]$ are denoted as $p_{kn}$, joints $P[X_k = s_n, X_l = s_m]$ as $p_{kn,lm}$ and conditionals $P[X_k = s_n, X_l = s_m] = p_{kn|lm}$. With these elements, a probabilistic model for the conditional probabilities is built (see details in Appendix A). The aim is to provide independence between the events ”$\{x_k = s_n\}$” and ”$\{x_l = s_m\}$” when there is no order relationship between jobs $J_k$ and $J_l (A_{kl} = 0)$ while parametrizing the strength of the likelihood of sharing a mode when an order dependency holds. Through a simple model [13, 33], the likelihood of two jobs $J_k$ and $J_l$ for arriving at the same mode $p_{kl}$ can be parametrized as follows: when there is an order relationship in the DAG, $p_{kl} = c$, where $c \in [0, 1]$ is the task clustering parameter, and when that relationship is missing the mapping is simply uniformly random $p_{kl} = 1/n_S$. This way, $c$ controls the likelihood of two adjacent jobs for being mapped to the same mode (i.e., zeroing communication costs). This process is termed as job clustering [17].

As no queuing delays and homogeneous processing speeds are considered, processing times are also constant and independent of the mapping. Hence, job processing times $PT = 48.32Kbytes/s$ and a maximum latency $\text{LAT} < 10^{-1}s$. This can be achieved either with small file sizes or with a high bandwidth network (as it is usual in dedicated research networks and many computing Grids). In the experimental study of the ES-NGI Grid described in Section [11] Round Trip Times (RTT) for every IP packet were also collected. We found a minimum $BW = 48.32K$ bytes/s and a maximum latency $\text{LAT} < 10^{-1}s$. Hence, the latency-bounded approximation is justified for file sizes of the order of bytes.

Another simplification is achieved if equally sized clusters are considered. Then, communication costs between jobs $J_k$ and $J_l$ can be computed as

$$CT_{kl} = [1 - \Delta_{kl}][\text{LAT} \cdot D_{kl}^C],$$

where $D_{kl}^C = D_{kl}^b + D_{kl}^b + D_{kl}^c$ is the inter-CE joint distance (see Fig. 1) composed of: 1) Intra-mode CE-broker.
distance $D_{kb}^c$ at the departure mode $x_k$. 2) Inter-mode 
broker-broker distance $D_{kl}^c$ from modes $x_k$ to $x_l$, 3) Intra-
mode broker-CE distance $D_{kl}^c$ at the arriving mode $x_l$. It must 
be highlighted that this linear relationship (also used in [10]) between 
communication times and hosts distance is only a rough approxima-
tion to the real internet communication processes; as communications 
depend heavily on network congestion (i.e., traffic) the assumption 
of a constant latency may not be adequate in general. On the 
other hand, a strong correlation has been found 
between hop-count and packet transfer times [32] suggesting 
that packet transfer times may increase linearly 
with the distance. This evidence empirically supports the 
linear relation (3.8) that can be thought as a limit case when $O_{kl}/BW \to 0$ and no background traffic effect 
is included.

Under these assumptions we now define the set of met-
rics for the quantification of applications performance 
used in this work. From the adjacency matrix $A$, the 
finish time $FT_k$ of job $J_k$ is given by

$$FT_k = PT + \max\{A_{ik}(FT_i + CT_{ik})\}_{i=0,\ldots,n_j}. \quad (3.4)$$

As for the entry node it holds that $FT_0 = 0$, and 
by noticing that for the proposed DAG $A$ is upper-
triangular, a perfect recurrence relation is found: $FT_k = 
PT + \max\{A_{ik}(FT_i + CT_{ik})\}_{i<k}$, $k = 1, \ldots, n_j$.

The first performance metric considered is the total ex-
ecution time of a parallel application makespan $(or MK$ 
short) defined as $MK = \max\{FT_k\}_{k=1,\ldots,n_j}$ (in seconds).

Another common measure is the Scheduled Length Ratio $SLR = MK/CPIC$, which normalises makespan 
with the Critical Path Length Including Communication (CPIC) [17, 31]. CPIC is computed in the same way 
as $MK$ but in this case tasks are mapped to an aver-
age mode $S$ where both intra-mode and inter-mode dis-
tances are equal to the average shortest path length $l$. 
According to this, the CPIC processing and communi-
tication times are defined as: $PT_{k}^{CPIC} = M/V$ and 
$CT_{k}^{CPIC} = [1 - \Delta_{k}]/(\Phi \cdot L \cdot LAT)$ respectively. Then, CPIC 
is obtained as $CPIC = \max\{FT_k\}_{k=1,\ldots,n_j}$ by plugging 
$PT_{k}^{CPIC}$ and $CT_{k}^{CPIC}$ into (3.4). A value of $SLR = 1$ 
means that the allocation achieves a performance equal 
to that obtained in the average network. Values lower 
than 1 indicate a better mapping whereas values higher 
than 1 render a worse strategy. In general, the lower $SLR$ 
the better performance obtained. In a uniform mapping 
$SLR$ has a lower-bound of 1, but as it will be shown, 
smarter mappings (i.e., taking into account the services 
degree or betweenness) can achieve values with $SLR < 1$.

Other common metric is the parallel efficiency PE or 
normalised speedup. As defined in Ref. [10] PE is the 
normalised ratio between sequential and parallel time for 
an application to execute $PE = \sum_{i} PT_{i}/(n_S MK) = 
n_P PT/(n_S MK)$.

D. Probabilistic modelling and order statistics

The proposed model has three main elements: network, 
computing services mappings and job allocation process $AP$. 
Starting from the sample space $\Omega = G_R \times R^{R^C} \times 
R^{nc}$, each possible configurations $x \in \Omega$ with $x = 
(g, \beta, \theta, \hat{X})$ represents; a graph instance $g$, a brokers $\beta = 
(\beta_1, \beta_2, \ldots, \beta_{nc})$ and CEs mapping $\theta = (\theta_1, \theta_2, 
\ldots, \theta_{nc})$ and an allocation vector: $\hat{X} = (X_1, X_2, \ldots, X_{nc})$. 
This way, any time function $f$ is regarded as a simple ran-
dom variable with expectation $E\{f\} = \sum_{x \in \Omega} f(x)P(x)$. 
Graph instances are referred to any of the two classical 
random graph models $G_R \in \{G_{n_R,m}, G_{n_R,p}\}$ as defined 
in [33] while $E[]$ and $P()$ denote expectation and probability 
in the $G_{n_R,m}$ and $G_{n_R,p}$ models interchangeably.

Through a simple re-ordering of job indexes and by 
assuming equally sized job clusters $\omega_k - \omega_{k+1} = \omega, \forall k$ in 
the DAG shown in Fig. 2 it is found that

$$MK = \omega PT + \max\{\sum_{i=\omega(k-1)+1}^{\omega k-1} CT_{it+1}\}_{k=1,\ldots,n_j} \quad (3.5)$$

(an equivalent expression is obtained for CPIC) that can 
be rearranged as

$$MK = \omega PT + LAT \cdot Y \quad (3.6)$$

$$SLR = \frac{\omega PT + LAT \cdot Y}{\omega PT + LAT \cdot \Phi \cdot L \cdot Z}, \quad (3.7)$$

where the quantities $Y = \max\{U_k : k = 1 \ldots c_J\}$ and 
$Z = \max\{W_k : k = 1 \ldots c_J\}$ are defined through the 
relations

$$U_k = \sum_{i=\omega(k-1)+1}^{\omega k-1} [1 - \Delta_{ii+1}]D_{ii+1}^c \quad (3.8)$$

$$W_k = \sum_{i=\omega(k-1)+1}^{\omega k-1} [1 - \Delta_{ii+1}]. \quad (3.9)$$

Since the expressions for $Y$ and $Z$ involve the 
maximum function, obtaining their expected values is 
not straightforward and requires techniques from order 
statistics analysis. By series expansion up to second 
order of (3.7) around $E[MK]$ and $E[CPIC]$ (see Ref. [34]) 
and by neglecting terms $O^2(LAT)$, $E[SLR]$ can be approximated 
by

$$E[SLR] \approx \frac{\omega PT + LAT \cdot E[Y]}{\omega PT + \Phi \cdot E[\Phi] \cdot LAT \cdot E[Z]}, \quad (3.10)$$

where we have used that $l$ and $Z$ are independent variables 
since $l$ depends solely on topology while $Z$ depends 
on allocation. The quantities $E[Y]$ and $E[Z]$ can be 
computed by using the conditional expectation rule (the corre-
ponding partition in $\Omega$ is always possible): $E[Y] = 
\sum_{g \in G_R} P(g)E[Y|g]$ (same for $Z$) where $P(g) = P([g])$ 
and $E[Y|g]$ represent the conditional expectation of $Y$ 
for a given graph instance $g$. By assuming that both $U_k$
and \(W_k\) with \(k = 1, \ldots, c_j\) form Independent and Identically Distributed (IID) sets of random variables, the set \(\{U_1, \ldots, U_{c_j}\}\) (same for \(W\)) can be thought as a set of statistical samples from the simple random variables \(U : \{g\} \times R^{n_u} \times R^{nc} \times S^{\omega-1} \rightarrow \mathbb{N}\) and \(W : S^{\omega-1} \rightarrow \mathbb{N}\) with distributions \(f_U(x)\) and \(f_W(x)\) respectively. Hence, the first moments of \(Y\) and \(Z\) can be computed in the same order statistics of \(U = \sum_{i=1}^{\omega-1} (1 - \Delta_{ii+1})D_{\text{ii}}^{cc}\) and \(W = \sum_{i=1}^{\omega-1} (1 - \Delta_{ii+1})\) with supports \(\{0, 1, \ldots, S_{\omega}\}\) and \(\{0, 1, \ldots, W_{\omega}\}\) respectively. By using the results reported in (3.34) it is found that

\[
E[Y|g] = S_U - T_U(S_U, c_j) \quad (3.11)
\]

\[
E[Z|g] = S_W - T_W(S_W, c_j), \quad (3.12)
\]

where the quantities: \(T_U(S_U, c_j) = \sum_{x=0}^{S_U-1} [f_U(x)]^{c_j}\) and \(T_W(S_W, c_j) = \sum_{x=0}^{S_W-1} [f_W(x)]^{c_j}\) have been introduced. By noticing that the value \(S_W\) is reached when \(\Delta_{ii+1} = 0, \forall i\), it follows that \(S_W = \omega - 1\). Further, as these quantities do not depend on the network

\[
E[Z] = E[Z|g] = (\omega - 1) - \sum_{x=0}^{\omega-2} [f_W(x)]^{c_j}. \quad (3.13)
\]

The quantity \(S_U\) can be computed by realising that the compound distances \(D_{\text{ii}}^{cc}\) can be rearranged as

\[
D_{\text{ii}}^{cc} = (1 - \delta_{1n_R})D_{\text{ii}}^{cb} + (1 - \delta_{P2P})(D_{\text{ii}}^{bb} + D_{\text{ii}}^{cc}).
\]

As the services can be mapped to any node in the network, the support of \(D_{\text{ii}}^{cb}\) and \(D_{\text{ii}}^{cc}\) is \(\{0, \ldots, D, \forall i\}\). This way, the support of \(D_{\text{ii}}^{cc}\) is \(\{0, 1, \ldots, \Phi D\}\), \(\forall i\). Hence, as task clustering and topology are considered as independent, it is obtained that \(S_U = \Phi DS_W\), which couples DCI solution \(\Phi\), network diameter \(D\) and task clustering \(W\). With these values (3.11) renders

\[
E[Y] = \Phi E[D](\omega - 1) - \sum_{g \in G_R} P(\omega)T_U(\Phi D(\omega - 1), c_j).
\]

A series expansion of \(T_U\) around \(E[S_U]\) in the sum of (3.14) leads to

\[
E[T_U(S_U, c_j)] \approx T_U(E[S_U]) + \frac{1}{2} Var(S_U) \partial^2 T_U(z, c_j) |_{z = E[S_U]}, \quad (3.15)
\]

where \(Var(S_U) = \Phi(\omega - 1)Var(D)\).

In our analytical model the diameter variance is neglected (reasons for this will be provided later). Moreover, as \(D\) can only take positive integer values, considering \(E[D]\) introduces round-off biases. A way to tackle this is by setting \(\tilde{D} = \text{round}(E[D])\) where the round function outputs the nearest natural number of \(E[D]\). Then from (3.14) and (3.15)

\[
E[Y] = \Phi \tilde{D}(\omega - 1) - \sum_{x=0}^{\omega-2} [f_U(x)]^{c_j}. \quad (3.16)
\]

and the expressions for the first moments of \(MK\) and \(CPIC\) result in

\[
E[MK] = \omega PT + \Phi \tilde{D}(\omega - 1) - \sum_{x=0}^{\omega-2} [f_U(x)]^{c_j}. \quad (3.17)
\]

and

\[
E[CPIC] = \omega PT + \Phi^2 \tilde{D} E[TL](\omega - 1) - \sum_{x=0}^{\omega-2} [f_W(x)]^{c_j}. \quad (3.18)
\]

These relations depend on DAG \((\omega, c_j)\), computing network \((f_U(x), l, \tilde{D}, \Phi)\) and allocation \((c, f_W(x))\) parameters. It can be noticed how \(E[MK]\) increases with processing time, cluster size, latency, DCI solution and network diameter while it decreases with clustering probability and with the number of task clusters. The summation term in (3.17) and (3.18) captures the effect of the distance distribution in the underlying topology. The minimum is achieved for configurations such that \(f_U(x) = 1\), (e.g., the complete graph limit) where \(E[MK] = \omega PT\). Conversely, if \(f_U \approx 0\), \(MK\) reaches its maximum value at \(E[MK] = \omega PT + \tilde{D} \cdot \Phi^2 \cdot \tilde{D}(\omega - 1)\).

### E. Limit cases

Despite the simplifications introduced so far, the handling of the expressions for \(E[MK]\) and \(E[SLR]\) from (3.10), (3.17) and (3.18) quickly becomes unmanageable. These expressions can be substantially simplified if some additional restrictions are imposed. In particular (see Appendix B for a more detailed description) we assume

1. **Job pairs limit.** In this case we only consider task clusters of size \(\omega = 2\).

2. **Random distribution of resources.** Both brokers and CEs are uniformly mapped over the network with probability \(1/n_R\). For P2P (\(\Phi = 1\)) configurations CEs and brokers in a computing mode are mapped to the same router.

Under these assumptions, two limit cases are found: 1) **Total task attraction** \(c = 1\), where the clustering probability reaches its maximum and all tasks are mapped to the same mode (a rather trivial case) and 2) **Task repulsion** \(c = 0\); dependent jobs tend to avoid being mapped to the same mode (a sort of load balancing solution) and tasks are mapped in a **RoundRobin** fashion. In the latter case, if a single cluster is also considered (i.e., \(c_j = 1\)) expressions for the metrics are greatly simplified. Finally, if the distance distribution is regarded as uniform and the number of task clusters is large, further simplifications in the metrics expressions are obtained. All these approximations are collected in Table I. Notice that under
In this section we numerically investigate the relationships between performance metrics and network topology, DAG structure, and allocation. A set of Monte Carlo simulations consisting on $N$ independent samples for each configuration $x \in \Omega$ were generated. Then, values for the performance metrics introduced in Sec. IV were computed. This was performed in a specifically designed software tool: Stochastic Grid Workbench (SGW).

Although a number of popular Grid and Cloud simulators exist (e.g., GridSim for Grids [36]), the implementation of customised complex topologies and parametrized DAG structures in these frameworks is not a trivial issue as these environments are mainly designed for general purpose configurations. This motivated the development of SGW, a JAVA based tool which uses the Java Universal Network/Graph Framework API [37] to numerically validate the expressions obtained. In SGW graphs can be: 1) Loaded (as the empirical ES-NGI obtained through the procedures described in section II) 2) Enriched from a seed graph by the random addition of new links, and 3) Generated through different graph generation algorithms provided in [37]. In the case of 1) the graph instance $g \in G_R$ in $x$ is constant and hence $E[l] = l$ and $E[D] = D$ where $D$ is the ES-NGI diameter $= 15$. In 2) once new links are added, the resulting graph remains constant in the ensemble during the tests and hence it holds again that $E[l] = l$ and $E[D] = D$. In 3), both $l(g)$ and $D(g)$ have in general a distribution whose moments depend on the graph generation mechanism used. From a battery of tests designed to gauge $l$ and $D$ variances we verified that for the random graph, according to theory, $\text{Var}(l) \to 0$, once the phase transition in connectivity at $p_{nR} = \log(n_R)$ is reached [38].

TABLE I. Performance metrics where simple relations are found for three limit cases of the parameters involved. The dimensionless parameter $\eta = LAT/2PT$ is also introduced.

| Normalised metrics | $c = 1$ | $c = 0$ |
|--------------------|----------|----------|
| $(E[MK] - 2PT)/LAT$ | $0$ | $\Phi E[l]$ |
| $E[SLR]$ | $1$ | $\Phi D$ |
| $E[PE]$ | $\frac{c_J/n_S}{1+\Phi E[l]}$ | $\frac{c_J/n_S}{1+\Phi E[l]}$ |

* Considering uniform topologies limit

Also, equivalent graphs with the same size and approximately the same edges than ES-NGI were obtained for both the random (Erdős-Rényi model) and for the scale-free versions. These graphs are termed as ER-NGI and SF-NGI respectively. ER-NGI were built by linking $n_R = 189$ nodes with probability $p = 2n_R/n_R(n_R - 1) = 0.02685$. The ES-NGI values of $n_B = 5$ and $n_C = 23$ were also used for computing services mapping. For SF-NGI networks we slightly modified the preferential attachment model [39] by starting from $v_0 = 30$ initial vertices and by adding $e_0 = 3$ new edges per iteration through approximately $n_R/e_0 = n_R - v_0$ iterations.

Once the network is obtained, non-connected solutions are filtered in order to keep the largest connected component only. Then, after setting the values of latency $LAT$ and bandwidth $BW$, the desired DCI solution is implemented as a mapping of brokers and CEs on that network. Random, degree or betweenness based mapping algorithms are possible. Besides, proximity based strategies, where dependent jobs are preferentially mapped to closer modes, are also implemented. Application DAGs are specified through the graphML format, where DAG structure and both node and link weights are provided. Finally, the probabilistic task allocation addressed in Sec. III is implemented resulting into an allocation vector $X$. Once all tasks are mapped to modes, the performance metrics $MK$, $CPIC$, $SLR$ and $PE$ for every configuration point $x \in \Omega$ are obtained for their statistical analysis.

IV. MODEL VALIDATION AND MAIN RESULTS

As stressed, a major factor for the performance in DCIs is the greater or lesser degree of hierarchization in their services overlay network.

In a first experiment (Fig. 3) we monitored the average $MK/(LAT \cdot l)$ from a set of $N = 1000$ samples for different network configurations and $n_C = 50$ CEs as the amount of brokers was ranged from 1 to 50 in ER-NGI. To highlight communication overheads, the allocation strategy was task repulsion ($c = 0$) and job pairs $(\omega = 2$ and $c_J = 1)$ with zero processing time $PT = 0$. The analytic limit for this case is $E[MK]/(LAT \cdot E[l]) = \Phi$ (case II in table I). In the $\Phi = 2$ case only a point at $E[MK]/(LAT \cdot E[l]) = \Phi = 2$ coinciding with the value for the $\Phi = 3$ solution for $n_B = 1$ is obtained (in this case, the P2P solutions render $E[MK]/(LAT \cdot E[l]) = 0.0$, as communication overheads are zeroed). Hence, from a strictly topological perspective, it is concluded that the centralised configuration $\Phi = 2$ outperforms the less hierarchical $\Phi = 3$ scheme if services are mapped randomly.

When a proximity optimisation is enhanced, the inter-node broker-broker distance is minimised and the case $\Phi = 2$ is recovered. It must be stated, however, that in real DCIs, finite queuing effects at brokers can not be neglected (in fact brokers downfall is a common cause for
In Fig. 4 we compare the average SLR is ranged from the critical probability \( p \) is clustered around finite sets of values for the range \( D \) is almost surely concentrated on at most four values \( z = 3, 2, 1, 0 \) for different graph sizes \( n_R \). From that study we found a linear relationship between the jumps at \( z' = p' n_R / \log(n_R) \) and \( p \) in a P2P solution with \( n_B = n_C = 50 \) and \( c = 0 \).

From Fig. 4 it is noticed that the model with Fronczak et al. distribution starts to reproduce numerical experiments for \( z > 2 \), where the diameter is concentrated on at most 3 values \( z = 2, 1, 0 \). In this case the variance of the diameter is negligible and the approximation of \( f_\text{Fronczak}(x) \) becomes reliable. Notice that a wide range of convergence between tests and model is found. However, this agreement is weakened as we approach the complete graph transition (the diameter abruptly changes from 2 to 1).

A detailed analysis about how the diameter transitions affect the performance is out of the scope of the present work. However, in a set of tests we monitored the diameter transitions from values 3 to 2 for different graph sizes \( n_R \). From that study we found a linear relationship between the jumps at \( z' = p' n_R / \log(n_R) \) and \( p \) in a P2P solution with \( n_B = n_C = 50 \) and \( c = 0 \).
point $z$ in Fig. 4 where the uniform distribution model starts to reproduce experiments.

It should be noted the local minimum found at $z = 10$. In this case a random mapping of tasks (neglecting topological and smart service mapping strategies) would render an optimal performance. If the ES-NGI topology was random, this would involve a connection probability of $p = 0.277$ (a factor of 10.66 times higher than the connection probability of ER-NGI). From this value, $D$ keeps constant while $l$ smoothly decreases for a wide range or $z$ and, hence, SLR increases. This trend is inverted near the complete graph limit ($z > 65$) where $D$ falls sharply to 1. In this region the model is unable to reproduce the last unstable values of $D$ (peak point at $SLR \approx 2$).

In another set of tests we gauge the empirical ES-NGI makespan as new links are randomly added. We allocated job pairs with $PT = 0$ on a $\Phi = 1$ configuration ($n_B = n_C = 50$ randomly mapped modes) with $c = 0$. These experiments were also designed to validate the model in the case of an empirical distance distribution (L7). In Fig. 5 we compare $\langle MK \rangle / LAT$ obtained from numerical simulations ($N = 500$ samples) with $E[MK]/LAT = E[D] - \sum_{x=0}^{D-1} [f_d(x)]^{c_f}$ for $c_f = 20$ and $c_f = 1$. In this case $f_d(x)$ was estimated through the empirical cumulative distribution function from the ES-NGI graph. For $c_f = 1$ it can be noticed that the linear relationship $E[MK]/LAT = l$ (case II in table 1) is reproduced.

Finally, we explored how the parallel efficiency of the ES-NGI could be improved by allowing more efficient computing services mappings. For a P2P solution ($\Phi = 1$) with $c = 0$ the expression for $E[PE]$ (neglecting second order terms in the variance of $MK$) renders

$$E[PE] = \frac{c_f/n_S}{1 + \frac{2\eta}{\omega} [D(\omega - 1) - \sum_{x=0}^{D-1} [f_D^{ \omega x}(x)]^{c_f}]}. \quad (4.2)$$

We launched $c_f = 20$ job pairs ($\omega = 2$) in a RoundRobin ($c = 0$) fashion with $\eta = 1/2$. The expression for $E[PE]$ from table 1 is a lower bound of (4.2) (solid line in Fig. 6). In this regard, the efficiency of a topology in a P2P network is driven by the quantity: $\sum_{x=0}^{D-1} [f_D^{ \omega x}(x)]^{c_f}$. The higher this value, the more efficient P2P infrastructures are obtained. In Fig. 6 we show $\langle PE \rangle$ in the ES-NGI and its random, and scale-free equivalents when the average shortest path $l$ is decreased by adding new links. For these tests Monte Carlo simulations from samples with $N = 500$ size were conducted for different computing services mappings and task allocation algorithms. As it was found in (10), the parallel efficiency increases by adding new links randomly (i.e., decreasing $l$) for both random and scale-free graphs. This is also confirmed in our tests for both the empirical ES-NGI Grid and for its synthetic counterparts.

We also monitored the efficiency for smarter mappings of computing services. In particular it was found that
when brokers were preferentially mapped to nodes with higher betweenness connectivity ($\beta = \text{bet. in Fig. 4}$), $\langle PE \rangle$ increased significantly with respect to the random mapping version. Also, by providing a distance optimisation mechanism (tasks are first allocated to closer modes) these results could be further improved. The abscissas (dotted lines in Fig. 3) correspond to the limit case of $E[PE]$ in Table II for uniform distributions $E[PE] = (c_J/n_S)/(1 + \eta D)$ at values $D = 1, 2, 3$. It can be noticed how the betweenness based mappings cluster around these limits. In this situation the contribution of the graph distance distribution to the efficiency is not significant. Finally, the maximum efficiency is clearly achieved for the complete graph when $D = 1$.

### B. Effect of the application workflow structure and allocation strategy

We shall now examine how the application workflow parameters ($c_J$, $\omega$) and the job clustering $c$ affect the performance metrics. Firstly, we monitor (Fig. 7) the normalised parallel efficiency $E[PE]n_S/c_J^{\max}$ in the ES-NGI topology as the normalised number of job clusters $c_J/c_J^{\max}$ increases. A P2P scheme with $n_S = 150$ randomly mapped modes and $c = 0$ was used. As expected from (4.2) (an equivalent result was reported in [10]) the parallel efficiency grows with the number of task clusters. An upper bound is found of $E[PE]n_S/c_J^{\max} = c_J/c_J^{\max}$ (diagonal solid line in Fig. 7). According to (4.2) this limit is reached when $\eta \to 0$ or when $\omega >> 1$ for a finite $\eta$ value. Conversely, if $c_J$ is small there will be not enough computational work to distribute in the computing modes resulting in small efficiencies. Also in da Fontoura et al. [10] it is claimed that efficiency increases with the task execution time. In our case this has been also observed when $\eta$ decreases; larger values for $PE$ are obtained for big processing times and low latency networks (see Fig. 7).

In the last set of experiments we investigate how $SLR$ depends on the task clustering parameter $c$. For this case from (B.1) and (B.6) it is found

$$E[SLR] = \frac{1 + \eta[\bar{D} - \sum_{n=0}^{c_J-1} c_{J,n} n \sum_{x=0}^{D-1} (f_d(x))^n]}{1 + \eta E[l](1 - c^c)}$$

with the limiting case of $SLR = 1$ for $c_J = 1$. In Fig. 8 we monitor the numerical average of $SLR$ ($N = 200$ samples) for the ES-NGI graph in a P2P solution of $n_S = 50$ modes deployed through a betweeness optimisation scheme by allocating job pairs with different clusterings and $\eta = 0.5$. On the left side ($c = 0$) de-

![FIG. 7. Normalised parallel efficiency vs. task cluster number. Solid lines represent the model from (4.2) for $\omega = 2$. Symbols represent the results from numerical tests and dotted lines represent numerical solutions for $\omega = 5, 30, 600$ with $\eta = 0.5$. The upper limit in efficiency (diagonal solid line) corresponds to $E[PE]n_S/c_J^{\max} = c_J/c_J^{\max}$.

![FIG. 8. Average $SLR$ for different cluster numbers as the task clustering parameter increases. The monitored DCI is the ES-NGI topology in a P2P solution when services are mapped through a betweenness based optimisation scheme.

dependent jobs are allocated in a RoundRobin fashion and communication overheads are maximum. As $c$ increases tasks tend to be allocated in the same computing modes and those communication costs decrease. In the right side for $c = 1$, all tasks are mapped to the same computing mode which, in average, has a distance of $D^{ee} \approx l$.
and, hence $E[SLR] = 1$. Interestingly, lower values for $E[SLR]$ than the minimum achievable in a random services mapping (solid line corresponding to $E[SLR] = 1$ in Fig. 6) can be obtained for different $c$ and $c_J$ combinations. Furthermore, from $c > 0.6$, the strategy for $c_J = 5$ outperforms the solution for $c_J = 1$; the allocation strategy scales with the number of tasks and the higher $c_J$ the better task allocation is achieved with respect to the random mapping. This evidences that when more accurate services mappings in a distributed computing infrastructure are addressed, optimised strategies can be enhanced by coupling task allocation and topology. However, such analysis is out of the scope of the present contribution and is left for future research.

V. CONCLUSION AND FINAL REMARKS

This work has been aimed at finding analytical expressions for the expected values of distributed computing performance metrics when the communication network has a complex structure. Through active traceroute-like probing tests a seed network was built from a real computing Grid. From this graph, ensembles of synthetic graphs with additional structure were generated to verify the model and to monitor different scenarios through numerical simulation. Although inspired by the Grid case, the models and results reported can be easily extended without loss of generality to other DCI solutions. Finally, to the authors’ knowledge, this is the first work that provides analytical expressions for the expected performance metrics when job dependencies and a complex network structure are considered. The main results obtained are summarised below.

1. Centralised schemes in distributed computing infrastructures in general increase parallel applications execution time due to the extra communication overheads between computing services. The optimal solution corresponds to a P2P model ($\Phi = 1$). On the other hand, if no additional optimisation strategies in the services mappings or task allocations are enhanced, common Grid infrastructures ($\Phi = 3$) do not improve performance when the brokers/CEs proportion increases. Moreover, the totally centralised ($\Phi = 2$) solution outperforms the ES-NGI Grid, although this improvement can also be achieved when tasks are preferentially mapped to closer resources.

2. By addressing more intelligent computing services mappings (in particular, a preferential mapping based on betweenness centrality or if tasks are more likely to reach closer modes), significant improvements in the parallel efficiency and scheduled length ratio are achieved. For random and scale-free equivalent graphs these improvements are less appreciable.

3. When the seed topologies are incrementally modified by the addition of new edges, the analytical expressions for the expected values of $MK$ and $SLR$ are numerically reproduced in a wide range of graphs. For the random graph, a local optimum is found in $SLR$ meaning that if distributed computing topologies hold a random-like communication network structure, targeted reconnections could improve the infrastructure performance significantly.

4. Once the ES-NGI topology has been enriched with a mode betweenness based mapping, an optimisation in the parallel efficiency can be achievable by coupling distance distribution $f_d(x)$ and task clustering parameter $c$. In this case, solutions with higher number of task clusters outperforms the performance of a single job pair ($c_J = 1$).

5. The parallel efficiency increases for denser graphs and for larger task clusters and processing times. This confirms qualitatively the results from [10].

Finally, we make some remarks on the scope and assumptions of the addressed model. Firstly, a Virtual node based communications network (where every node does not necessarily correspond to a communication device) with symmetrical paths and minimal distance based routing was considered. Further, the communication network was regarded as homogeneous in both latency and bandwidth. Background traffic effects were also neglected.

The computing elements were considered as uniform in processing speed and with no additional queuing delays. It was also assumed that brokers did not introduce any delay when mapping tasks. This constitutes a threat for the reliability of the model since, on the one hand, the local queue effect is a major factor in the performance of DCIs (and Grids in particular) [8] and, on the other hand, a notable characteristic of global DCIs is their heterogeneity in resources [8]. In a future we will investigate the coupling of those effects in the addressed model.

Moreover, no additional ordering is imposed to tasks when they reach a CE: they are processed in parallel if no precedence relationship exists between them. This slightly deviates our model from the traditional scheduling approach where both allocation and order should be specified.

We considered a minimalist DAG model with equally sized jobs and transfer files arranged into task clusters with the same dimension. Further, job sizes were regarded as small compared with the network bandwidth (i.e., latency-bounded approximation).

While these assumptions may turn the addressed model less reliable, it is our belief that including analytical expressions that link topology, application structure and allocation strategy parameters may render the distributing computing problem more manageable, easing the incorporation of additional effects. Furthermore, the techniques addressed in this work (e.g., the application
of order statistics analysis to DCI scheduling) have been validated numerically and are amenable to be used with real data.

ACKNOWLEDGMENTS

CETA-CIEMAT acknowledges the support received from the European Regional Development Fund through its Operational Program Knowledge-based Economy. A. A. acknowledges support from the Ministerio de Ciencia e Innovación (Spain) through Grant No. FIS2010-16587 and from the Junta de Extremadura (Spain) through Grant No. GR10158, partially financed by FEDER (Fondo Europeo de Desarrollo Regional) funds.

Appendix A: Probabilistic job allocation model

From the conditions addressed in Sec. III the following simple model is proposed

\[ p_{kn|lm} = A_{kl} \Gamma_{kl} + (1 - A_{kl}) p_{lm}. \]  
(A.1)

Here \( \Gamma_{kl} \) is a function of \( n,m \) to be determined. By imposing normalisation conditions it holds: 1) \( \Gamma_{kl} \in [0,1] \) and 2) \( \sum_m \Gamma_{kl} = 1 \). We also require that \( \Gamma_{kl} = c \in [0,1] \) if \( m = n \) being \( c \in [0,1] \) defined as the job clustering parameter. By using the former conditions over \( \Gamma_{kl} \)

\[ p_{kn,lm} = \frac{A_{kl} \cdot p_{kn}}{n_S - 1} \left[ \delta_{nm}(c \cdot n_S - 1) + 1 - c \right] + (1 - A_{kl}) p_{kn} \cdot p_{lm}. \]  
(A.2)

We also define the clustering probability \( \tilde{p}_{kl} = P[\Delta_{kl} = 1] = \sum_n p_{kn,lm} \) as the probability of jobs \( J_k \) and \( J_l \) are mapped into the same mode. By assuming also \( S \) as equiprobable \( \left( p_{kn} = 1/n_S \forall k \right) \) \( \left( A.2 \right) \) renders

\[ \tilde{p}_{kl} = c \cdot A_{kl} + \frac{(1 - A_{kl})}{n_S}. \]  
(A.3)

Appendix B: Derivation of analytical expressions for limit cases

1. job-pairs limit. In this case, a single clustering variable exists \( W = 1 - \Delta \) and \( f_W(0) = P[1 - \Delta \leq 0] = P[\Delta = 1] = c \). Then, (3.18) reduces to

\[ E[CPIC] = 2PT + \Phi \cdot LAT \cdot E[\lfloor 1 - e^{c \cdot j} \rfloor]. \]  
(B.1)

Further \( U = (1 - \Delta) D^{cc} \), and hence

\[ E[MK] = 2PT + \]  
(B.2)

\[ LAT[\Phi \tilde{D} - \sum_{n=0}^{e \cdot j} (f_{(1-\Delta)D^{cc}}(x))^{c \cdot j}]. \]  
(B.1)

It is noticed that for the computation of \( f_{(1-\Delta)D^{cc}}(x) \) a product form for \( S^{c \cdot j} = S^{e \cdot j} = (S^{2})^{c \cdot j} \) can be used through the partition: \( S^{2} = \{(X_{i}, X_{i+1})\} \), where the sample space is split into overlapping \( \Delta = 1 \) \( \Rightarrow \{(X_{i}, X_{i+1}) \in S^{2} : X_{i} = X_{i-1} \} \) and non-overlapping events. Since it also holds that \( \Delta = 1 \subseteq D^{cc} \leq x \), the event \( (1 - \Delta)D^{cc} \leq x \) can be rewritten as \( \Delta = 1 \cup (\Delta = 0) \cap D^{cc} \leq x^{n} \). As a consequence, distribution functions can be expressed as

\[ f_{(1-\Delta)D^{cc}}(x) = f_{D^{cc}}(x) + c\tilde{f}_{D^{cc}}(x). \]  
(B.3)

2. Random mapping of computing services. In this case \( D^{xy}, xy \in \{cb, bb, bc\} \) are IID simple random variables with distribution \( f_{d}(x) \). Then \( f_{D^{xy}}(x), xy \in \{cb, bb, bc\} \) can be thought as the empirical distribution functions from samples of size \( n_B \) and \( n_C \). If \( n_B \) and \( n_C \) are large enough compared with \( n_R \), by Glivenko-Cantelli theorem \( \Phi(\tilde{D}) \) it holds that \( f_{D^{xy}}(x) \) converges to \( f_{d}(x) \) \( \Rightarrow \{cb, bb, bc\} \). Furthermore, as in this case brokers and CEs can be interchanged, by symmetry we set that: \( D^{bc} \approx D^{bb} \approx D^{cc} \), and, hence, \( D^{cc} \approx \Phi(D^{xy}), xy \in \{cb, bb, bc\} \). As a consequence, \( \forall xy \in \{cb, bb, bc\} \)

\[ f_{D^{cc}}(x) \approx f_{D^{xy}}(x/\Phi) \rightarrow f_{d}(x/\Phi) = f_{\Phi d}(x). \]  
(B.4)

In other words, the random variable \( U \) can be thought as a rescaling of \( d \) in a factor of \( \Phi \). In this case \( \Phi(\tilde{D}) \begin{cases} \Phi(\tilde{D}) \end{cases} \) by using \( \Phi(\tilde{D}) \) and \( \Phi(\tilde{D}) \) renders

\[ E[MK] = 2PT + \]  
(B.5)

\[ LAT[\Phi \tilde{D} - \sum_{x=0}^{e \cdot j} \Phi_{\Phi^{-1}}(f_{\Phi d}(x))]^{c \cdot j}. \]  
(B.1)

As \( f_{\Phi d}(x) + \tilde{f}_{\Phi d}(x) = 1 \), it is noticed that

\[ E[MK] = 2PT + \]  
(B.6)

\[ LAT[\Phi \tilde{D} - \sum_{n=0}^{e \cdot j} P_{1-c}(c_j, n) T_{\Phi d}(\Phi \tilde{D}, n)], \]  
(B.6)

where through the binomial theorem \( P_{1-c}(c_j, n) = \binom{c_j}{n} (1 - c)^{n} e^{c_j - n} \) is the success probability of obtaining \( n \) successes in a Bernoulli trial with probability \( 1 - c \) and \( T_{\Phi d}(\Phi \tilde{D}, n) = \sum_{x=0}^{\Phi_{\Phi^{-1}}(f_{\Phi d}(x))^{n}}. \) It is noticed that \( P_{0}(c_j, n) = \delta_{n,0} \) and \( P_{1}(c_j, n) = \delta_{n,c_j}. \) Further, from \( \Phi(\tilde{D}) \) and \( \Phi(\tilde{D}) \) it can be found that \( T_{\Phi d}(\Phi \tilde{D}, 0) = \Phi \tilde{D} \) and \( T_{\Phi d}(\Phi \tilde{D}, 1) = \Phi(\tilde{D} - E[\lfloor 1 - e^{c \cdot j} \rfloor]) \).
Two limit cases are investigated:

- \( c = 1 \) (total task clustering), where \( E[CPIC] = E[MK] = 2PT \)
- \( c = 0 \) (total task repulsion). In this case \( E[CPIC] = 2PT + LAT \cdot \Phi \cdot E[l] \) and
  \[ E[MK] = 2PT + LAT(\Phi \tilde{D} - T_{\Phi d}(\Phi \tilde{D}, c_J)). \quad (B.7) \]

Two sub-cases are considered:

1. \( c_J = 1 \). Now a linear relationship between \( E[MK] \) and \( E[l] \) is found and
  \[ E[MK] = 2PT + \Phi \cdot LAT \cdot E[l]. \quad (B.8) \]

2. \( c_J >> 1 \) and uniform distance distribution. If \( f_{\Phi d}(x) \) is also regarded as uniform, from \( (B.4) \)
  \[ f_{\Phi d}(x) = x/(\Phi \tilde{D}) \] and it is obtained
  \[ T_{\Phi d}(\Phi \tilde{D}, c_J) = \frac{1}{(\Phi \tilde{D})^{c_J}} \sum_{x=0}^{\Phi \tilde{D}-1} x^{c_J}. \quad (B.9) \]

In the limit \( c_J >> 1 \) this term is neglected in \( (B.7) \) and the expression for \( E[MK] \) renders
\[ E[MK] = 2PT + LAT \cdot \Phi \tilde{D}. \quad (B.10) \]

---

[1] A. Barabási, Science 325, 412 (2009).
[2] B. Kahanwal and T. P. Singh, International Journal of Latest Research in Science and Technology 1, 183 (2012).
[3] I. Foster, C. Kesselman, and S. Tuecke, International Journal of High Performance Computing Applications 15, 200 (2001).
[4] K. Stanoevska-Slabeva, T. Wozniak, and S. Ristol, Grid and Cloud Computing (Springer, Dordrecht, 2010).
[5] A. Barabási, V. Freeh, H. Jeong, and J. Brockman, Nature 412, 894 (2001).
[6] R. Buyya and J. Brogerr, Cloud computing: principles and paradigms, Wiley series on parallel and distributed computing (Wiley-Blackwell, 2011).
[7] T. Karagiannis, Tech. Rep., UC Riverside (2003).
[8] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja, The advanced theory of statistics, 1 Ave, Cambridge (1992).
[9] M. G. Kendall, First Course in Order Statistics. (C. Griffin & company limited, 1945).
[10] V. Freeh, H. Jeong, and J. Brockman, Nature 412, 894 (2001).
[11] T. Karagiannis, Tech. Rep., UC Riverside (2003).
[12] L. Da Fonseca, F. Granelli, and D. Kliazovich, in 2007 IEEE International Conference on Communications (IEEE, 2007), pp. 344–349.
[13] B. Jones, in Peer-to-Peer, Grid, and Service-Orientiation in Digital Library Architectures, edited by C. Türker, M. Agosti, and H.-J. Schek (Springer Berlin Heidelberg, 2005), vol. 3664 of Lecture Notes in Computer Science, pp. 1–8.
[14] C. L. McCreary, A. A. Khan, J. J. Thompson, and M. E. McArdle, in Proceedings of 8th International Parallel Processing Symposium (IEEE, 1994), pp. 446–451.
[15] J.-J. Pansiot and D. Grad, ACM SIGCOMM Computer Communication Review 28, 41 (1998).
[16] L. Amini, A. Shaikh, and H. Schulzrinne, Computer Communications 27, 557 (2004), ISSN 0140-3664.
[17] W. Willinger, D. Alderson, and J. C. Doyle, Notices of the AMS 56, 586 (2009).
[18] A. Clauset, C. R. Shalizi, and M. E. Newman, SIAM review 51, 661 (2009).
[19] A. Clauset, http://tuvalu.santafe.edu/aarone/powerlaws/.
[20] chocolate chunk cookies, 282–289 (2004).
[21] Y. Derbal, Journal of Grid Computing 4, 373 (2006).
[22] R. Buyya, J. Brogerr, and I. Foster, in ITPPS ‘01 Revised Papers from the First International Workshop on Peer-to-Peer Systems (Springer-Verlag, 2002), pp. 232–241.
[23] M. Faloutsos, P. Faloutsos, and C. Faloutsos, in SIGCOMM ’99 Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication (ACM, 1999), pp. 251–262.
[24] K. Fujii and S. Goto, APAN/IWS2000, February (2000).
[25] H. E. Rewini, T. G. Lewis, and H. H. Ali, J. Electron. Comput. Syst. 894 (2001).
[26] M. Faloutsos, P. Faloutsos, and C. Faloutsos, in SIGCOMM ’99 Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication (ACM, 1999), pp. 251–262.
[27] Q. Chen, Q. C. Hynseok, R. Govindan, S. Jamin, and W. Willinger, in Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE, 2002), vol. 2, pp. 608–617.
[28] M. Faloutsos, P. Faloutsos, and C. Faloutsos, IEEE/ACM Transactions on networking 11, 514 (2003).
[29] J.-J. Pansiot and D. Grad, ACM SIGCOMM Computer Communication Review 28, 41 (1998).
[30] L. Amini, A. Shaikh, and H. Schulzrinne, Computer Communications 27, 557 (2004), ISSN 0140-3664.
[31] W. Willinger, D. Alderson, and J. C. Doyle, Notices of the AMS 56, 586 (2009).
[32] A. Clauset, C. R. Shalizi, and M. E. Newman, SIAM review 51, 661 (2009).
[33] A. Clauset, http://tuvalu.santafe.edu/aarone/powerlaws/.
[34] H. E. Rewini, T. G. Lewis, and H. H. Ali, Task scheduling in parallel and distributed systems (Prentice-Hall, 1994).
[35] L. Peterson and B. Davie, Computer networks: a systems approach, The Morgan Kauffman Series in Networking (Morgan Kauffman Publishers, 2007).
[36] Y. Zhang, Ph.D. thesis, Rice Univ. (2009).
[37] K. Fujii and S. Goto, APAN/IWS2000, February (2000).
[38] B. Bollobás, Modern graph theory (Springer, 1998).
[39] M. G. Kendall, The advanced theory of statistics, 1 Ave, Cambridge (1945).
[40] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja, A First Course in Order Statistics (Society for Industrial and Applied Mathematics, 2008).
[36] R. Buyya and M. Murshed, Concurrency and Computation: Practice and Experience 14, 1175 (2002).

[37] J. O’Madadhain, D. Fisher, S. White, and Y. Boey, Tech. Rep., UCI-ICS (2003), URL http://www.datalab.uci.edu/papers/JUNG_tech_report.html.

[38] F. Chung and L. Lu, Advances in Applied Mathematics 26, 257 (2001).

[39] A. Barabasi and R. Albert, Science 286, 509 (1999).

[40] B. Bollobás, Graph theory and combinatorics (Cambridge Academic Press, 1984), chap. The evolution of sparse graphs, pp. 35–57.

[41] A. Fronczak, P. Fronczak, and J. A. Holyst, Physical Review E 70, 056110 (2004).

[42] V. D. Blondel, J.-L. Guillaume, J. M. Hendrickx, and R. M. Jungers, Physical Review E 76, 066101 (2007).