Towards Ubiquitous Computing Clouds

Ahmed Khalifa
Bradley Department of Electrical and Computer Engineering, Virginia Tech
Blacksburg, Virginia, USA
Email: akhalifa@vt.edu

Riham Hassan
Computer Science, Virginia Tech
Blacksburg, Virginia, USA
Email: rhabdel@vt.edu

Mohamed Eltoweissy
Bradley Department of Electrical and Computer Engineering, Virginia Tech
Blacksburg, Virginia, USA
Email: toweissy@vt.edu

***Abstract*— With the emergence of ubiquitous resource-infinite computing, the opportunity arises to significantly enhance on-demand resource availability, particularly for under-connected areas, extreme environments and distress situations. To this end, we propose a new concept, termed PlanetCloud, that enables the provisioning of the right, otherwise idle, fixed and mobile resources to effect ubiquitous computing clouds. PlanetCloud utilizes our new opt-in, dynamic spatio-temporal resource calendaring mechanism that aims at providing a dynamic real-time resource scheduling and tracking system. Analysis of dynamic data, from both calendars and social networking, enhances resource forecasting and provides the right-sized cloud resources anytime and anywhere. Our solution enhances computing effectiveness and efficiency by forming local and hybrid clouds to increase cloud availability. In addition, the calendaring mechanism empowers PlanetCloud with an on-demand scalable computing capability through inter-cloud cooperation to provide additional resources beyond single cloud computing capabilities.**

Keywords-Cloud computing; Ubiquitous computing; Social networking; Distributed system; Resource scheduling; Graph theory.

I. INTRODUCTION

The number of mobile nodes is increasing rapidly, which forms massive computing resources. At the same time, there is a need to exploit their idle resources as suggested in [1] Recently, cloud computing has enabled users to exploit outsourced computing power from fixed servers through the Internet [2]. However, Internet connectivity is not always available, especially in rural areas. We advocate the decoupling of computing resources from Internet availability, thus enabling Ubiquitous Computing Cloud (UCC). The UCC utilizes the massive pool of computing resources, in both fixed and mobile nodes.

As a working scenario, consider a resource-provisioning scenario for field missions of the Medicins Sans Frontieres (MSF) organization [3]. MSF provides primary healthcare and assistance to people suffering from distress or even disaster anywhere in the world. Before a field mission is established in a country, a MSF team visits the area to determine the nature of the humanitarian emergency, the level of safety in the area and what type of aid is needed. The field mission might arrive many days after the disaster occurs. A vast quantity of data on several factors, damages and losses, are collected and analyzed during each field mission. However, field missions depend only on their limited resources, which leads to delayed reports to governing bodies to take the right decisions. Moreover, it is difficult to monitor and track the other available resources to cooperate in performing their tasks. Although MSF has consistently attempted to increase media coverage of the situation in these areas to increase international support, there is a lack of support to media coverage due to extreme conditions, e.g. lack of connectivity.

Volunteers of the field missions might travel with their resources among different locations due to the changes in the situation in disaster zones. Some volunteers and infrastructure resources may have withdrawn or be lost due to deteriorated security or disaster damage impact.

MSF, like many other organizations seeking services that require resources, may suffer from the tight coupling between the current cloud computing and the Internet infrastructure. Such coupling sometimes leads to service disruption while decreasing resource availability and computing efficiency. Currently, there is a massive amount of idle computing resources in fixed and mobile nodes. However, we lack an effective resource allocation and scheduling mechanism capable of exploiting these resources to support ubiquitous computing clouds.

A possible solution would be to provide a dynamic real-time resource scheduling, tracking, and forecasting of resources. Further, the solution should enhance the computing efficiency, provide ”on-demand” scalable computing capabilities, increase availability, and enable new economic models for computing service.

We propose a ubiquitous computing clouds' environment, PlanetCloud, which adopts a novel distributed spatio-temporal calendaring mechanism with real-time synchronization. This mechanism provides dynamic real-time resource scheduling and tracking, which increases cloud availability, by discovering, scheduling and provisioning the right-sized cloud resources anytime and anywhere. Our solution would provide the resources needed by the MSF field missions while they travel among different locations. In addition, it would provide a resource forecasting mechanism by using spatio-temporal calendaring coupled with social network analysis. In situations similar to MSF's, forecasting of resource availability is invaluable before or even during disaster occurrence to save the time required for both surveys and decision-making. PlanetCloud might discover that uploading or downloading the data to or from a stationary cloud is prohibitively costly in time and money. In this

 ISBN: 978-1-61208-154-0

situation, a group can request resources from PlanetCloud to form an on-demand local clouds or hybrid clouds to enhance the computation efficiency. In case of the MSF scenario, this could be used, if the disaster for example warrants evacuation, to assign evacuation routes for the movement of tens of thousands of evacuees.

PlanetCloud provides "on-demand" scalable computing capabilities by enabling cooperation among clouds to provide extra resources beyond their computing capabilities. In the above scenario, PlanetCloud may provide cooperation among the mission field's cloud and the UN vehicle computing cloud to increase the support of media coverage.

The rest of the paper is organized as follows. In Section II, we give an overview of related works and address the gaps between literature works and our proposed approach. We then detail the architecture in Section III. In Section IV, we present the calendaring mechanism of the proposed approach. We discuss our evaluation approach in Section V. Finally, we present open research issues and conclude this paper in Section VI.

II. RELATED WORKS

The state of the art in research shows four categories of solutions for scheduling and allocating resources relevant to supporting ubiquitous computing clouds. They are resource mapping and discovery techniques, exploiting idle resource approaches, approaches for analysis of complex graphs and networks, and tracking applications.

The resource mapping problem has been broadly considered in the literature. NETEMBED considered allocating resources when deploying a distributed application [4]. Internet Maps are useful for tracking the Internet evolution and studying its properties. In [5], an approach is presented to automatically generate world-wide maps by using traceroute measurement from multiple locations. A detailed survey of various decentralized resource discovery techniques is introduced in [6]. These techniques are driven by the Peer-to-Peer (P2P) network model. A layered architecture was presented to build an Internet-based distributed resource indexing system.

Exploiting idle resources has been proposed in some work. For example, Search for Extra-Terrestrial Intelligence (SETI) @ home focuses on analyzing radio signals, and searching for signs of extra terrestrial intelligence. The software of SETI@home runs either as a screen saver on home computer and only processing data when the screen saver is active, making use of processor idle time [7].

The third category of solutions includes approaches for analysis of complex graphs and networks. Authors in [8] used an extensible data structure for massive graphs: STINGER (Spatio-Temporal Interaction Networks and Graphs (STING) Extensible Representation). It includes a computational approach for the analysis based on the streaming input of spatio-temporal data. GraphCT, a Graph Characterization Toolkit, for massive graphs representing social network data has been presented in [9]. It has been used to analyze public data from Twitter, a micro-blogging network. This work treated social network interactions as a graph and used graph metrics to ascribe importance within the network.

The fourth category includes tracking applications. In [10], an application for the Apple iPhone™ was presented and used to report in real-time flow of traffic in order to build the most accurate map of traffic patterns to be used by commuters or departments of transportation for making decisions. The outcome of the case study is used to determine that the iPhone™ is relatively as accurate as a vehicle tracking device.

Both resource mapping [4, 5, 6] and idle resource exploitation approaches [7] consider stationary resources. Such resources are globally distributed and directly connected to the Internet, as Internet-based systems. On the other hand, PlanetCloud enables ubiquitous computing clouds in a dynamic resource environment by exploiting idle resources that could be either fixed or mobile. In addition, it supports cloud mobility and hybrid cloud formation. PlanetCloud extends the concept of social network analysis presented in [9] to predict and provide the right resources at anytime and anywhere. Moreover, it supports resource infinite computing. An expected limitation of PlanetCloud is providing hard QoS guarantees. PlanetCloud is anticipated to provide soft QoS guarantees based on resource prediction, collection, and stability of environment.

III. OVERVIEW OF PLANETCLOUD

A. *Preliminary PlanetCloud Architecture*

Our solution provides a novel resource calendaring mechanism deployed on each client. The client calendar is updated dynamically. Initial information about resource availability is collected from clients via server agents in the form of spatio-temporal calendars. User authentication is mandatory before accessing the PlanetCloud system. The user obtains the access credentials from the server agent she registered at before. Calendars and requested tasks are collected from mobile clients, as resource providers, and sent to distributed resource servers. UCCs might be formed to suit the needs of the requesters.

We propose a hierarchical model based on the concept of an agent as a fundamental building block in PlanetCloud. Agents manage tasks proactively without direct user interventions. In addition, agents maintain local data and state functionalities to distributed resource servers (RSs). A RS manages cloud calendars through a data repository storing user profiles, and spatio-temporal calendars of clients and clouds. The PlanetCloud architecture is depicted in Figure 1.

Our proposed spatio-temporal calendaring mechanism enables PlanetCloud to automatically adjust resources to each cloud. Resource allocation abides by the Service Level Agreement (SLA) when conditions or policies change due to user movement [11]. Heterogeneous clients can interact with PlanetCloud throughout the application interface of each client. There would be multiple interfaces for the calendaring mechanism on different devices.

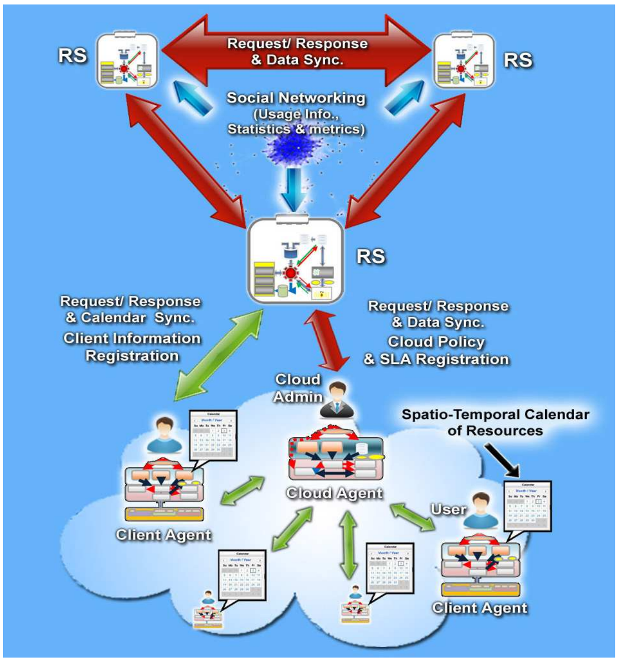

Figure 1. PlanetCloud Architecture.

There are two types of agents in PlanetCloud namely client agents and cloud agents. A single node could be a client agent, a cloud agent, or both.

1) Client Agent

The client agent application manages the client spatio-temporal resource calendar. It handles incoming requests for cloud formation, notifies a user of the next incoming clouds, connects with all other agents involved in the cloud formations, and synchronizes the calendar's content with the RSs data. The client agent consists of two units, the knowledge unit and management unit as shown in Figure 2. The knowledge unit consists of three subunits describing the agent knowledge. The client agent has it own spatio-temporal calendar, which includes spatial and temporal information of the involved resources. The ID-Card unit is used by agents to identify the unique agent ID (AID) as well as the access credentials. The client agent obtains settings about the scheduled/requested clouds and some priority defined parameters through the preferences & settings unit. The Management Unit has five components. The Policy Information Base (PIB) contains predefined or on the fly policies created by a cloud admin, which are used to regulate the operation of PlanetCloud components. The agent manages interactions to form a cloud and prevents inconsistency in the calendar by its controller unit, which provides both policy and client control functions. Agents synchronize their calendar with the data in RSs and cloud agent side through the synchronizer unit. The communicator unit provides the required communications for different activities such as cloud formation requests and responses. Moreover, it is involved in the synchronization process among calendars and PlanetCloud data repositories. The client agent monitors the internal state of the user resources using its observer unit. The lowest layer, of the client agent's building blocks, consists of the application, networking, and computing resources, which are involved in the delivery of the service.

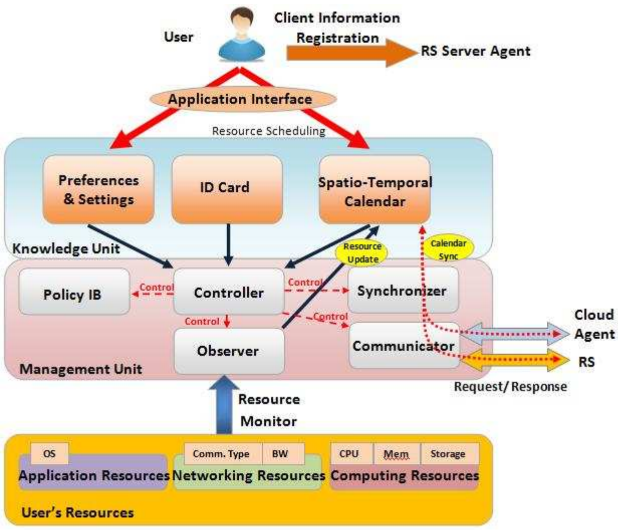

Figure 2. Client Agent Building Blocks

2) Cloud Agent

The cloud agent application is deployed on a high capability client to manage and store the data related to spatio-temporal calendars for all clients within a cloud. Figure 3 shows the building blocks of the cloud agent. The cloud agent uses a central data repository, as a part of its knowledge unit, to store the user profiles, and spatio-temporal calendars of clients within a cloud.

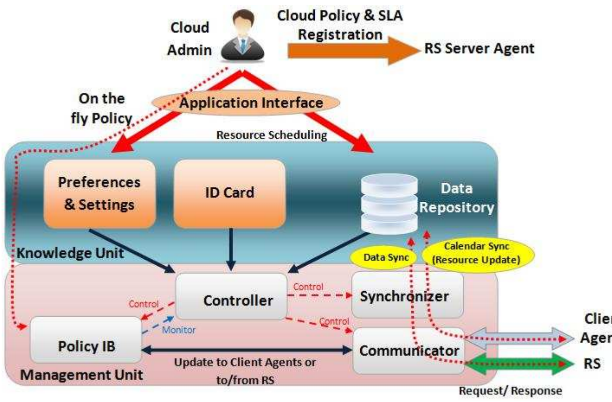

Figure 3. Cloud Agent Building Blocks.

In addition, it includes resource profiles and clients preferences. The cloud agent maintains the overall picture of the resource capability within the cloud. The cloud agent controller can query the data repositories in the RSs for extra resources. Further, the controller provides the necessary resources to the formed cloud based on the SLA information.

3) Resource Server (RS)

Distributed RSs operate on the updated data from clients' calendars and clouds data, which are stored in a data repository as shown in Figure 4. The RS provides resource forecasting using an implemented prediction unit. It uses calendaring data coupled with social network data analysis to increase the forecasting precision. We need to discover the association rules among calendaring data in the data repository. The association rules between different resource types are necessary to define their co-occurrence in each location at different periods of time. We use the calendaring data as following: Generate data files to be excavated; (2) Perform the mining job applying the fuzzy calendar association rules; and (3) Obtain association rules which is extracted from the data files.

We can identify four inputs to the prediction unit namely association rules as an input from calendaring mechanism, and three other inputs from social networks. The inputs from social networks are usage information, usage statistics, and graph type metrics calculated using graph methods. The RS has some other building blocks as follows:

a) Information Bases: It includes two other types, rather than PIB mentioned before namely User Information Base (UIB) and Cloud Information Base (CIB). UIB contains user information (personal, subscribed services, etc.). CIB contains information about the cloud formed by users, SLAs, types of resources needed, amount of each resource type needed, and billing plan for the service.

b) Account manager: It adjusts the bill of a user according to billing policies set by the Cloud Agent.

c) Synchronizer: Repository data are synchronized via updated calendar modifications to/from all clients, updated data modifications to/from all clouds or the other RSs.

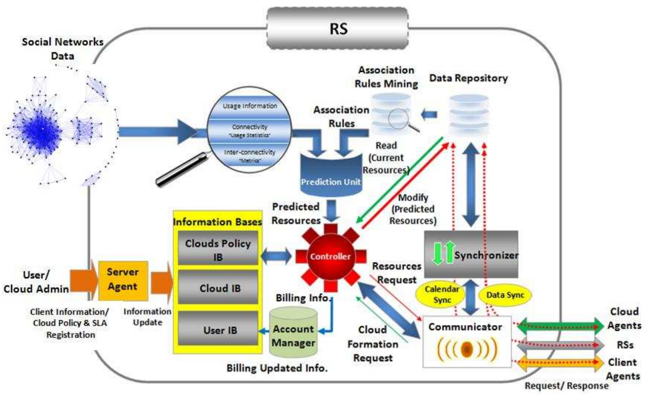

Figure 4. PlanetCloud Resource Server (RS) Building Blocks.

B. Calendar maintenance

We describe our algorithm to maintain the dynamics of the network. The cloud consists of static resources of fixed virtual nodes that meet cloud task requirements. Each virtual node is emulated by a subset of the real mobile nodes, participating in its tasks, while being monitored and tracked by PlanetCloud. Both virtual and mobile nodes have similar communication capabilities allowing them to communicate with one another. The subset locally stores the state of the emulated virtual node. The real nodes perform parts of the task assigned to their emulated virtual node. However, for an emulated virtual node, only one of the real nodes is responsible for aggregating the outputs of the subset, while maintaining their state consistency. If a mobile node fails or leaves the cloud, it ceases to emulate the virtual node; a mobile node that joins the cloud attempts to participate in the emulation.

IV. THE CALENDARING MECHANISM

1) Spatio-temporal resource calendar: We have employed a calendaring schema determined by a hierarchy of calendar units (month, day, hour, minute) [12]. We use a calendar pattern to determine the spatio-temporal resource availability. This pattern includes all time intervals in the corresponding time granularity. A cloud administrator has to specify a calendar pattern which describes his desired periodicities. Then, a query is issued to data repository to locate the proper resources.

Figure 5 shows an example of a spatio-temporal resource calendar. Data could be indexed by three main attributes: time, location, and resources, or combinations of them. Query is performed by any of the main attributes. For example, we can query the availability of certain resource type, or the resources available during a certain time period or at certain location.

Location		Time (Month, Day, Hour, Minute)		Computing resources			Communication resources		Application resources	Cloud ID
X	Y	Start time	End time	Processing	Storage	Memory	Comm. types	Band-width		

Figure 5. Spatio-Temporal Resource Calendar Example.

2) Calendaring Protocol: The cloud administrator requests some tasks through its cloud agent along with the cloud preferences. The cloud agent forwards the cloud formation request to a RS to locate the required resources according to the cloud settings and preferences. The RS looks up the data repository and determines the clients or clouds having the requested resources in terms of time and location domains. The process for scheduling the resources needed for cloud formation is illustrated in Figure 6.

3) Calendaring Application: A software application for calendaring is implemented, where a user can access their spatio-temporal calendars, modify preferences, reschedule his resource availability, and locate resources at anytime and anywhere. In addition, this software enables a user to request an on-demand or hybrid cloud formation,

 ISBN: 978-1-61208-154-0

synchronize the calendar with the PlanetCloud RS data, and be notified about the new requests for his resources.

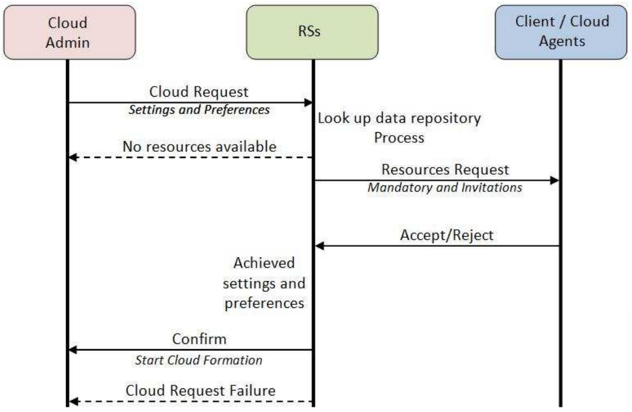

Figure 6. The Cloud Formation Process

V. PERFORMANCE EVALUATION: A DISCUSSION

Our ongoing work focuses on realizing and evaluating the PlanetCloud architecture through a prototype and simulation. The PlanetCloud preliminary prototype spans a specific region by constructing clouds of nodes connected in an ad-hoc manner. We implement an interface to deal with resources of heterogeneous mobile nodes, e.g. mobile devices or laptops. The evaluation will be performed by measuring the query response time, which is the time it takes to answer a query and locate the required resources, and the cloud formation time. In addition, we evaluate the cloud availability in terms of continuity of service in case of failure. We measure the cloud availability by the mean time to recovery & repair from failures. On the other hand, the model for resource forecasting using spatio-temporal calendaring analysis coupled with social network analysis is evaluated by its prediction accuracy, which is measured by the prediction error. Therefore, we measure the difference between the resources correctly predicted and the actual set of resources that are discovered for a given time and location [13]. The parameters for performance evaluation include: number and mobility of nodes as services, calendar size and distribution, and number and type of resources.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed the concept of PlanetCloud, which enables ubiquitous computing clouds through using a new dynamic spatio-temporal calendaring mechanism, seamless provisioning of trustworthy resources and socially-intelligent resource forecasting. Our ongoing research comprises the following tasks:

(1) Develop methodology to enhance the analysis performance of complex graphs and clouds with vast quantity of dynamic data from both social networking and spatio-temporal calendars to provide efficient resource discovery, assignment and forecasting. Enhancement is achieved by providing efficient data structure to calculate all association rules among different resource types to find the interesting resource pattern.

(2) Transparently maintain applications' QoS by providing an efficient mechanism to maintain local data and state functionalities in a highly dynamic environment. This mechanism provides virtual static resources that meet cloud task requirements, while being emulated by subsets of the real mobile nodes, which maintain their state consistency.

(3) Provide a solution to achieve privacy preservation when sharing resources and data for calendaring. There is a need to support distributed data access and computations without compromising the raw data of cloud nodes.

REFERENCES

[1] M. Eltoweissy, S. Olariu, and M. Younis, "Towards autonomous vehicular clouds," in Ad Hoc Networks, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2010, vol. 49, pp. 1–16.

[2] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," IEEE Network, vol. 24, no. 4, pp. 19 –24, 2010.

[3] B. Kouchner, "Globalization and new world order: are we ready for "scientists without borders"?" IEEE Transactions on Applied Superconductivity, vol. 15, pp. 1071 – 1077, 2005.

[4] J. Londono and A. Bestavros, "Netembed: A network resource mapping service for distributed applications," IEEE International Symposium on In Parallel and Distributed Processing, 2008. IPDPS 2008., 2008, pp. 1 –8.

[5] Y. Shavitt and N. Zilberman, "A structural approach for pop geo-location," in INFOCOM IEEE Conference on Computer Communications Workshops , 2010, 2010, pp. 1 –6.

[6] R. Ranjan, A. Harwood, and R. Buyya, "Peer-to-peer-based resource discovery in global grids: a tutorial," IEEE Communications Surveys Tutorials, vol. 10, no. 2, pp. 6 –33, 2008.

[7] D. Werthimer, J. Cobb, M. Lebofsky, D. Anderson, and E. Korpela, "Seti@homemassively distributed computing for seti," Computing in Science and Eng., vol. 3, pp. 78–83, January 2001.

[8] D. Ediger, K. Jiang, J. Riedy, and D. Bader, "Massive streaming data analytics: A case study with clustering coefficients," 2010 IEEE International Symposium on In Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010, pp. 1 –8.

[9] D. Ediger, K. Jiang, J. Riedy, D. Bader, C. Corley, R. Farber, and W. Reynolds, "Massive social network analysis: Mining twitter for social good," 39th International Conference on In Parallel Processing (ICPP), 2010, pp. 583 –593.

[10] T. Menard, and J. Miller, "FreeSim_Mobile: A novel approach to real-time traffic gathering using the apple iPhone," IEEE in Vehicular Networking Conference (VNC), 2010, pp. 57 -63, Dec. 2010.

[11] Y. Cheng, R. Farha, M. Kim, A. Leon-Garcia, and J. W.- K. Hong, "A generic architecture for autonomic service and network management," Computer Communications, no. 29, pp. 3691–3709, 2006.

[12] J.-Y. Jiang, W.-J. Lee, and S.-J. Lee, "Mining calendar-based asynchronous periodical association rules with fuzzy calendar constraints," in 14th IEEE International Conference on Fuzzy Systems, 2005, pp. 773–778.

[13] S. Lili, Y. Shoubao, G. Liangmin, and W. Bin, "A Markov chain based resource prediction in computational grid," in Fourth International Conference on Frontier of Computer Science and Technology (FCST), pp. 119 –124.