Monte Carlo Approach to the Computational Capacities Analysis of the Computing Continuum

Vladislav Kashansky[1,2][0000−0001−9437−1790], Gleb Radchenko[2][0000−0002−7145−5630], and Radu Prodan[1][0000−0002−8247−5426]

[1] Institute of Information Technology, University of Klagenfurt, Austria
[2] School of Electronic Engineering and Computer Science, South Ural State University, Russia
vkashansky@acm.org, gleb.radchenko@susu.ru, radu.prodan@aau.at

Abstract. This article proposes an approach to the problem of computational capacities analysis of the computing continuum via theoretical framework of equilibrium phase-transitions and numerical simulations. We introduce the concept of phase transitions in computing continuum and show how this phenomena can be explored in the context of workflow makespan, which we treat as an order parameter. We simulate the behavior of the computational network in the equilibrium regime within the framework of the XY-model defined over complex agent network with Barabasi-Albert topology. More specifically, we define Hamiltonian over complex network topology and sample the resulting spin-orientation distribution with the Metropolis-Hastings technique. The key aspect of the paper is derivation of the bandwidth matrix, as the emergent effect of the "low-level" collective spin interaction. This allows us to study the first order approximation to the makespan of the "high-level" system-wide workflow model in the presence of data-flow anisotropy and phase transitions of the bandwidth matrix controlled by the means of "noise regime" parameter η. For this purpose, we have built a simulation engine in Python 3.6. Simulation results confirm existence of the phase transition, revealing complex transformations in the computational abilities of the agents. Notable feature is that bandwidth distribution undergoes a critical transition from single to multi-mode case. Our simulations generally open new perspectives for reproducible comparative performance analysis of the novel and classic scheduling algorithms.

Keywords: Complex Networks· Computing Continuum· Phase Transitions· Computational Model· MCMC· Metropolis-Hastings· XY-model· Equilibrium Model

1 Introduction

Recent advancements [3, 30] in the field of parallel and distributed computing led to the definition of the computing continuum [6] as the environment comprising highly heterogeneous systems with dynamic spatio-temporal organizational structures, varying in-nature workloads [14], complex control hierarchies [8], governing computational clusters with multiple scales of the processing latencies, and diverse sets of the management policies [17, 31]. The emergence of these systems is the natural response to the ever-growing variability of computational demands.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

This paper investigates how to approach the problem of modeling the computing continuum, considering it as an active part of evolving multi-agent networks, similar to Internet, with complex emergent properties. This problem remains still mostly uncovered in the literature of the scheduling problems. Moreover, since emergence of the concept in 2020 [6], there is a lack of a *reproducible model* of the computing continuum, especially for better understanding scheduling heuristics, as real systems do not preserve this quality and hinder the comparative performance analysis of the novel scheduling approaches.

Our paper aims to fill these gaps through simulation, allowing to study the computing continuum with the preferential attachment topology [1] and given workload, based on a theoretical framework with some initial conditions. We implement our theoretical model as module of a simulator called *Modular Architecture for Complex Computing Systems Analysis (MACS)* [16]. We introduced a parameter related to the noise regime in the system that allows us to study different congested states of the computational network, considered as a graph cut from a global self-organizing network.

The contribution of this paper is as follows:

1. We analyse the definition [6] of the computing continuum and provide a theoretical model of how high-order computational properties emerge from elementary pairwise interactions;
2. We study the behavior of the *fully observable* agent computational network, simulated in the equilibrium regime within the framework of the modified XY-model [15] defined over complex network with Barabasi-Albert topology [1]. We define the Hamiltonian, which encodes dynamical properties of the system over a complex network topology, and sample the resulting Gibbs distribution of spin-orientations with the Metropolis-Hastings technique [15].
3. We derive the bandwidth matrix, as the emergent effect of the "low-level" collective spin interaction, which allows us to study the first order approximation to the behavior of the "high-level" system-wide workflow model in the presence of emergent data-flow anisotropy. We obtained the DAG network structure from the PSLIB project scheduling library [21], namely the state-of-the-art j60-60 benchmark instance consisting of 60 tasks.

The remainder of the paper is structured as follows. Section 2 discusses the related work. Section 2.1 describes the background definitions and introduces reader to our vision of the computing continuum concept, meanwhile Section 2.2 and 2.3 devoted to the specific structural aspects. Sections 3 describes proposed model. Section 4 presents an experimental evaluation and Section 5 concludes the paper.

2 Related Work and Definitions

2.1 Computing Continuum

To better understand the case of computing continuum, we should recall to complex multi-layer computational networks, that are operated by large corporations and governments. Examples of such networked systems are:

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

- social platforms that analyse concurrently various motion patterns and opinion dynamics [4, 5] related to human behaviour at various spatio-temporal scales;
- self-organizing vehicle fleets and drone swarms [9, 24] that receive information from a large group of spatial sensors and need to make decisions accordingly.

Such networks consist of a large number of locally interacting *agents* that form stable emergent flows of data and execute arbitrary workloads. It is important to highlight that we are no longer dealing with machines in the case of computing continuum, but with agents [4] capable of consensus formation of various kind.

A key property typical to all sufficiently complex ecosystems is their tendency to grasp the sub-systems, provide uniformity over heterogeneity and optimize the end-point *workload execution* in the presence of *target criteria with system-wide constraints*. Haken [12] introduced the concept of "order parameter", as lower-dimension projection, governing large-scale dynamics of the complex system as whole, emerged from local interactions. He noted the role of natural language of any kind to synchronize all the individuals, reduce/compress the complexity of mind and maintain the synchronous state across the society, similarly to the order parameter of the social network. The same logic can be seen in the computing systems with the introduction of formal grammars, programming languages and interfaces [13].

Computing continuum can be defined as a *convolution hierarchy* of interacting order parameters from inherently local, associated with tasks to global which are structurally prescribed by the workload network. In our work we interpret global order parameter of whole convolution hierarchy as Makespan.

Finally, we can characterize computing continuum as transfusion of the three components, namely:

- *Computational network*, which provides structural knowledge via statistics about the possible information flows within the system and possible interactions of agents via adjacency matrix;
- *Recursive network* which forms a multi-layer DAG and provides knowledge about non-equilibrium dynamic processes in the computational network. This component is optional and only required if behaviour of the network is considered far from equilibrium. Examples include: Dynamic Bayesian Networks (DBN) and Hidden Markov Models (HMM);
- *Workload network* is the set of tasks, represented as the DAG that governs computational process in the computing continuum, prescribing arrow of the time.

Computational and workload networks are inherently *complex networks*, described in terms of random graphs with corresponding statistical properties [5]. In the computational network, data transfer and local performance rates are emerging at various noise regimes. Tasks are interconnected by a precedence relation and have different data volumes to transmit between each other and computation time, as well as transmission rate, will depend on matching with computational agents. Together tasks form a global collective order parameter called *Makespan*, which does not exist when each task is considered individually. The order parameter for a computational network is the degree of consensus of the agents. As a result, we obtain a hierarchy, which at its highest level characterized by single parameter - Makespan, and at its lowest level is described by the local behavior of the computing agents.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

2.2 Computational Network with Scale-Free Topology

In our study, we decided to model computational network topology as a scale-free network, as many of the networks under study fall into the class. This means that they have a power-law distribution with respect to the degree of a node. Scale-free networks are important phenomena in natural networks and human networks (Internet, citation networks, social networks) [1]. Barabasi-Albert theoretical model of scale-free network incorporates two important general concepts: network growth mechanism and preferential attachment principle (PA). Both concepts are widely represented in real-world networks. Growth means that the number of nodes in a network increases over time. Preferential attachment principle prescribes, that the more links a node has, the more likely it is to create new links. Nodes with the highest degree have more opportunities to take over the links added to the network. Intuitively, the principle of preferential attachment can be understood if we think in terms of social networks that connect people together. Strongly connected nodes are represented by known people with a large number of connections. When a newcomer enters a community, it is more preferable to connect with one of the known people than with a relatively unknown person. Similarly, on the World Wide Web, pages link to hubs, for example, well known sites with high Page Rank (PR) [28], than to pages that are not well known. If you choose a new page to link to randomly, then the probability of choosing a particular page will be proportional to its degree. This explains the preferential attachment principle.

The PA principle is an example of positive feedback, where initially random variations in the node number of node links are automatically amplified, thereby greatly increasing the gap and allowing hub formation.

2.3 Modelling Congested States of the Computing Continuum through Phase Transitions

Natural phase transitions refer to the change between various states of matter, such as the solid, liquid, and gaseous states. The transition from one state to the other depends upon the value of a set of external parameters such as temperature, pressure, and density characterizing the thermodynamic state of the system. In the context of complex networks, various Monte Carlo simulations of Ising model on small-world networks confirmed the existence of a phase transition of mean-field character [5]. Specifically, the critical behavior of the XY-model in small-world networks was studied in [26].

The research on phase transitions of various kinds has a long tradition in physical sciences. Due to the lack of space, we refer to [5, 15, 18] for general and specific aspects of the problem. For further reading, it is only important to relate "order parameters" with small changes induced by the noise regime. A remarkable connection of the phase transitions and NP-hard problems [27] shows the important influence of the input data distribution on the sensitivity of optimization algorithms. Namely, it is interesting to identify how these small changes will result in dramatic changes in the convolution hierarchy of interacting order parameters due to non-linear effects, from *local* associated with tasks, to *global* measures such as Makespan.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

3 Mathematical Model

3.1 Workload Model

We define a *workload* as a directed acyclic graph (DAG) where V is the set of tasks and E is the set of edges between them: $G = (V, E)$. Each edge $(i, j) \in E$ represents a precedence constraint indicating that the task i must complete its execution before the task j starts. Preemption is not allowed.

3.2 Agent Model

We consider a set of independent agents A with different processing speeds and synchronized clocks. Computational properties of the agents are expressed with a non-symmetric positive matrix $\mathcal{B}$. Precisely speaking, off-diagonal entries model bandwidth between agents m and q and diagonal elements correspond to the agent self-performance. All entries in the matrix are normalized and dimensionless.

3.3 Network Model

We model the computational network as a random graph with Barabasi-Albert scale-free topology that provides structural knowledge about its topological properties: $N = (A, Q)$, where A is the set of the agents and $Q \subset A \times A$ is the set of interconnections $(m, q) \in Q$ between them.

For an analytical insight into simulation and phase transitions, we rely on the equilibrium statistical physics framework, which considers that the network N can be in any possible microscopic configuration. Considering XY-model [5, 15] defined over the network N we map the two-dimensional unit vector $\vec{S_m}$:

$$\vec{S}_m = (cos(\theta_m),\ sin(\theta_m)) \tag{1}$$

to each vertice in A with angle $\theta_m \in [0; 2\pi]$. Spin vector is inherently local property and has no significant impact in our model when considered separately. In the context of our problem, it accounts on how the noise η numerically impacts the ability of the network to perform computations.

Direct interpretation consists in considering the existence of a centralized policy-maker in the computing continuum. A spin vector θ_m then represents the dynamic scalar degree of agent's belief [5, pp. 216–241] to the "center", prescribing mechanism of the consensus formation. In this scenario, agents assumed to have a continuous opinion amplitude from 0 to 2π. However, this interpretation may not be important, for example when simulating a comparative study of the various scheduling heuristics. The information about the connection between agents is encoded in the adjacency matrix J of the graph N, considering local interaction of the adjacent agents. The Hamiltonian then associates an energy $\mathcal{H}$ to each configuration:

$$\mathcal{H}(\boldsymbol{\theta}) = \sum_{m \in A} \sum_{q \neq m} J_{mq} \cdot [1 - \cos(\theta_m - \theta_q)]. \tag{2}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

Further, the configuration probability is given by the Boltzmann distribution with noise regime $\eta \geq 0$:

$$P(\boldsymbol{\theta}) = Z^{-1} \cdot e^{-\frac{\mathcal{H}(\boldsymbol{\theta})}{\eta}}; \quad Z = \int_{[0,2\pi]^{\Lambda}} \prod_{m \in A} d\theta_m \, e^{-\frac{\mathcal{H}(\boldsymbol{\theta})}{\eta}}. \tag{3}$$

where Z is the normalization factor.It is important to highlight that we operate with a *fully observable* computational network.

3.4 Order Parameter of the Computational Network

The distribution $P(\boldsymbol{\theta})$ provides important knowledge about network behaviour and importantly, order formation at low noise regime limit. The first important characteristic is the average orientation $\hat{\theta}_m$ computed via the following integral:

$$\hat{\theta}_m = \int \theta_m \cdot f(\theta_m) \, d\theta_m. \tag{4}$$

Further, we require a two-point co-variation function R_{mq} given by the following two-dimensional integral [18]:

$$R_{mq} = \iint (\theta_m - \hat{\theta}_m) \cdot (\theta_q - \hat{\theta}_q) \cdot f(\theta_m \cdot \theta_q) \, d\theta_m \, d\theta_q. \tag{5}$$

Averaging across all spins in the network N leads to the synchronization degree $\mathcal{M} \in [0; 1]$, also known as mean magnetization in the conventional physics literature on the d-dimensional lattices:

$$\mathcal{M} = \frac{1}{|A|^2} \cdot \left\langle \left(\sum_{i=1}^{|A|} \cos\theta_i \right)^2 + \left(\sum_{i=1}^{|A|} \sin\theta_i \right)^2 \right\rangle. \tag{6}$$

This is zero above a critical temperature in many lattice magnetic systems and becomes non-zero spontaneously at low temperatures. Likewise, there is no conventional phase transition present that would be associated with symmetry breaking. However, it is well-known, that system does show signs of a transition from a disordered high-temperature state to a quasi-ordered state below some critical temperature, called the Kosterlitz-Thouless [22] transition.

3.5 Network Bandwidth Model

We propose in our model the definition of bandwidth $\mathcal{B}_{mq}$ between two agents m and q as a non-symmetric positive matrix, based on the following assumptions:

- Mutual spin correlation at a finite noise regime, as stronger correlation results in the better bandwidth;
- Average spin orientation of a agent m projected to the mean-field orientation $\mathcal{M}$ given by the γ-factor, which reflects the synchrony of the agent to the network;

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

– Inter-node degree relation $\Omega(m,q)$, obtained from the graph structure, which reflects asymmetry in the traffic handling. A larger amount of links forces an agent to split its bandwidth;
– Non-symmetric structure of the matrix, handling the realistic scenario of non-identical bandwidth (m,q) and (q,m) permutation.

Accounting these assumptions results in a bandwidth $\mathcal{B}_{mq}$ defined by the following S-curve equation:

$$\mathcal{B}_{mq} = \Omega(m,q) \cdot \gamma_m \cdot \frac{\min(B_m, B_q)}{1 + e^{-R_{mq}}} > 0 \ , \ \forall \ (m,q), \tag{7}$$

where the inter-node degree relation is given by:

$$\Omega(m,q) = 1 - \frac{\deg(m)}{\deg(m) + \deg(q)}, \tag{8}$$

and $\deg(m)$ is the degree of the vertex m and γ-factor is given by vector product:

$$\gamma_m = 1 - \frac{1}{4 \cdot \pi^2} \cdot \frac{\mathcal{M} \cdot (\hat{\phi} - \hat{\theta}_m)^2}{1 + e^{-\mu}}; \qquad \phi = \frac{1}{|A|} \cdot \sum_{m}^{|A|} \theta_m \tag{9}$$

The random variable ϕ expresses the collective field ϕ averaged across the all graph nodes, and μ-factor is the co-variance:

$$\mu = \int (\phi - \hat{\phi}) \cdot (\theta_m - \hat{\theta}_m) \cdot P(\boldsymbol{\theta}) \ d\theta_1 \ldots d\theta_n, \tag{10}$$

which accounts how strong collective field ϕ is "feeling" arbitrary spin vector θ_m orientation. Finally, $\min(B_m, B_q)$ assumed as amplitude of the S-curve function, shows that the channel bandwidth saturates to the minimum of the two possible nominal values defined for each agent. Hats are indicating sample average.

The special case of diagonal elements $\mathcal{B}_{mm}$ corresponding to the dimensionless self-bandwidth of the given agent:

$$\mathcal{B}_{mm} = \gamma_m \cdot \frac{F_m}{e^{\cdot \sqrt{R_{mm}}}}, \tag{11}$$

where $F_m \in (0;1]$ is the nominal normalized dimensionless self-bandwidth, and $\sqrt{R_{mm}}$ is the variance of the local spin orientation, which prescribes decrease of the computational capacity with growing variance.

3.6 Communication and Computation in the Network

We further define a data matrix D_{ij} that indicates the amount of data transmitted from a task i to a task j. Consequently, we obtain the *delay tensor* D^* for transferring data from task i to task j assigned to the agents m and q:

$$d^*_{ijmq} = \overbrace{\mathcal{T}_{mq}}^{\text{Connection Delay}} + \overbrace{\mathcal{D}_{ij} \cdot \mathcal{B}^{-1}_{mq}}^{\text{Data Transfer Latency}} . \tag{12}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

The first term $\mathcal{T}_{mq}$ in Eq. 12 represents a connection estimation delay, assumed as a small constant. This approach models realistic scenarios of synchronized routing information, leading to the fast connection estimation with low delay. The second term results from the time required to transfer the data from task i to j residing on the agents m and q, obtained by dividing the components of data matrix by the corresponding bandwidth. We compute the execution time of the given task $i \in V$ with the following formula:

$$\tau^1_{im} = \max_j \{d^*_{ijmq};\ \ \forall j \in \mathcal{P}_i\} + \tau^0_{im} \cdot \mathcal{B}^{-1}_{mm}, \tag{13}$$

where $\mathcal{P}_i$ is the set of predecessors of task i, the upper indices give the order of the approximation, and the term τ^1_{im} corresponds to the execution time perturbed by the transfer times, meanwhile τ^0_{im} corresponds to the execution time measured in the perfect conditions when there is no noise and data transfer delays. We assume first-order approximation reasonable, when we consider a bandwidth independent of the number of simultaneously transferring agents.

4 Experimental Study

4.1 Monte Carlo Simulation of the Computational Network

We obtain the distribution $P(\boldsymbol{\theta})$ via Monte Carlo simulation using the Metropolis algorithm [5, pp. 108–111], [23]. At each time step, we induce a random walk on the graph N, choose one random spin and rotate its angle by some random increment, keeping it in a range $[0; 2\pi]$. States are accepted with the probability:

$$p(\boldsymbol{\theta}_{n+1} |\, \boldsymbol{\theta}_n) = \min\left\{1, e^{-\Delta\mathcal{H}}\right\}, \tag{14}$$

where $\Delta\mathcal{H} = \mathcal{H}_{n+1} - \mathcal{H}_n$, where n defines number of the step in a random walk. The transitions to the lower energies are certainly accepted, while the transitions to slightly higher energies are accepted with the probability $e^{-\Delta\mathcal{H}}$.

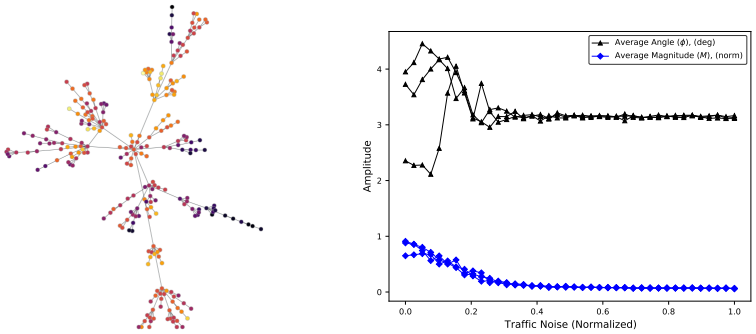
4.2 Implementation Details

We implemented the simulation [16] model in `Python` 3.6 with network processing performed by the graph-tool [29] library. We implemented the scheduler using the `SCIP` 6.0.2 optimization suite [11] with the default configuration and `SoPlex` 4.0.2 LP solver. FB-SGS Heuristic [19, 20] is implemented as stand alone callable application via Python integration wrapper, that we have developed. We compiled the source files for the model and SCIP using `gcc` version 4.8.5 and handled the experimental data using the `BASH` 4.2.46, `MariaDB` 5.5.64 and native NumPy (.npy) format.

4.3 Numerical Results and Discussion

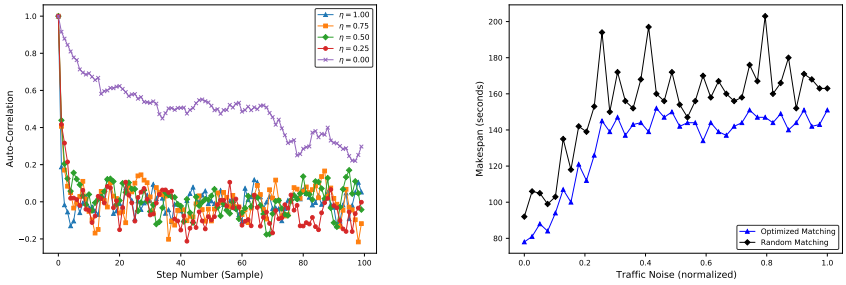
We considered nominal capacities $B_m,\ F_m$ of the computing agents equal to 1 without loss of generality. While this brings homogeneity across agents, the heterogeneity is

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62



(a) (b)

Fig. 1: (a) Snapshot after $1.5 \cdot 10^3$ iterations of the Metropolis-Hastings dynamics of the model defined over Barabasi-Albert network ($n = 256$, $\eta = 0.1$) with triangular initial graph; darker colors correspond to values of θ_m close to 0. Visualized with Fruchterman-Reingold layout algorithm and Cairo library. (b) Order parameters ϕ and $\mathcal{M}$ as functions of the noise regime η.



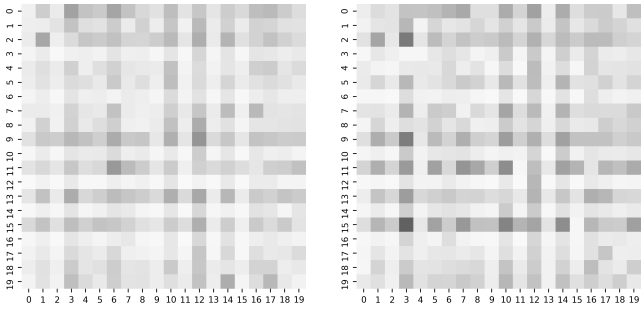
(a) (b)

Fig. 2: (a) Auto-correlation of the global state-transition of the network graph N, with respect to the 0^{th} step. (b) Makespan of the DAG obtained with a variant of the classical local descent with monotonic improvement in the objective function. Single and several ($i = 20$) iterations agent selection vs noise regime variation. Both cases use Forward-Backward SGS (RCPSP/max) for minimum makespan derivation. Task times obtained by scaling normalized execution and transfer times to seconds.

encoded in the network structure and its stochastic behaviour, which depends on the noise regime.

During the first stage of the experiment, we focused on the analysis of the computational network. The topology visualization of the computational network N is depicted on the Fig. 1a. For layout generation we have used force-directed Fruchterman-Reingold

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62



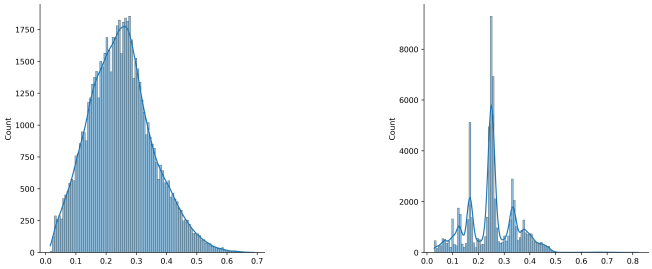
(a) $\eta = 0.50$ (b) $\eta = 0.20$

(c) $\eta = 0.10$ (d) $\eta = 0.05$

Fig. 3: Distribution of the emergent topological excitations $\mathcal{B}_{mq} \in (0; 1]$ on the 20×20 sub-matrix of 256×256 bandwidth matrix, where darker colours correspond to the higher rates of the bandwidth. From left to right, the system evolves from high to low noise influence, exhibiting various computational capabilities of the network N.

algorithm [10] with 2000 iterations. We generated the network structure an from initial triangle graph with three nodes, by selecting one node with probability proportional to the number of its links. We observe heavily linked hubs that tend to quickly accumulate even more links, while nodes with only a few links are unlikely to be chosen and stay peripheral. The new nodes have a "preference" to attach themselves to the already heavily linked nodes. We stopped growth of the network at 256 nodes, since this scales are sufficient [5, pp. 92 – 115] to study emergent properties. We performed a Monte Carlo simulation with 1500 random walks for each of 100 "statistical snapshots" of N with $\eta \in [0; 1]$, taking 14 minutes on average. Figure 1b illustrates the existence of the critical transition around $\eta \sim 0.21$. For further reduction of $\eta \leq 0.20$, one can readily see a rapid growth of the $\mathcal{M}$ order parameter to 1, which indicates that the network achieves a synchronous consensus state. At the same time, the average angle demonstrates a bifurcation-like behaviour. Figures 3 and 4 demonstrate a significant difference in the network operation regimes. Even in a relatively simple homogeneous configuration, we are able to achieve a non-trivial transition of the network bandwidth behaviour. As the noise regime η approaches zero, some agents loose ability to transmit the data and

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62



(a) $\eta = 0.90$ (b) $\eta = 0.05$

Fig. 4: Histogram of the bandwidth matrix with non-parametric kernel density estimation before (a) and after (b) phase transition.

become more "computationally" efficient, as reflected by the bright diagonal in Fig. 3d indicating high computational abilities of the agents. Fig. 4b shows the distribution of the overall bandwidth in the computational network before and after phase transitions, augmented with non-parametric kernel density estimations. It is remarkable how the bandwidth distribution undergoes a critical transition from single to multi-mode case. These important features allow us to simulate complex behaviour of contemporary computational environments, such as computing continuum. Fig. 2a shows an exponential decay of the auto-correlation function. Such ergodic behaviour is natural for Metropolis and Glauber dynamics [5] and indicates that the computational network rapidly looses memory on its initial conditions. The advantage of the ergodic description is that such networks can be described by statistical methods given a sufficient observation time. Depending on the number of transitions in time, this actually rises a question of ergodic hypothesis fairness and suitability for the particular computational network. The discussion of this old issue goes far beyond the scope of this article, however, this approximation is suitable in the low-traffic scenario. We keep the holistic analysis of the more general non-equilibrium scenario for future works in frames of Kinetic schemes and Bayesian's framework.

We further evaluated the behaviour of the DAG network depicted in Fig. 5 for different values of η. It consists of 60 tasks with high inter-dependency and node degrees ranging from 1 to 3. To compute makespan of the DAG network, we use a variant of the classical local descent with monotonic improvements by trying different matching of tasks and agents. Fig. 2b depicts a single ($i = 1$) and multiple ($i = 20$) iterations of the random matching. Once the agents are matched with tasks, both cases use Forward-Backward SGS (RCPSP/max) for minimum makespan derivation, with a complexity of $O\left(J^2 \cdot K\right)$, where $J \subseteq V$ and $K \subseteq N$ are the number of tasks and resources required for execution. We obtain the task times by setting $D_{ij} = 1$, scaling the normalized execution and transfer times to seconds and $\mathcal{T}_{mq} = 0$, according to the Eq. 13. This evaluation demonstrates how complex precedence networks of the tasks can be analysed in the framework of our simulator. Makespan values from Fig. 2b demonstrate high sensitivity in relation to critical transition at $\eta \leq 0.20$, as the computational efficiency

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

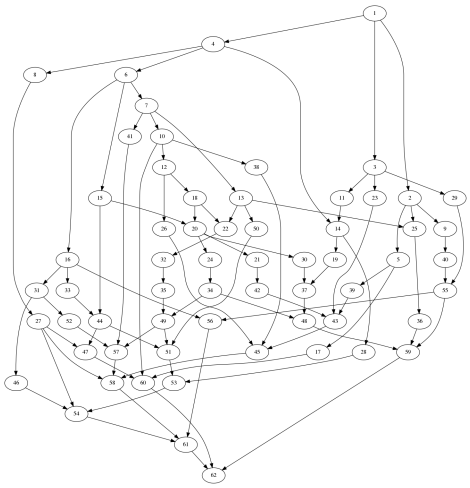


Fig. 5: Directed acyclic graph from PSLIB j60 dataset. Transformed to .dot format and Visualized with GraphViz DOT render engine.

grows. When the amplitude of the noise regime increases, the optimized allocation shows higher makespan values due to the increase in the network disorder. Interestingly, optimized allocation gives not only smaller values of the makespan, but also smaller variance, leading to the more robust estimations of the makespan when the noise regime changes.

5 Discussion and Future Work

In this paper, we analysed the definition of the computing continuum and provided a theoretical model of how high-order computational properties can emerge from elementary pairwise interactions in the context of a topology with preferential attachment policy. We studied the behavior of the computational network by simulating it in the equilibrium regime within the framework of the modified XY-model defined over a complex network with Barabasi-Albert topology. We developed the simulation engine in Python 3.6, which allowed us to study the first order approximation to the behavior of the "high-level" system-wide workflow model in the presence of emergent data-flow anisotropy.

It is important to note that the DAG and scheduling heuristic is non-exhaustive and mainly used for test purposes. We expect to carry out more detailed comparative study of the several heuristics, including Heterogeneous Earliest Finish Time (HEFT), The Dynamic Scaling Consolidation Scheduling (DSCS) [25], Partitioned Balanced Time Scheduling (PBTS) [7] Deadline Constrained Critical Path (DCCP) [2], and Partition Problem-based Dynamic Provisioning and Scheduling (PPDPS) [32]. We hypothesize that Forward-Backward SGS (RCPSP/max) will outperform the aforementioned heuristics in the quality of schedules for a relatively small amount of tasks $|G| \sim 200 - 300$. Particularly interesting is to research this question at a larger scale, where possibly non of

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

these schemes provide satisfactory outcome in terms of convergence speed and quality of the solutions. The only applicable scenario will evolve around multi-agent mapping.We will also continue to work on kinetic schemes and Bayesian's framework to incorporate non-equlibrium phenomena and partial-observations of the network. Integration of the MPI stack to speed up the computations on larger scales and expand our work on new types of network topologies with specific statistics is an important research direction too.

Acknowledgement

This work has been supported by the ASPIDE Project funded by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 801091.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of modern physics **74**(1), 47 (2002)
2. Arabnejad, V., Bubendorfer, K., Ng, B.: Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources. Future Generation Computer Systems **75**, 348–364 (2017)
3. Asch, M., Moore, T., Badia, R., Beck, M., Beckman, P., Bidot, T., Bodin, F., Cappello, F., Choudhary, A., de Supinski, B., et al.: Big data and extreme-scale computing: Pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry. The International Journal of High Performance Computing Applications **32**(4), 435–479 (2018)
4. Axelrod, R.: The dissemination of culture: A model with local convergence and global polarization. Journal of conflict resolution **41**(2), 203–226 (1997)
5. Barrat, A., Barthelemy, M., Vespignani, A.: Dynamical processes on complex networks. Cambridge university press (2008)
6. Beckman, P., Dongarra, J., Ferrier, N., Fox, G., Moore, T., Reed, D., Beck, M.: Harnessing the computing continuum for programming our world. Fog Computing: Theory and Practice pp. 215–230 (2020)
7. Byun, E.K., Kee, Y.S., Kim, J.S., Maeng, S.: Cost optimized provisioning of elastic resources for application workflows. Future Generation Computer Systems **27**(8), 1011–1026 (2011)
8. Copil, G., Moldovan, D., Truong, H.L., Dustdar, S.: Multi-level elasticity control of cloud services. In: International Conference on Service-Oriented Computing. pp. 429–436. Springer (2013)
9. D'Andrea, R., Dullerud, G.E.: Distributed control design for spatially interconnected systems. IEEE Transactions on automatic control **48**(9), 1478–1495 (2003)
10. Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. Software: Practice and experience **21**(11), 1129–1164 (1991)
11. Gleixner, A., et al.: The scip optimization suite 6.0. Tech. Rep. 18-26, ZIB, Takustr. 7, 14195 Berlin (2018)
12. Haken, H.: Synergetics. Physics Bulletin **28**(9), 412 (1977)
13. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to automata theory, languages, and computation. Acm Sigact News **32**(1), 60–65 (2001)
14. Ilyushkin, A., Ali-Eldin, A., Herbst, N., Papadopoulos, A.V., Ghit, B., Epema, D., Iosup, A.: An experimental performance evaluation of autoscaling policies for complex workflows. In: Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering. pp. 75–86 (2017)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62

15. Kadanoff, L.P.: Statistical physics: statics, dynamics and renormalization. World Scientific Publishing Company (2000)
16. Kashansky, V.: Modular architecture for complex computing systems analysis. http://www.edmware.org/macs/, accessed: 2021-01-29
17. Kashansky, V., Kimovski, D., Prodan, R., Agrawal, P., Iuhasz, G., Fabrizio, M., Justyna, M., Garcia-Blas, J.: M3at: Monitoring agents assignment model for data-intensive applications. In: 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). pp. 72–79. IEEE (2020)
18. Klimontovich, Y.L.: Statistical Theory of Open Systems: Volume 1: A Unified Approach to Kinetic Description of Processes in Active Systems, vol. 67. Springer Science & Business Media (2012)
19. Kolisch, R., Drexl, A.: Local search for nonpreemptive multi-mode resource-constrained project scheduling. IIE transactions **29**(11), 987–999 (1997)
20. Kolisch, R., Hartmann, S.: Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis. In: Project scheduling, pp. 147–178. Springer (1999)
21. Kolisch, R., Sprecher, A.: Psplib-a project scheduling problem library: Or software-orsep operations research software exchange program. European journal of operational research **96**(1), 205–216 (1997)
22. Kosterlitz, J.M., Thouless, D.J.: Ordering, metastability and phase transitions in two-dimensional systems. Journal of Physics C: Solid State Physics **6**(7), 1181 (1973)
23. Landau, D.P., Binder, K.: A guide to Monte Carlo simulations in statistical physics. Cambridge university press (2014)
24. Langbort, C., Chandra, R.S., D'Andrea, R.: Distributed control design for systems interconnected over an arbitrary graph. IEEE Transactions on Automatic Control **49**(9), 1502–1519 (2004)
25. Mao, M., Humphrey, M.: Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In: SC'11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 1–12. IEEE (2011)
26. Medvedyeva, K., Holme, P., Minnhagen, P., Kim, B.J.: Dynamic critical behavior of the xy model in small-world networks. Physical Review E **67**(3), 036118 (2003)
27. Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., Troyansky, L.: Determining computational complexity from characteristic 'phase transitions'. Nature **400**(6740), 133–137 (1999)
28. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
29. Peixoto, T.P.: Graph-tool - efficient network analysis. https://graph-tool.skewed.de/, accessed: 2021-01-29
30. Reed, D.A., Dongarra, J.: Exascale computing and big data. Communications of the ACM **58**(7), 56–68 (2015)
31. Reuther, A., Byun, C., Arcand, W., Bestor, D., Bergeron, B., Hubbell, M., Jones, M., Michaleas, P., Prout, A., Rosa, A., et al.: Scalable system scheduling for hpc and big data. Journal of Parallel and Distributed Computing **111**, 76–92 (2018)
32. Singh, V., Gupta, I., Jana, P.K.: A novel cost-efficient approach for deadline-constrained workflow scheduling by dynamic provisioning of resources. Future Generation Computer Systems **79**, 95–110 (2018)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_62