Reconfiguration Durations Optimization for High-availability Distributed Systems: The case of ICT Rural and Elderly Infrastructures for Development

Erick G. Gbetie [1[0000-0001-6397-9728]] and Moindze M. Soidridine [1,2[0000-0002-1760-9977]]
Thierry Oscar Edoh

[1] Ecole Supérieure Multinationale des Télécommunications (ESMT), Dakar, Sénégal
[2] Université Cheikh Anta Diop (UCAD), Dakar, Sénégal
erick.gbetie@esmt.sn, moindze1@yahoo.fr

Abstract. In this paper, we address the problem of very long execution durations during the management activities in rural infrastructures and elderly community information systems. This problem is a major challenge in the NGNM field in terms of constructing ICT4D large-scale collaborative infrastructures for a rural community in developing countries. It is an optimization problem in the strong sense with resource constraints in the context of real-time reconfiguration systems. It is also related to the operability of distributed high-availability networks systems used in NGN communications networks in order to support development in all life areas such as Education, Health, Economy, Agriculture and even to ensure the survival of living beings in developing countries. The resolution of such problems requires a significant and simultaneous reduction of several performance temporal criteria. The reduction of the execution durations allows optimizing performances of network resources. From a practical point of view, the reduction of network resources consumption automatically decreases the overall energy consumption in the remote networks infrastructures in rural areas technologies such as WSN, RFID, NFC, IoT, LoRa, WiMax, AirMax, Wideband satellites access and VSAT.

We propose thus an algorithm capable of determining and decreasing in real-time the execution durations for ICT4D rural infrastructures and rural community information systems. This algorithm allows guaranteeing data access, data integration, information sharing and rapid professionals assistance across multiple heterogeneous centers. This algorithm is based on minimal temporal criteria, lower and upper bounds, dominance rules, a branching scheme, and an exploration strategy. Our experiments show that the results of the proposed algorithm are more efficient than those of the existing reference algorithms.

Keywords: Reconfiguration systems, High-availability, Distributed systems, Execution duration, Temporal criteria; Dominance rules; Lower and upper bounds; QoS-QoE, ICT4D, ICT for Development, Rural Development, Digital divide, Rural Informatics, Rural healthcare system, Rural areas, Service level measurement, Developing countries.

 IREHI-2019: International Conference on rural and elderly health Informatics, Dakar, Sénégal, December 04-06, 2019

1 Introduction

In this work, we address the issue of the diminution of execution durations in the management and reconfiguration operations of new types of switching telephony networks and mobile networks as well as computer infrastructures in rural areas of developing countries. Indeed, the rapid improvements in the ICT field have led to the creation of new telecommunications services [1]. In addition to the traditional voice services offered through historical fixed circuit-switched networks such as PSTN and ISDN [2], new types of services such as VoIP, IPTV and TSTV have emerged with the deployment of very high-speed access standards such as xDSL, CPL and FTTx/PON [3]. Additionally, with the deployment of NGN technologies such as IMS and MPLS in network backbones, the development of multimedia services and value-added services has become a reality [2]. Another continuous evolution concerns cellular and wireless technologies, which have evolved from the second generation (2G) with GSM, GPRS and EDGE standards, deployed in the early 90s, to the third generation (3G) with UMTS, CDMA2000 and HSDPA/HSUPA technologies, standardized in the years 2000, and then to the fourth generation (4G) with LTE and WiMax standards, normalized in early 2010 [4]. A fifth-generation (5G) with MIMO, UDN/SDN, NFV/FBMC, and F-OFDM/LDPC technologies is planned for 2020 by 5GPP [5].

In the IP networks, we also note several recent improvements, notably the ongoing migration from IPv4 to IPv6 in network backbones [6]. These developments enabled the large-scale deployment of dematerialized infrastructures, such as cloud computing [7]. Moreover, the generalization of packet-switching with IPv6 has also introduced the Big Data and Artificial Intelligent concepts [8]. All these innovations have resulted in a technological revolution that has recently launched the concepts of the Internet of Things and home automation or demotics [9].

All these advanced technologies enable multiple-play services to be delivered anywhere and anytime, with any device, creating the "AAA (Anywhere, Anytime, and Any Device)" concept in the area of networks in the cities and developed countries [1, 2].

Recently, the concept of ICT4D (ICT for Development) has introduced in order to support on one hand the remote and isolated communities of indigenous minorities in rural areas and on the other hand to support e-inclusion initiatives strategies development in terms of e-Health, e-Agriculture, e-Government, e-Education, e-Banking and so on [2, 3, 4, 5]. Nowadays, from a user point of view, the use of ICT has been claimed to improve quality of life of diverse types of people in the world especially rural community and people with low economic incomes such as the poor, farmers and fishermen and people with special needs [5, 8, 9].

In the rural areas, recent improved technologies such as WSN (Wireless Sensor Networks), RFID (Radio Frequency Identification), NFC (Near Field Communications), IoT (Internet of Things), WiMax, AirMax, LoRa, Wideband satellites access, VSAT, are emerged to huge digital divide because rural community often are lagging from the cities in terms of economy and public health [2, 4, 6, 7].

All these improved technologies have increased the number of network subscribers and the amount of information exchanged by users in recent years. In the Measurement of the Information Society (MSI) report, published in late 2016, ITU has shown that the total number of mobile subscribers increased from 2.2 billion to 7.5 billion between 2006 and 2016 [10]. It also confirmed that 3.2 billion people or 43% of the world's population are now on-line. Moreover, it showed that the percentage of the population with access to 3G and 3G+ networks has increased from 45% to 69% in the last four years. Mobile broadband subscriptions increased from 0.8 billion in 2010 to 3.5 billion in 2015. Between 2013 and 2016, access prices to fixed and mobile broadband networks fell by more than 55%.

According to IWS [11], the total number of internet users was estimated at 3,731,973,423 from March 31, 2017. This number represents a penetration rate of 49.6% of the world population. In addition, existing end-users require a higher quality of experience (QoE) and not just the availability of services (QoS) [16, 19].

As a result, these developments have created many problems in the network management infrastructures, which generate major research challenges in the field of high dependability notably in the rural areas of developing countries [1, 2, 17]. Indeed, the systems used in today's multi-technology architectures are critical systems with high-availability, which must aggregate a large amount of traffic particularly in the ICT4D infrastructures [8, 12]. These systems must also perform several operations such as scientific calculations, collaborative works, and healthcare information [20]. Resource managers employed in such situations must be able to orchestrate the execution of several jobs simultaneously and efficiently [21]. The systems and network resources must be used optimally, to reduce CAPEX and OPEX costs. Besides, execution durations must be reduced to achieve economic viability of network infrastructures especially in the rural areas [13, 22].

The critical and real-time nature of ICT4D infrastructures such as e-Health systems, e-Agriculture systems, e-Government, e-Education and Digital economy systems leads significant management and reconfiguration issues, such as the diminution of execution durations, the reduction of processing durations, the diminution of system and network resource consumption, the reduction of energy consumption and the decreasing of greenhouse gas emissions from ICT infrastructures [8, 10, 15]. Recently, many works have been proposed in the literature to solve such problems. However, the existing reference algorithms have several shortcomings [2, 12, 14, 16, 17].

In this paper, we propose an algorithm for dynamic management of temporal resources during executions of the management and/or reconfiguration operations of the high-availability systems used in the Next Generation Telecommunication Networks especially in the ICT4D systems in order to demonstrate the opportunities for sustainable development in a remote and isolated rural community from the use of Information and Communication Technologies (ICTs).

This paper is organized as follows: section 2 is devoted to the problem statement. In section 3, we present some challenges related to ICT4D infrastructures for rural and elderly informatics systems. In section 4, we perform a critical analysis of the related works. In section 5 we present our algorithm. In section 6 we evaluate the performance of the algorithm. The results are compared to those of other works in section 7. Section 8 presents a discussion of the results. The last section contains the conclusion and perspectives of our work.

2 Problem statement

The management and reconfiguration activities of distributed high-availability systems such as e-Health systems, e-Agriculture systems, and Digital economic development infrastructures are characterized by several sets of data. The most essential of them is the set of n jobs $J = \{J_1, J_2, \dots, J_n\}$, the set of m distributed systems $M = \{M_1, M_2, \dots, M_m\}$ and $R = \{R_1, R_2, \dots, R_r\}$, the set of resources r they offer [18-20].

The problem of diminution resources in the management and reconfiguration activities of such high-availability systems notably ICT4D systems consists to find in real-time viable allocations in dominant subsets while reducing overall execution durations [21-22]. These viable solutions ensure the utilization of systems without exceeding the work schedule for any set of feasible configurations [9]. Expressly, temporal-based performance criteria are particularly interesting in rural and ICT4D systems for several reasons. On one hand, the finishing of the executions as soon as possible makes to avoid bottlenecks in critical systems like e-Health systems. On the other hand, it ensures load balancing according to traffic priorities and routing rules in high-use vital networks such as e-Education and e-Government infrastructures [7, 12, 15].

From a practical point of view, an ICT4D system is considered as a disjunctive or a cumulative resource consisting of k processors in which operations are performed [4, 18, 22].

The problem of reducing execution durations is presented as a set of jobs to be performed on processors $S = \{S_1, S_2, \dots, S_n\}$ of m systems [6, 19, 26]. Each job J_i represents n operations whose completion requires a number of time units such as the start time s_i, the end date e_i and the processing time p_i. Thus, a job is represented by
$J_i = \{O_{i1}, O_{i2}, \dots, O_{ij}\}$, where O_{ij} is the j^{th} operation of the task J_i.

From the execution point of view, let us consider S, a defective ICT4D system of the partial or overall deterioration performance of the initial management plan. The execution duration problem consists of considerably reducing the duration of job execution while passing from a non-viable to the viable plane without inducing any violation of the capacitive and temporal constraints of under-production systems [22,]. In order to resolve this problem, we propose in this work an algorithm that ensures the determination of several temporal performance criteria intrinsically related to the management and reconfiguration operations. This algorithm moreover ensures a considerable diminution of execution durations during the managing of the critical and high-availability systems used in rural and ICT4D networks. This algorithm guarantees the determination and diminution of several temporal performance criteria intrinsically linked to the reconfiguration operations.

3 Materials and Challenges

The origins of ICT4D (e-Government, e-Health, e-Agriculture, e-Education, e-Digital economy, e-Assistance,…) and rural community informatics systems are associated with the search for technology-based solutions to allow isolated or scattered population access to remote basics services such as health, education, communication, and Internet [1, 2, 10, 8, 11].

Hence, challenges in building collaborative infrastructures for rural areas, rural community and ICT4D systems in developing countries fall into many technical aspects such as data access (centralized, distributed, site to network access, network to user access,…), data integration, technical infrastructure, on one hand [10, 45]. On the other hand, these technology infrastructures require high availability in terms of temporal criteria such as discovery duration, reading duration and completion execution duration [11, 46]. The resolution of any of the issues from these challenges categories could dramatically increase the efficiency of the network [12, 45]. For example, reducing task completion duration could increase systems efficiency and facilitate rapid decision-making in several areas such as e-Health and e-Government [14, 16, 18].

Indeed, for example, primary healthcare systems in developing countries are based generally on health centers (HCs) and health posts (HPs) [13, 42]. From a practical point of view, HCs are usually located in towns with access to telephone networks and HPs are used under HCs in the establishment’s hierarchy. In many developing countries, current health information systems are based on paper [15, 43]. Table I presents the time spent to fill in and to send reports, as well as the costs to send reports in three Latino America rural areas namely Peruvian Initial Application Provinces (PIAP), Chinandega Region in Nicaragua (CReN), and Alto Amazonas Province in the Loreto region of Peru (AAPLReP) [12, 39, 42, 43].

Table I: Example of Time and Costs to fill in and to send reports [12, 39, 42, 43]

Exchange duration	**PIAP**	**CReN**	**AAPLReP**
Monthly time devoted to fill in reports	41 hours	65,3 hours	23,3 hours
Monthly time devoted to send in reports	14 hours	20 hours	27,9 hours
Monthly cost to send reports	$ 20	$ 12	$ 22

Lack of high availability communication systems makes it difficult for quick data access and to confirm data when a possible error is suspected [42-44].

In terms of ICT service access, the digital divide exists between those living in rural and urban areas, uneducated and educated, poor and rich especially in developing countries [42-45]. Many recent works are showed the digital divide that exists among different groups of communities in many countries such as Elderly People in Hong Kong [48], an Aboriginal community in Australia [49] and Deaf Signer User [50].

Currently, multiple efforts and research are ongoing by governments, rural communities and other organizations to bridge the digital divide in different countries such as Strategy/Policy in China [51], Better Internet Connectivity in Africa [52], Artefact Development in Portugal [53], Policy, structure, and application in Turkey [54], Intelligent, Interactive and Adaptive Web application in India [55], Strategic framework in Thailand [56] and Strategic framework in Malaysia [57].

4 Analysis of Existing Algorithms

In this section, we analyze the most important works related to resource management and duration reduction issues especially those related to rural and ICT4D infrastructures.

In [21], the authors proposed an algorithm named DynReconf in order to ensure the reconfiguration of systems in less time. However, it involves a high consumption of system resources and introduces bottlenecks overloading network links. The disadvantages of the Task_Migration algorithm proposed in [22] are among other the amount of memory needed to store the variables and the increasing of the temporal excess costs.

In [23, 24], the authors proposed Avahi and Bonjour algorithms for discovery of nodes and services respectively. These algorithms are interesting but they require using mDNS and DNS-SD protocols as well as D-Bus libraries. Besides, for sequential and or simultaneous operations, the discoveries durations are very high regardless of the size of the network. In [25], the authors proposed the Pastry algorithm for the construction of peer-to-peer network topologies. However, this algorithm does not take into account the location according to the network size. It also leads to rapid growth in discovery times.

The authors of [27-29] have proposed the JMX-Base, the DelayCaracterisation and the DelaysFramworks algorithms respectively for the analysis of the variations of reading durations in the management environments. However, these algorithms are not suitable for estimating read durations in network infrastructures with real-time constraints. In [18], the authors proposed an algorithm named ClockPrecise for clock management in packet-switched networks. This algorithm is very interesting, however, its execution duration and its convergence duration are high.

In [19], the authors have proposed the Process_Move algorithm for reconfiguration systems by reusing the branch and bound algorithms concepts. Although this algorithm is interesting, it is expensive in computing time and generates high durations during its execution. In [30], the authors introduced the concept of acceptable solutions with the Aproximate_Resource algorithm based on simulated annealing. However, this algorithm is expensive in execution durations and generates repeated access to the memory.

The authors of [20, 31-33] have proposed the reconfiguration algorithms Entropy_Cluster, Entropy_Grilles, Cluster_VJobs and Entropy_Consolidate respectively. These algorithms are based on constraint programming and they use the Entropy architecture. These algorithms offer acceptable results but they have several shortcomings. Indeed, any mechanism for suspending non-realizable actions is available. In addition, these algorithms require the use of several bypass nodes in the reconfiguration plane, which are extra storage devices that can be used to store objects temporarily. These bypass nodes introduce other problems related to their processing capacity and storage that must be hardened beforehand. Finally, these algorithms induce very long execution durations.

According to [43], many works related to calculating service level through availability are done for services providers in cities or well-developed areas [44], but, there are not many works related to service availability in the rural areas and ICT4D infrastructures in the developing countries.

Although, many papers provide service level agreement using Ping method [46], but there are not much works using Average durations of discovery nodes, average reading durations of the current configurations, and the completion execution durations of operations particularly in the rural community information technology and remote high availability systems dedicated to rural ICT4D infrastructures such as rural community health centers, e-Learning systems, e-Health systems, e-Agriculture systems in developing countries [45].

The existing reference algorithms have several limitations, among others [26, 35, 36]:

1) The considerable increase of systems discoveries durations;
2) The considerable increase of the information systems reading durations;
3) The considerable increase of the execution duration;
4) The inability of determining the durations corresponding to any type of management operation;
5) The increase of energy consumption in the systems;
6) The increase in the consumption of systems resources (memory, CPU, ...);

7) The increase in the cost of processing configurations;
8) The failure to consider the network capacity of each node, of each link and of each interconnection equipment;
9) The non-consideration of using of the network in terms of traffic;
10) The failure to take routing rules in management plans;
11) The unsecured consolidation of managed entities;
12) The disappointment to take self-management and self-adaptation properties of network infrastructures such as self-detection, self-repair, self-protection, self-configuration, and self-optimization.

5 Proposition of an Algorithm for Decreasing execution durations

In this section, we describe the algorithm that we proposed. We named this algorithm ADRDR (An Algorithm for Determining and Reducing Durations during the management and Reconfiguration of high-availability distributed systems in ICT4D Infrastructures) using temporal criteria, lower and upper bounds, branching scheme and dominance rules and solutions.

5.1 Features of Algorithm

Our algorithm solves the five (5) first problems out of the twelve (12) main problems that we have summarized in the previous section 3. Its key features are:

1) *The decreasing of discovery durations regardless of the types of nodes and or the size of the network;*
2) *The reducing of reading durations regardless of the type and number of nodes;*
3) *The diminishing of execution durations despite network size, technologies used and types of operations;*
4) *The determination of the duration corresponding to any management and or reconfiguration operation;*
5) *The diminution of energy consumption in the systems by reducing calculation and execution durations.*

5.2 Basic elements

Our algorithm is based on the following four basic types of elements:

- ❖ Branching scheme;
- ❖ Lower and upper bounds;
- ❖ Dominance relations;
- ❖ Temporal criteria of minimally.

Branching scheme

The branching scheme represents the set of rules for arranging actions according to the constraints in the variety of N management and reconfiguration jobs [16, 19]. The execution of the rules is done form a search tree. For a management plan P of Φ

works, the strategy consists in carrying out the first task of duration t_i on the available node k, while the i^{th} task of duration h_i must be performed on one of the other m unoccupied nodes while respecting the constraints of precedence and dependencies.

When arranging works on the corresponding nodes of a plane P, we add fictitious works and we also weight the values of the durations obtained with the processing delays [18]. Since some nodes of the plane P are target nodes for allocations from source nodes in the same plane, the eligibility of the final allocations implies that at least one complete job is feasible in the minimum amount of time possible. If moving certain allocations to certain nodes adds significant additional delays, it is necessary to begin work on the free nodes minimizing these delays until a total arrangement with optimal allocations [2, 22].

Dominance rules

Dominance rules represent the constraints that must be added to the initial problem without changing the values of the objective function [37]. We used these rules to evaluate sets of configurations in systems across the lower and upper bounds in order to bring them closer to the overall optimum. The defined dominance rule consists in comparing the execution end durations obtained for all the operations.

Let us consider the sequences (m, q) and (q, m) which are correspond to a couple of configurations (c_m, c_q). We qualify the configurations starting with (q, m) of dominated solutions by those starting with (m, q) if the ending duration is smaller for the couple (m, q) than for (q, m) and vice versa [36]. Thus, the subsets (m, q) are dominant subsets and they contain at least one optimal solution.

From a practical point of view, we declare that a solution is dominant when its execution duration represents the lowest value of all execution durations. Thus, a dominant solution is a viable solution with a short execution duration. We then prove the dominance of the calculated solution by showing that its total execution durations represents the lowest value among the all local minima and global minima of all the configurations obtained at the end of all the management operations.

Lower and upper bounds

In order to determine the minimum and the maximum values of the execution durations, we used lower and upper bounds. The lower bounds allowed us to determine the minorities of the smallest values among all the acceptable solutions [12]. These bounds correspond to the earliest start and end dates of all operations. We have defined two lower bounds namely terminal placement at the end LBES (Lower Bound at the End of Sequence) and at the earliest LBE (Lower Bound at the Earliest).

The LBES bound consists of placing the instances that they have not yet been placed at the end of the sequencing associated with a subset P of instances without taking into account time lags [38]. Then, it consists keeping the smallest duration of end placements on all systems. This bound allows to minimize the smallest durations induced by the execution of first operations.

The LBE bound consists to place as soon as possible the variables of a subset Q at the end of the associated sequencing. It is obtained by considering successively all the operations not appearing in P, but appearing in Q keeping the minimum among the obtained values. This bound allows to minimize the greater duration induced by the execution of the last operations.

The upper bounds are conditions that can influence the reduction of the exploratory domain. Therefore, these bounds permit the algorithm to be limited to the useful information in the exploration procedure [39]. They generally correspond to the start and end dates at the latest of operations. We have defined two upper bounds namely the bound of added by step UBS(Upper Bound by Step) and the bound of added at the end of all the works UBEJ(Upper Bound at the Earliest of Jobs).

The UBS bound consists adding to each step the task not yet placed which minimizes the duration by constructing a feasible allocation at the end of the placement. The UBEJ bound consists to add a job i^* to each iteration of the total arrangement works.

Temporal criteria of minimally

We have characterized a management and reconfiguration operation by many temporal values such as its period $P(C_i)$, its activation request date $D_A(C_i)$, its deadline $D_L(C_i)$, its processing duration $D_E(C_i$, its start date of execution $D_D(C_i)$ and its end date of execution $D_F(C_i)$ which is equal to $D_D(C_i) + D_E(C_i)$ [38]. From the practical point of view, the algorithm uses these temporal criteria for finding the viable and dominant solutions according to the dominance relations. From the point of view of execution, the dominant solutions are kept, while the dominated solutions are pruned.

In order to find dominant and dominated solutions, we use four temporal criteria namely the start dates at the earliest and at the latest, the end dates at the earliest and at the latest, the execution durations at the earliest and at the latest and the completion execution duration. We estimated these criteria based on the relationships described in the sections below [39-41].

The earliest start dates(D_{dp}) of a configuration C_i are given by:

$$D_{dp} = min\ [D_D(C_i)] \qquad (1)$$

The latest start dates (D_{dt}) of a configuration C_i are given by:

$$D_{dt} = max\ [D_D(C_i)] \qquad (2)$$

The earliest end dates (D_{fp}) of a configuration C_i are given by :

$$D_{fp} = min[D_F(C_i)] \qquad (3)$$

The latest end dates of a configuration noted D_{ft} are given by :

$$D_{ft} = min[D_F(C_i)] \qquad (4)$$

Let us denote PD_i, the completion processing duration of a configuration C_i. Let N_C denote the configuration number of a global plane P. Let us also denote d_e the elapsed duration of C_i. For each C_i, we note n_{am} the maximum number of allocations and n_{ae} the exact number such that $n_{ae} \leq n_{am}$. For every plane P, we consider the boolean coefficients of confidence α, β et γ according to the recommendations of [38, 40].

For a set of management and reconfiguration operations Φ, the value of the delay needed in order to execute all operations noted DOE (Delay of all Operations Execution) is determined by :

$$DEO = \left[\left(\sum_{j=1}^{N_c}(n_{ae} \times d_e)\right) \Big/ (n_{am} \times N_c)\right] \qquad (5)$$

For any configuration C_i of Φ, the PD (*Processing Duration*) is calculated by:

$$PD_i = \left(\sum_{i=1}^{q}(w_k)\right); \ \ k = \{1, \ldots, n\} \qquad (6)$$

with

$$w_1 = \alpha \times R \times A_{pp}$$

where

$$\alpha = \begin{cases} 1 & \text{if the allocations are migrables} \\ 0 & \text{otherwise} \end{cases};$$

$$w_2 = \beta \times R \times d_e$$

where

$$\beta = \begin{cases} 1 & \text{if migrations are feasibles} \\ 0 & \text{otherwise} \end{cases};$$

and

$$w_3 = \gamma \times R \times D_E(C_i)$$

where

$$\gamma = \begin{cases} 1 & \text{if operations are feasibles} \\ 0 & \text{otherwise} \end{cases};$$

where

$$R = \begin{cases} 1 \text{ if configurations are feasibles} \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases}.$$

The *Earliest Completion Duration* noted CR_1, is given by the following relation:

$$CR_1 = min\left\{ \sum_{T_i \in \Phi'} PD_i,\ \Phi' \in \Phi \right\} \qquad (7)$$

In the same way, *Latest Completion Duration* noted CR_2 is obtained by:

$$CR_2 = max\left\{ \sum_{T_i \in \Phi'} PD_i,\ \Phi' \in \Phi \right\} \qquad (8)$$

Let CD_Φ(*Completion Duration*) denote the completion durations of all operations in the set of Φ such that :

$$CD_{\mathrm{i}/\Phi} = \left\{ PD_{\mathrm{i}} + \sum_{i=1}^{k} CR_{\mathrm{i}} \right\};\ i = \{1, \dots, l\} \qquad (9)$$

Finally, let us denote TDC(*(Total Duration of Completion)* the total duration of completion of a configuration C_i such as:

$$\mathrm{TDC}_{c_i^k/\Phi} = \mathrm{Min}\{CD_j\}; j \in T \qquad (10)$$

The dominant solutions that our algorithm must retain are those whose total durations TDC_i are the lowest.

5.3 Operation of the algorithm

Using the basic elements presented in the previous sections, our ADRDR algorithm determines the dominant viable solutions. This determination is applied k times where k is a previously fixed threshold value [19, 22, 26]. The algorithm starts with the initialization of the necessary parameters, constraints and various criteria intrinsically related to the execution durations. Subsequently, the set of solutions is also initialized.

Then, the algorithm proceeds by calculating the viable solutions C_i of durations CD_i, by performing all the necessary operations. At each end of the execution of the operations, the algorithm ADRDR retrieves the durations of the viable solutions obtained which represent the local minima and global minima. Finally, the algorithm classifies the solutions according to their durations by checking their dominance. The dominant solutions, which have the lowest total completion duration (TDC) are saved.

5.4 Pseudo-code of the ADRDR algorithm

The pseudo-code executed by the ADRDR algorithm during the determination of dominant and dominated solutions is shown below.

Algorithm: Algorithm for estimating execution durations of all management operations

START
Initialization $\boldsymbol{T}$, the set of management candidates operations;
Initialization $\boldsymbol{\Phi}$, the set of viable and no-viable configurations;
Initialization $\boldsymbol{\Delta}$, the set of no-viable configurations to optimize;
While set of $\boldsymbol{\Delta}$ is not empty ***Do***
  Select the operation $\boldsymbol{t_i}$ whose configuration is most urgent among those of $\boldsymbol{\Delta}$;
   // the first variables declared no-viable are the highest priority;
   If $\boldsymbol{t_{i1}}$ is the first instance of the operation $\boldsymbol{t_i}$ ***Do***
    If $\boldsymbol{t_i}$ was assigned to a single component ***Do***
     Reconfigure the instance $\boldsymbol{t_{i1}}$ on this system;
      If the instance satisfies the conditions ***Do***
       Realize its configuration;
       Calculate the start dates of the other instances of this operation
       $\boldsymbol{S(t_{iz}) = S(t_{i1}) + T(t_i)(z-1)}$;
       // $\boldsymbol{T(t_i)}$ represents the period of operation $\boldsymbol{t_i}$
       // $\boldsymbol{z}$ represents the repetition number of the instance $\boldsymbol{t_{iz}}$
      Else
       Put this instance in the set $\boldsymbol{\Delta}$ to be reconfigured later;
      End If
    Else
     Reconfigure the instance $\boldsymbol{t_i}$ on the location where it satisfies the conditions in terms of duration;
     If the conditions are not respected on any location ***Do***
      Put $\boldsymbol{t_i}$ in the set of $\boldsymbol{\Delta}$ to be reconfigure later;
     End If
      If multiple slots can be chosen ***Do***
     This instance is reconfigured on the one for which the execution duration is lowest;
      End If
    End If
    Calculate the end date of the first configuration of this operation;
    Calculate the start date of the last configuration of this operation;
    Calculate the end date of the last configuration of this operation;
    Calculate start dates for other configurations;
    Calculate the completion dates of other configurations;
   Else
    Reconfigure this variables directly on the start date of execution that was calculated at the time of reconfiguration of the first instance;
    Stop this operation on the completion date that was calculated at when the last instance was reconfigured;
   End If
  Delete the instance $\boldsymbol{t_{i1}}$ from the set $\boldsymbol{\Delta}$;
  Update set $\boldsymbol{\Delta}$ by adding the new candidate configurations;
  Update set $\boldsymbol{T}$ by adding new candidate operations;
  Retrieve the value of the execution start date of the last instance $\boldsymbol{t_{in}}$;

Retrieve the value of the last execution date of the last instance t_{in};
End While
// Determination of execution durations
While the set of **T** is not empty ***Do***
//**T** represents the set of reconfiguration operations containing $\boldsymbol{N}$ configurations
For $\boldsymbol{i = 1}$ to $\boldsymbol{b}$ ***Do***
//$\boldsymbol{b}$ corresponds to the capacity of an operation, i.e size or number of variables
Calculate the approximation of the lower duration $\boldsymbol{DEO}$;
// Sensitivity of configurations ;
Calculate the earliest duration completion $\boldsymbol{CR_1}$;
Calculate the latest duration completion $\boldsymbol{CR_2}$;
Calculate the processing duration $\boldsymbol{PD_i}$ for each operation ;
Calculate the completion duration for all operations $\boldsymbol{CD_i}$ *;*
Determine the values of $\boldsymbol{TDC_i}$ for each candidate for all configurations;
Compare the final solutions according to the values of $\boldsymbol{TDC_i}$;
Choose the lowest values of $\boldsymbol{TDC_i}$ among those calculated;
Consider the corresponding solutions as the dominant solutions;
End For
Recover dominant solutions;
End While
END

6 Computational Experiments

In this section, we detail the performance evaluations of our algorithm in terms of execution durations.

6.1 Runtime experiments environment

Here we briefly describe the software implementing ADRDR algorithm and the framework in which the series of experiments were performed.

System environment

All experiments were performed on a Sony Vaio computer with Intel (R) Core (TM) CPU i5-2430M @ 2.4GHz, 2.40 GHz of microprocessor, 8.00 GB of RAM memory, 500 GB of storage hard disk and the Windows 7 Professional SP1 platform with a 64-bit operation system.

Experimentation software

We have implemented the ADRDR algorithm in an experimental software that we have developed. This software is written from C, CGI/Shell, Java/JavaScript and PHP-POO programming languages. For the construction of generalized graphics for a better visualization of results graphically, the Jfreechart and Graphviz libraries are incorporated into this software .

Experimentation network

We evaluated the behavior of ADRDR algorithm on a heterogeneous, multiplatform and multi-system infrastructure of a local and wide area networks. This infrastructure is composed of several different nodes with diverse operating systems thus providing different nature of services.

6.2 Computational and preliminaries results

We report the computational results through the following performance criteria:

1) The average durations of discovery nodes;
2) The average reading durations of the current configurations in the nodes;
3) The completion execution durations of operations.

The choice of these three criteria is based in the fact that in the context of the resolution of management and or reconfiguration problems, many recent works have studied these metrics. In addition, these criteria represent reference indicators in terms of predicting the swiftness of management and reconfiguration algorithms. Finally, these indicators allow measuring the efficiency of an algorithm in terms of reducing the consumption of temporal resources in order to improve data access and sharing information in remote high availability systems.

Evaluation of discovery nodes durations

Figure 1 shows the average discovery times based on the total number of nodes discovered in a wide area network.

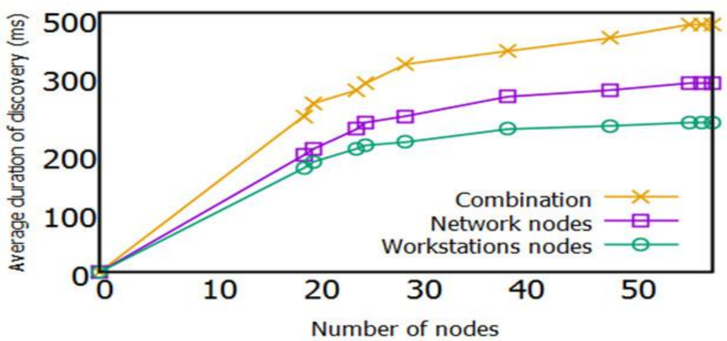

Figure 1 : Average durations of discovery nodes in a WAN

These results show that ADRDR algorithm is able to determine the durations corresponding to the discovery operations of any type of system (network equipments, workstations, servers or combinations of systems). These results also show that discovery durations fluctuate slightly regardless of the considered criteria (number of nodes, types, nature, area of networks,...). Indeed, many authors have shown that discovery durations less than or equal to 2000 ms are acceptable for the discovery of fewer than 50 nodes in large-scale ICT4D networks [16, 20]. These results corroborate the resolution of the first problem related to the diminution of the discovery durations by ADRDR algorithm.

Evaluation of data reading durations

Figure 2 shows the average reading times of configurations on active systems. This figure shows the variation of this duration according to the type of configurations. We observe reading times limited to 120 ms.

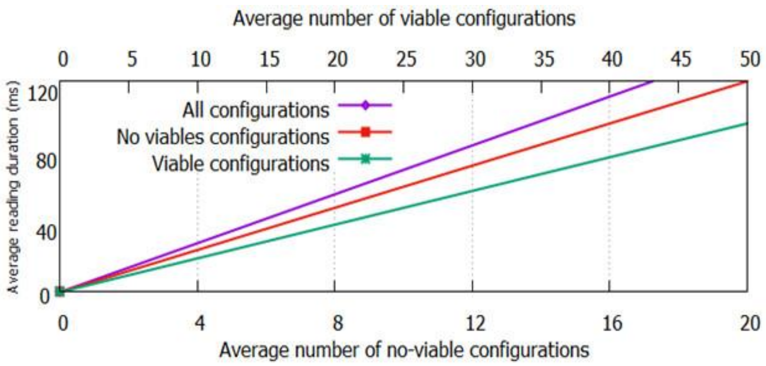

Figure 2: Average durations of data reading

These results show that ADRDR algorithm is able to quickly read all configurations in all types of systems and estimate and reduce the corresponding times. Many authors have shown that read times of less than or equal to 1000 ms ensure that management and reconfiguration operations are triggered as soon as possible in large scale ICT4D infrastructures [15, 31]. Thus, the results obtained confirm the resolution by ADRDR algorithm of the second problem relating to the reduction of the durations of data reading information.

Evaluation of the completion execution durations

Figure 3 gives the run times as a function of the number of nodes and the number of parameters executed in each node.

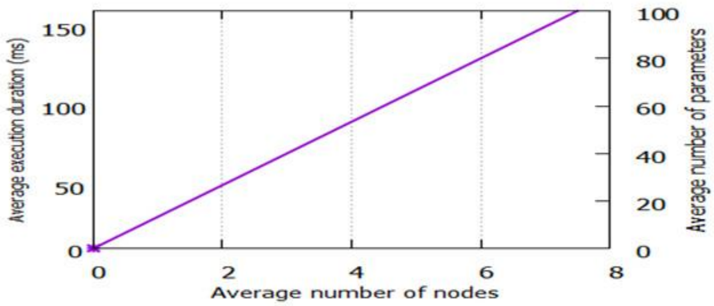

Figure 3: Average execution durations

The results of this evaluation illustrate the capabilities of ADRDR algorithm in terms of high-speed execution of operations. We note that execution durations are low regardless of the type of operation performed or the nature of the nodes. These durations remain acceptable as part of the real-time management and reconfiguration of highly available distributed systems. In fact, for such systems, runtimes of less than or equal to 1000ms are required to ensure their availability and reliability in large-scale ICT4D networks [18, 21]. These results confirm the resolution of the problem relating to the reduction of the completion execution duration by ADRDR algorithm.

7 Comparison of Results

In this section, we compare ADRDR algorithm with other algorithms of the literature.

7.1 Comparison in terms of discovery nodes durations

The objective of this comparison is to study the capacity of overcrowding in terms of systems of algorithms. For this comparison, we use a sequential discovery where each node requests a registration at a time t without the list of nodes already collected being initialized. We compare ADRDR algorithm with Avahi [23], Bonjour [24] and Pastry [25] algorithms. Figure 4 illustrates the nodes recording times for each algorithm. We note that the number of discovered nodes has a significant impact on the resolution process for Bonjour, Avahi and Pastry algorithms. Compared to others, ADRDR algorithm obtains significantly weak discovery durations.

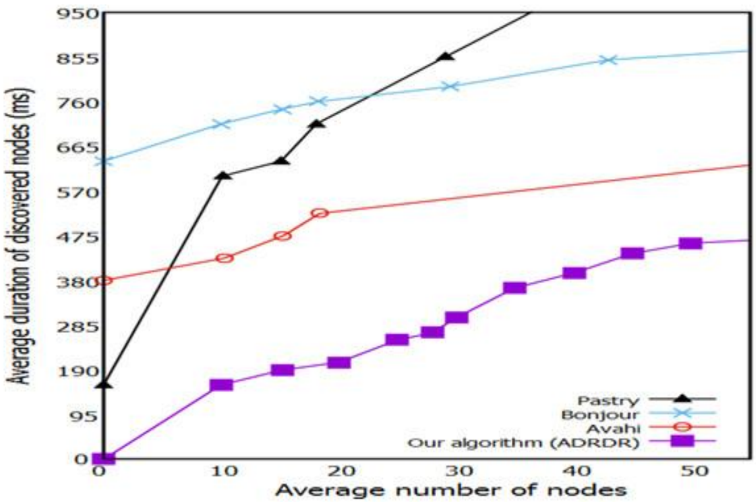

Figure 4: Comparison of discovery nodes durations

7.2 Comparison in terms of data reading durations

In this section, we compared ADRDR algorithm with existing algorithms in terms of systems data read times. We compared ADRDR algorithm with ClockPrecise [18], JMX-Base [27], DelayCaracterisation [28] and DelaysFramworks [29] algorithms. Figure 5 gives the results.

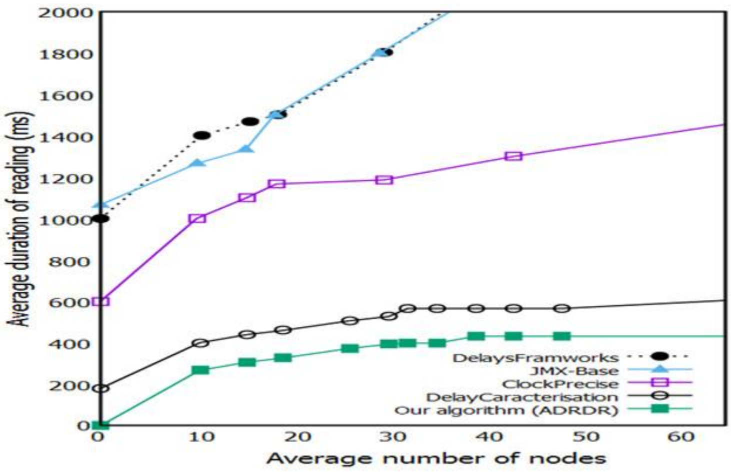

Figure 5: Comparison of data reading durations

These results show that ADRDR algorithm ensures the data reading in very low durations than the others whatever the number and types of reading values. These results also show that the number of nodes and their nature do not have a great influence on the reading times for ADRDR algorithm.

7.3 Comparison in terms of execution durations

In this section, we compared the practical relevance of ADRDR algorithm in terms of execution times. The diminution of this criterion makes it possible to considerably reduce the systems unavailability periods to the strict minimum especially in large-scales ICT4D networks. We compared the execution durations of ADRDR algorithm with those obtained by Process_Move [19], Entropy_Cluster [20], DynReconf [21], Task_Migration [22], Aproximate_Resource [30], Entropy_Grilles [31], Cluster_VJobs [32] and Entropy_Consolidate [33] algorithms. Figure 6 shows the results of this comparison.

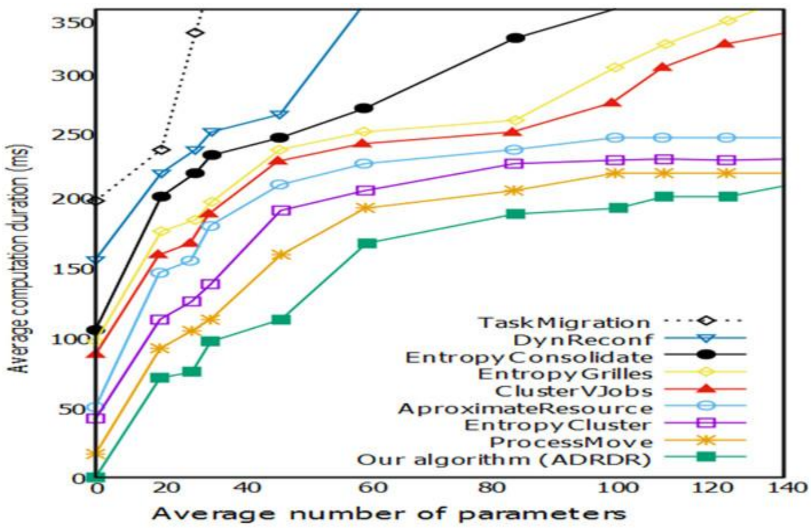

Figure 6: Comparison in terms of execution durations

These results show that the management and reconfiguration operations are executed by ADRDR algorithm with very inferior durations to those of the other algorithms in spite of their types. Moreover, we note that the ADRDR algorithm durations are lower than the thresholds required for such operations. Indeed, several reference works have been shown that to ensure the availability of systems in terms of continuity of services and their reliability in terms of fidelity of operation, the execution times must be less than or equal to 230 ms for the modification of the parameters according to their number.

8 Discussion

The experimental results presented in the previous sections show the efficient performance of ADRDR algorithm in terms of reducing the consumption of temporal resources during the processing of management and or reconfiguration operations in the high-availability systems used in the new ICT4D telecommunication networks such as rural areas technologies in order to guarantee the ICT4D services such as e-Health, e-Government and e-Agriculture.

As for the discovery durations, Bonjour, Avahi and Pastry algorithms obtained values between 100 and 950 ms for a number of nodes varying between 1 and 50. For the same node interval, ADRDR algorithm obtained durations varying between 1 and 450 ms.

Compared to the reading durations, JMX-Base, DelayCaracterisation, ClockPrecise and DelaysFramworks algorithms obtained values between 150 and 2500 ms. ADRDR algorithm ensures the reading of information with durations varying on average between 1 and 450ms.

Finally, concerning to the completion execution duartions, reference algorithms such as Process_Move, Aproximate_Resource, Entropy_Cluster, Cluster_VJobs, Entropy_Grilles, Entropy_Consolidate, DynReconf and Task_Migration have obtained durations varying on average between 10 and 900 ms. For the same types of execution, ADRDR algorithm executes operations with maximum durations varying on average between 1 and 250 ms.

The obtained results show the effectiveness of the adopted strategy with using dominance rules, lower and upper bounds, etc. These results moreover show that ADRDR algorithm is able to determine execution durations for any type of operation. Finally, they show the interest of this algorithm in an energy saving perspective, given the considerable reduction in execution durations. Indeed, the reduction of the consumption of the temporal resources automatically leads to a reduction of the energy consumption in the systems. This is an important result in rural areas where the energy represents a most problems [2, 10]. Certainly, the reduction of the consumption of the temporal resources in high distributed systems automatically leads to guarantee efficiency, usefulness, performance and tolerance in ICT4D community infrastructures [1, 22].

9 Conclusion and Perspectives

In this paper, we have proposed the ADRDR algorithm to efficiently manage real-time resources during the management and or reconfiguration of high-availability systems in rural and ICT4D infrastructures. This algorithm ensures the determination of execution durations corresponding to any management and reconfiguration operation. It also ensures the reduction of these durations by using temporal minimally criteria, dominant solutions, lower and upper bounds, dominance rules and an exploration strategy.

The obtained results are efficient in terms of reduced execution times. This decrease allows ADRDR algorithm to guarantee a reduction of the total consumption of energy in the systems used in the last generation distributed networks particularly in rural and ICT4D Technologies. From a practical point of view, the reduction of energy consumption in under production systems reduces the greenhouse gas emissions of ICT infrastructure. This ensures a better protection of the environment when using technologies.

Taking into account the general analysis of results we outlined, a high availability computer-based system (reducing of total tasks completion duration) notably in distributed voice systems, distributed data, and video systems in rural ICT infrastructures of developing countries could improve the ICT4D results in a rural community such as epidemiological surveillance system, emergency management, doubt consultation, elderly people assistant and could be used for distance training and e-learning.

Several perspectives are being studied in order to improve this work. Firstly, in the experimental plane, we are limited to 50 nodes during evaluations. We then want to evaluate the performance of ADRDR algorithm in terms of reducing execution times when managing and reconfiguring systems in large-scale infrastructures containing thousands of systems such as cloud computing environments [7], green cloud computing [8] and home automation [9] as well as e-Agriculture infrastructures.

Currently, ADRDR algorithm does not take into account the costs introduced by congestion network links. We are working on the definition of an economic function relative to the average costs of network links in order to ensure a better load balancing, especially whilst solving large-scale management problems.

Finally, we plan to make ADRDR autonomous algorithm by integrating the essential autonomous management properties, namely self-detection, self-repair, self-protection, self-configuration and self-optimization [12, 13]. This extension will allow ADRDR algorithm to ensure operations execution quickly and efficiently, without human intervention.

The result gained from this study may provide insights for further e-inclusion initiatives in terms of rural informatics engineering for rural communities. The value chain can be further integrated with many strategies towards the implementation of ICT4D based initiatives such as e-Education, e-Agriculture, e-Health, e-Government, e-Commerce for the rural communities in the developing countries in Africa.

References

1. T. Plevyak., A. Salah, "*Managing IP Networks: Challenges and Opportunities,* 384 pages, ISBN: 978-0-471-39299-6, Wiley-IEEE Press (2003)
2. P. Thomas, S. Veli, “NGN: Next generation telecommunications networks, services, and management”, IEEE press series on network management, 323p, ISBN 978-0-470-57528-4 (2010)
3. L. Sami, Pertti R, *”PON: Passive Optical Networks, Transport concepts*”, VTT Tech Res Centre of Finland, 18pages (2007)
4. R. Ajay R, “*Advanced cellular network planning and optimization 2G/2.5G/3G, evolution to 4G*”, John Wiley & Sons, 544p (2007)
5. http://www.networld2020.eu/relationship-to-5g-ppp/, last accessed 2019/08/20
6. P. Guy, *"Les réseaux*", EYROLLES, Edition 2006, ISBN : 2-212- 11987-9, 1124 pages (2006)
7. M. Cafaro, G. Aloisio, “*Grids, Clouds Computing and Virtualization”, Comp Comm and Net*, Springer, New York (2011)

8. Intel, *"Intel's IT Manager Survey on How Organizations Are Using Big Data:*http://www.intel.com/content/dam/www/public/us/en/documents/reports/data-insights-peer-research-report.pdf, last accessed 2019/07/22
9. B. Renaud, Guillaume, "*Objets communicants et internet des objets (IoT)",* Livre blanc, VF1.0, 2011, Institut Carnot (2011)
10. M. Clark et al, "Mesurer le degré de développement des TIC: Nouvelles tendances, nouveaux défis", UIT rapport, Japon, ISSN 1020–4156 (2016)
11. Internet World stat (2016), www.internetworldstats.com/stats.htm, estimated users, last accessed 2019/10/25
12. D. Jianguo, “Advances in Network Management”, Taylor and Francis Group, LLC, 375p, ISBN, 978-1-4200-6452-0 (2010)
13. A. Nazim, “Autonomic Network Management Principles, from concepts to applications”, Published by Elsevier Inc, 307p (2011)
14. M. Soidridine, Dr K. Konate, “A Survey of the distributed network management models and architectures: Assessment and Challenges”, 6th IEEE Conference, International Conference on Adaptive Science and Technology, Covenant University, Otta, Ogun State, Nigeria, 29-31, oct 2014 (2014)
15. M. Moindze Ali, N. Martaj, M. Mahdaoui, *"Flow And Heat Transfer In Thermoacoustic Resonators Implicit 2D Large Eddy Simulations",* 3rd IRSEC (2015)
16. R. Sirdey R, H. Kerivin, “Polyèdres et reconfiguration dynamique d’autocommutateurs répartis”, In JPOC3, (2006)
17. M. Soidridine, Pr K. Konate, “Critical analysis of the scientific literature in systems reconfiguration field: Assessment and Challenge”, In Proc of the Inter Conf on Ind Elect and Elect Eng, Chennai, India, Dec 25-262016 (2016)
18. R. Sirdey, F. Maurice, "A linear programming approach to highly precise clock synchronization over a packet network", 4OR 6:393- 401, 2008
19. R. Sirdey, J. Carlier, H. Kerivin, “*On a resource-constrained scheduling problem with application to distributed systems reconfiguration*”, Eur Journal of Oper Research, 183:546–563, 2007 (2007)
20. H. Fabien, X. Lorca, J. Menaud, "Entropy : a consolidation manager for clusters. In VEE SIGPLAN/SIGOPS, ACM on Virtual execution environments, pp 41-50, New York, NY, USA, 2009 (2009)
21. L. José, J. García, “Dynamic reconfiguration of node location in wormhole networks”, Journal SA 46, 873-888, Elsevier, 2000 (2000)
22. T. Megel, M. Jan, V. David, "*Evaluation of task migration mechanisms for hard real-time distributed systems*", In *RTNS’11:Proceedings of the 19th ICRTNS*, 2011 (2011)
23. A. Heithem, C. Christophe, J. Dubacq, "*Etude de performance des systèmes de découverte de ressources*", RenPar’18 / SympA’2008 / CFSE’6, Fribourg, 11-13/02, (2008)
24. S. Daniel, C. Stuart, *"Zero Configuration Networking: The Definitive Guide*", O’Reilly 1st Ed, Dec 2005 (2005)
25. I. Antony et al., *"Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems",* IFIP/ACM Inter Conference on Distributed Systems Platforms, p.329-350, UK, (2001)
26. G. Hassan et al.,*“Minimizing the Makespan in a Flow Shop Scheduling Problem with Sequence-Dependent Setup Times and Periodic Maintenance by a Hybrid Algorithm”*, ICIEOM, pp 806-814, (2012)
27. L. Abdelkader, L. Andrey, O. Festor, " On the Impact of Management on the Performance of a Managed System: a JMX- Based Management Case Study", 16th IFIP, IEEE Distributed System, Operations and management, p24-37, Spain (2005)

28. F. Olivier, A. Lahmadi, L. Andrey, *"Caractérisation des délais dans les applications de supervision de réseaux et de services"*, Technique et Science Informatiques (TSI) 28, 4/2009, pp 479-502
29. A. Lahmadi, A. Laurent,F. Olivier, "On delays in management frameworks: Metrics, models and analysis", In International Workshop on Distributed Systems: Operations and Management, p. 13-24, Springer Berlin Heidelberg, (2006)
30. R. Sirdey, J. Carlier, D. Nace, "*Approximate resolution of a resource constrained scheduling problem*", *Jour of Heuristics*, 15:1–17, (2009)
31. H. Fabie, X. Lorca et al., "Reconfiguration automatique du placement dans les grilles de calculs dirigée par des objectifs", In Proc. of 6eme Conf, Swiss, fev. (2008)
32. H. Fabien, A. Lebre, "*Cluster-Wide Context Switch of Virtualized Jobs*", Res Rep RR-6929,INRIA, (2009)
33. H. Fabien, X. Lorca, M. Jean-Marc, *" Entropy :a consolidation manager for clusters"*, In USENIX Symposium on Operating Systems Design and Implementation (OSDI) San Diego, Dec 2008 (2008)
34. M. Soidridine, Pr K. Konate, "A multivariables Algorithm for Dynamic Reconfiguration of real-time Distributed Systems", In Proc of IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC, pp.960-968, Xi'an, China, Oct 3-5, 2016 (2016)
35. H. Fabien, A Lebre, J. Menaud. Cluster-Wide Context Switch of Virtualized Jobs. Research Report RR-6929, INRIA, (2009)
36. E. Mokotoff E, "Minimizing the Makespan and Total Flow Time on the Permutation Flow Shop Scheduling Problem", In MISTA, pp 479-505, Dublin, 10-12, Aug 2009 (2009)
37. T. Ibaraki, "The power of dominance relations in branch-and-bound algorithms", *Journal of the ACM*, 24:264–279, 1977 (1977)
38. B. Jorgen, G. Gutin, "Digraphs Theory, Algorithms and Applications", Springer-Verlag, 772p, 2007 (2007)
39. B. Peter, K. Sigrid, "Complex Scheduling", Springer book, 292p, Gor public, ISBN-10 3-540-29545-3
40. J. Hao, P. Galinier, M. Habib, "Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes", Revue d'Intell Artif, vol.1999, pp.2-39, (1999)
41. H. Kerivin, R. Sirdey, "Polyhedral combinatorics of a resource- constrained ordering problem part I: on the partial linear ordering polytope", Rapport technique PE/BSC/INF/017912 V01/EN, Nortel GSM Access R&D, France, (2006)
42. A. Martinez, V. Villarroel, J. Seoane, F. Pozo, "", IEEE Trans on Inf Tech in Biomedecine, vol 9, N1, (2005)
43. C. Ivan, R. Nana, M. Ian, " Service Level Measurement Based On Uptime Data Monitoring For Rural Internet Access Services in Indonesia", IEEE conf, (2017)
44. P. E. Pratiwi, A. F. Isnawati and A. Hikmaturokhman, "Analisis QoS pada Jaringan Multi Protocol Label Switching (MPLS) Studi Kasus di Pelabuhan Indonesia III Cabang Tanjung Intan Cilacap," *Jurnal Infotel,(*2012)
45. A. Arisyakbani, Analisis Paramater Service Level Agreement (SLA) Pada Layanan Metro Ethernet PT Telkom Untuk Wireless Broadband Telkomsel (3G) dari Apartemen Widya Chandra ke STO Semanggi 2 Dengan RNC di Wisma Mulia Lantai 4, Depok: Universitas Indonesia, (2010)
46. SolarWInds, "SolarWinds Quick Tip: Mitigate Network Downtime by Monitoring CPU Load in Real-Time," (2016)

47. A. Wan, A. Lokman, S. Aris, " Investigating Technology Readiness of Rural Community: An Analysis of Age Differences", IEEE BEIAC (Bus Eng and Ind App Coll), pp-177-182, (2013)
48. J. Lam and M. Lee, “Bridging the Digital Divide - The Roles of Internet Self-Efficacy towards Learning Computer and the Internet among Elderly in Hong Kong, China,” *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 200.*, (2005)
49. M. Mohammad, M., and Y. Lan., “The dynamic business process development framework to bridge the digital divide among the aboriginal community in Australia”, *2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME), (*2010)
50. I. Fajardo, J. Abascal, J. Jose, “Bridging the digital divide for deaf signer users,” In *Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction* (ECCE '08), Julio Abascal, Inmaculada Fajardo, and Ian Oakley (Eds.). ACM, New York, NY, USA, Article 37, (2008)
51. L. Wei,., “Bridging the digital divide: china's strategy,” *Inter Conference on Information Management, Innovation Management and Industrial Engineering (ICIII), (*2011)
52. B. Barry, C. Barton,L., Kalim, U., Petitdidier, M., and Rabiu, B., “eGY-Africa: Better Internet connectivity to reduce the digital divide,” *IST-Africa, 2010,* (2010)
53. D. Carvalho, M. Bessa, E. Peres, E., Magalhaes, L., Guedes, C., and Oliveira, L., “Developing a multi-touch serious game to fight the digital divide: The Portuguese ATM: A pilot case study,” *7th Iberian Conf on Infor Sys and Tech (CISTI), (*2012)
54. M. Bayrakci, M., “E-inclusion for bridging the digital divide in Turkey,” *Inter Conf on Application of Information and Communication Technologies, AICT, (*2009)
55. S. Patel, O. Bataveljic, Lisboa, P.J., Hawkins,“Shakti--Crossing the Digital Divide in Rural India,” *IEEE/WIC/ACM Inter Conf on Web Intelligence WI, (*2006)
56. P.Punyabukkana, S. Thanawastien, S., and Jirachiefpattana, A., “Thailand's national digital divide strategic framework,” In *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A) (*W4A '08). ACM, New York, NY, USA, pp. 97-100 (2008)
57. R. Ramachandran, “A policy tool on developing a national strategic framework on bridging digital divide: a Malaysian public policy perspective,” In *Proceedings of the 2nd international conference on Theory and practice of electronic governance* (ICEGOV), Tomasz J. and Theresa A. P., Eds. ACM NY, USA, pp. 447-451, (2008)