Smart Distribution of Bio-Signal Processing Tasks in M-health

Hailiang Mei

Department of Computer Science, University of Twente, The Netherlands

H.Mei@utwente.nl

Abstract. The past few years have witnessed a rapid advancement of mobile healthcare systems. However, in the mobile computing environment, the resource fluctuations, stringent application requirements and user mobility have severely hindered the performance and reliability of the healthcare service delivery. The current approaches to solve this resource supply and service demand mismatch problem either limit the adaptation to an isolated node or require significant user's involvement. Given the distributed processing paradigm of m-health system, we propose that a new adaptation approach could be dynamically redistributing processing tasks across distributed nodes. This PhD research addresses two main issues to validate this approach: (1) computation of a suitable assignment of tasks at compile-time or run-time; and (2) dynamic distribution of tasks across the nodes according to this new assignment at run-time.

1 Introduction

Telemedicine has been receiving more and more attentions due to its ability to tackle the aging society problem, improve the quality of diagnosis and treatment and reduce the medical cost [1]. Being part of telemedicine, mobile healthcare (m-health) emerges with the fast adoption of advanced mobile technology into our daily life. Several m-health systems have been developed for mobile network environments [2-5], in which an m-health platform is introduced including a patient body-area network and some back-end healthcare service facilities. On top of the platform, multiple tele-monitoring applications can be operated to provide long term medical services to patients.

However, like other applications running in a mobile environment, the usability of m-health applications are seriously affected by the fluctuations of context and scarcity of the environment's resources, e.g. network bandwidth, battery power and computational power of handhelds, etc [5, 6]. From a technical point of view, to solve this application demand and resource mismatch, an appropriate adaptation mechanism can be embedded into the system. There are three approaches on building this adaptation mechanism [7]: (1) adjusting task - this is to automatically change task behavior to use less of a scarce resource, e.g. scalable video transmission over wireless network; (2) reserving resource - this is to ask the environment to guarantee a certain level of a resource, e.g. QoS management and reservation techniques; (3) informing user - this is to suggest a corrective action to the user. The second approach assumes that it is possible to reserve sufficient resources for the task, which is sometimes unrealistic, e.g. the drop of network bandwidth could be so significant that the required data transmission quality just cannot be met. The third approach could avoid the mismatch by giving the patient suggestions or warnings, e.g. "Please be near to a charging point to reduce the risk of draining battery power". However, it may not be a pleasant experience for users by limiting his activity. Therefore, we focus on the "adjusting task demand" approach to tackle the mismatch problem in m-health.

Traditionally, adjusting task is often performed within an isolated node, e.g. by a local application specific adaptor [8]. The applied methods include data compression, discarding less important information, handover to a better network, etc. The fundamental model common to the m-health systems consists of a set of bio-signal data processing tasks distributed across a set of networked nodes. Therefore, a possible adaptation scenario is to actually benefit from the distributed processing paradigm and adjust the assignment of tasks across available nodes at run-time. The rationale is that if one node cannot support its hosted task on computation or data communication, some other nodes with richer resources can take over some of the tasks. The advantage over traditional methods is that the user requirement would be less compromised and distributed resources could be better utilized.

The aforementioned analysis leads to the following hypothesis of this Ph.D. research: "To overcome the difficulty caused by context changes, the m-health system can adapt itself to a new configuration that application demand can be satisfied with the underlying resources by dynamically redistributing processing tasks across available distributed nodes." Following this hypothesis, this PhD research addresses two major topics: (1) computation of a good assignment of tasks in compile-time or run-time and (2) dynamic distribution of tasks across the nodes according to this new assignment in run-time.

2 Research Background and Problem Formulation

In this section, we will describe the research background of context-aware m-health system and propose a list of research questions.

2.1 Context-aware m-health system

An m-health platform under study [9] consists of the following nodes in the mobile and fixed networks: one or more body-worn sensorboxes, a handheld Mobile Base Unit (MBU), a back-end server and an end-terminal (Fig. 1). The sensorboxes collect biosignals of the patient or other data like location or activity and send the data to the MBU over wireless short-range connections, e.g. a Bluetooth connection. As a gateway, the MBU sends the data to the back-end server of the healthcare portal over a WiFi, GPRS or UMTS connection. Thereafter, the data can be streamed to or accessed by a healthcare professional using his end-terminal, e.g. his laptop or desktop PC.

This platform can support different tele-monitoring applications. One example is the epilepsy prediction application developed in [10]. It consists of six biosignal processing units/tasks (BSPU) as shown in Fig. 2. First, the patient's raw ECG data is filtered to remove signal artifacts and environment noise. Further, beat-to-beat heart rate (HR) is detected and heart rate variability (HRV) in the frequency domain is calculated. The frequency spectrum of HRV is used to calculate the probability of an occurring seizure. To reduce the chance of false alarm, the patient's activity information is combined in the final stage.

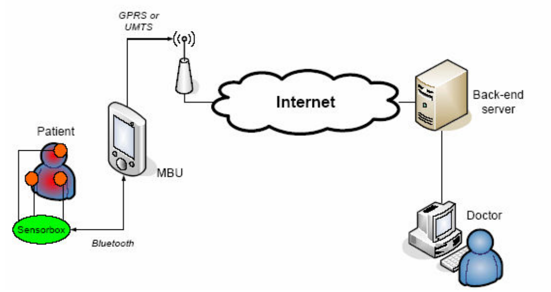

Fig. 1. Based on M-health platform, the (remote) healthcare professional can take appropriate treatment actions for the patient upon processed biosignals

One important initiative of introducing m-health system is to support patient with better daily social life. Therefore, m-health system should be capable of handling those frequent environment changes in every day life. One possible context-aware scenario targeted by this PhD work is described below:

*Sandra suffers epileptic seizures and she wears a set of m-health platform devices. The aim of using this system is to provide Sandra with higher levels of safety and independence in order to function more normally in society despite her disease. When Sandra is at home, a broadband network is available to transfer her raw ECG and activity information to the remote monitoring centre, e.g. the back-end server in (*Fig. 1*). In this case, all six BSPUs are deployed onto the back-end server and the doctor can be warned if a seizure is likely to occur. One afternoon, she runs her usual circuit in the forest. Since there is no broadband network available in the forest, the bio-signals of Sandra cannot be sent due to insufficient network bandwidth. Therefore, all six BSPUs are loaded into her MBU and her bio-signals are processed locally. During her running, a possible imminent epileptic seizure is detected and, very likely, is going to occur within a few minutes. Sandra is immediately warned, and stops running. At the same time, an alarm and the position of Sandra are sent to the monitoring centre via a narrow band connection, e.g. GSM. Therefore, a healthcare team can be sent to the exact location of Sandra and provide her medical help.*

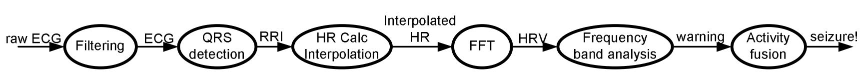

Fig. 2. Epilepsy prediction application consisting of six units (tasks)

2.2 System Model and Research Questions

The m-health system introduced in the previous section reveals two parts: a set of BSPUs, which represents the tele-monitoring application, and a set of m-health platform nodes. In general, BSPUs can be modeled as a *process DAG* (Directed Acyclic Graph), $G_P = (V_P, E_P)$, where V_P represents the set of BSPUs and E_P represents the set of precedence relation between BSPUs, while platform nodes can be modeled as a *node DAG*, $G_N = (V_N, E_N)$, where V_N denotes the set of available nodes and E_N denotes the set of (unidirectional) communication links between the nodes. The pre-described m-health platform (Fig. 1) and epilepsy prediction application (Fig. 2) exhibit as two chains, which are specializations of this general graph model.

In the process DAG, we could further weight the vertex and arc with the information about demand on computation (D_P) and communication (D_C) respectively. In the node DAG, we could weight the vertex and arc with the information about resource on computation (R_P) and communication (R_C) respectively. Thus, these two graphs are capable of representing various tele-monitoring applications and different m-health platforms. In particular, the fluctuation of node's resource can be denoted as the change of R_P and R_C and the dynamism of nodes' presence can be denoted as the topology change of node DAG.

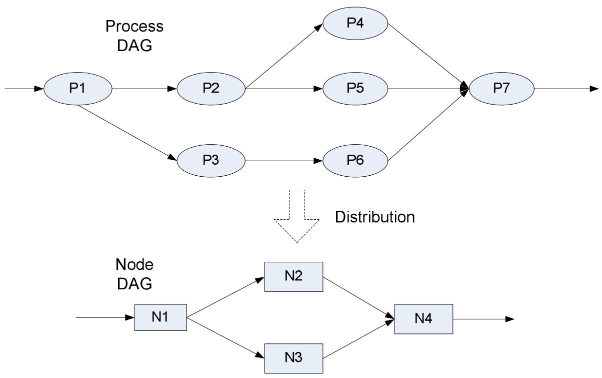

Fig. 3. Placement of seven BSPUs in a process DAG onto four platform nodes in a node DAG

Based on this m-health system graph model, the problem of BSPU distribution can be seen as a "mapping" between two graphs (Fig. 3). In order to validate our hypothesis, we ought to give answers to two research questions: (1) How to determine a suitable assignment of a process DAG onto a node DAG and (2) How to reconfigure the BSPUs across the nodes according to this suitable assignment with minimum disruption. These two central research questions yield the following sub-questions:

1. What does "a suitable assignment" mean in m-health system?
2. What is the performance gain by adapting the system towards a more suitable assignment?
3. How to design an adequate assignment algorithm for m-health system?
4. Can this algorithm provide a suitable assignment on-the-fly?

5. What are the requirement of BSPU distribution and how to support it in m-health system?
6. How to measure the reconfiguration cost of BSPU distribution and can this cost be taken into consideration of the assignment algorithm?
7. Can real m-health system benefit from the capability of distributing BSPU dynamically?

3 Research Approach

Our research roadmap consists of three consecutive phases that can be repeated cyclically (Fig. 4). In phase 1, we studied the actual problem and related concepts to propose the hypothesis and research question. In addition, we propose a general approach to evaluate the performance of m-health system given different assignments. This requires formulating a set of metrics (i.e. end-to-end delay, battery time) for quantitative measuring the m-health system. Sub-question 1 and 2 should be answered once this phase is completed.

In phase 2, we are building the artifacts to validate the hypothesis. We propose an overall approach for the design of context-aware dynamic BSPU distribution framework. It consists of a decision making component that can host the assignment algorithm and a software infrastructure to support the BSPU distribution. Since the assignment algorithm might be executed in run-time, special attention needs to be paid to the time and space complexity during the design of algorithm. Furthermore, we could measure the reconfiguration overhead based on the performance metric. This reconfiguration cost should be fed into the decision making component to evaluate the real gain of BSPU distribution. Sub-question 3, 5 and 6 will be tackled in this phase.

In phase 3, we will analytically and experimentally evaluate the developed system to validate the hypothesis. In particular, the disruption to the m-health system caused by reconfiguration will be analyzed. Sub-question 4 and 7 should be answered here.

By the end of 2nd PhD year, phase 1 is largely finished and research on phase 2 is in progress. In the next two sections, we describe some intermediary results obtained in phase 2 on optimal assignment algorithm and distribution framework respectively.

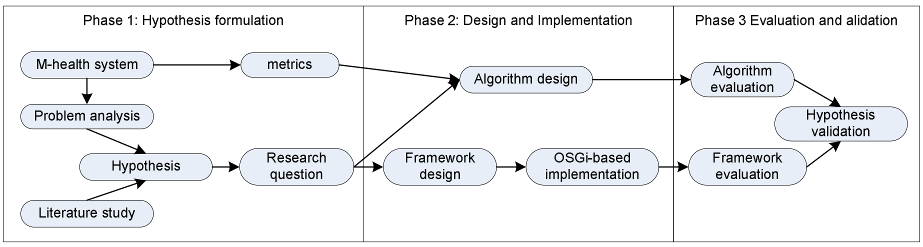

Fig. 4. PhD research roadmap

4 Assignment Algorithm

Based on the predefined m-health system model in Section 2.2, the objective of the assignment algorithm is to find an assignment function $\Psi_\alpha: V_P \rightarrow V_N$ among all possible assignments such that $\tau(\Psi_\alpha)$ is optimized, where τ is a performance measure for a particular assignment of BSPUs across nodes. This measure could be formed to denote the end-to-end process and transfer delay, system (battery) lifetime or a composite utility function.

There are two constraints this algorithm has to follow: *local constraint* and *ordering constraint*. The local constraint is that for every BSPU $P_i \in V_P$, the hosted node $\Psi_\alpha(P_i)$ should fulfill P_i's system requirement, e.g. minimal CPU speed, library support, etc. This local constraint reflects the heterogeneity in nodes' resource and BSPUs' demand. The ordering constraint is that for every BSPU $P_i \in V_P$, all the direct successor BSPUs of P_i have to be assigned to the node $\Psi_\alpha(v_i)$ or the *successor* nodes of $\Psi_\alpha(v_i)$.

The task assignment problem in its general form is NP-hard [11]. Therefore, we investigate into branch and bound method [12] and various heuristic approaches to tackle this problem [13]. We also plan to investigate into full-space search methods and propose a common programming model for m-health. Since the number of BSPUs and nodes in real system may not be that large, this exhaustive approach could be capable of handling the problem. However, some cases with special topological feature might have more efficient solutions. We have proposed polynomial-time algorithms for chain to chain assignment [14] and tree to star assignment [15] inspired by previous work [16].

5 BSPU distribution framework

We envision BSPUs as software components whose execution location may be distributed. A high-level architecture of our proposed BSPU distribution framework consists of a Coordinator, a BSPU Repository and a set of Facilitators (Fig. 5). Based on the observations of latest context, the Coordinator that hosts the assignment algorithm can make decisions to reconfigure the whole m-health system, e.g. redistribute some BSPUs across the network nodes. The reconfiguration can be either stateful or stateless depending on application specific requirements. When stateful transition is required, the Coordinator should identify a plan that the running states are properly transferred between BSPUs. Furthermore, there is no restriction of having only one coordinator: multiple coordinators might exist in the network with different start levels, to avoid the risk of single point failure.

Each node has a Facilitator to manage and facilitate the resident BSPUs. A Facilitator is the representative of its node in the distribution framework: it can report the presence/absence of the node, it can receive the control commands on BSPU life-cycle management from the Coordinator and it can report errors on distribution, back to the Coordinator. A Facilitator can also receive as input bio-signal data packets, feed the decoded bio-signal data into one of its managed BSPUs, encode the output bio-signal data from a BSPU into packets and send these packets to the next Facilitator on the path. When stateful transition is required, Facilitator should be able to intercept the resident BSPU and log its running state. Once a reconfiguration is initialized by the Coordinator, the state information should be transferred and fetch into the targeted BSPU by the corresponding Facilitator.

The BSPU Repository stores BSPU implementations that can be provided on request. If the BSPUs are preloaded onto each node, the BSPU repository becomes a distributed component, which eliminates the overhead of downloading new BSPU at runtime.

A prototype, based on the OSGi technology, has recently been built to validate the design of the proposed framework. By proposing the performance metrics on this reconfiguration framework, the interrupt to the m-health application can be measured and analyzed.

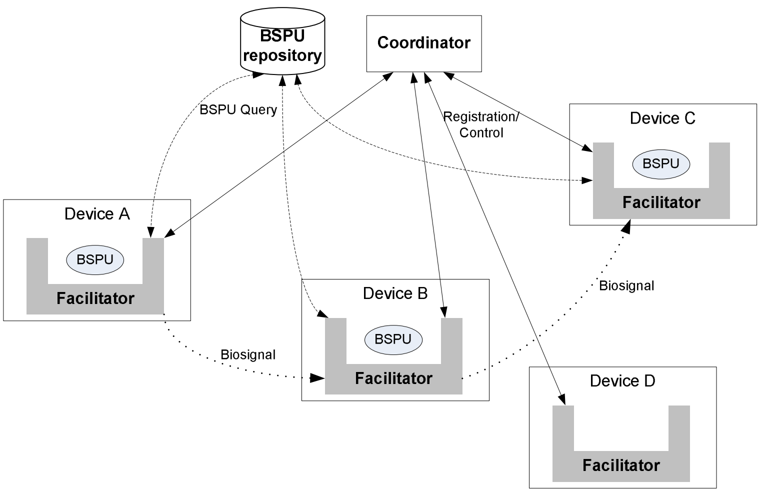

Fig. 5. Architectural components of BSPU distribution framework

6 Related Work

The task assignment problem (also referred as allocation, mapping or partitioning) has been studied for many years with various models. It is an essential research topic in the field of parallel computing [11], grid computing [17], and more recently, in the field of distributed database [18]. In [19], Stone proposed the optimal task assignment algorithm on two-processor problem by using Max-flow/Min-cut theorem. This work laid foundation for later heuristic [12, 20] and exact methods [21, 22]. In [20], Stone's procedure was applied recursively in multiple-processor network with consideration of interference cost. In [12], Stone's procedure was used to cluster the tasks as a pre-processing step for the proposed branch and bound method. A* algorithm was first proposed in [21] to tackled the unsolved problem from Stone and were further enhanced in [23, 24]. In [22], the assignment to a general array or tree network containing homogeneous processors are studied and polynomial-time exact algorithm is proposed. Another important discovery was led by Bokhari [16], in which several models with special topology are studied and graph-based efficient algorithms are proposed. Several improved algorithms on these topologies were further reported in [14, 25-27].

Most of these reported algorithms use task completion time or data throughput as the objective function since the problems are originated from parallel system in a fixed network. To m-health system, some other objective functions are more important and are addressed in this PhD work, e.g. maximizing the battery lifetime. Furthermore, in the m-health system model, both process graph and node graph have directions. The resulting ordering constraint can reduce the assignment search space but could complicate the design on assignment algorithm.

The core of BSPU distribution framework is a dynamic reconfiguration mechanism [28]. Dynamic reconfiguration is a technique to improve the service availability since it can update a running system without taking it off-line [29]. This technique has been applied to deal with the changing of user requirements [30] or underlying resources [31] in distributed streaming applications, which is closely related to m-health system. Three correctness requirements on dynamic reconfiguration were derived in [32]: the system satisfies its structural integrity requirements; the entities in the system are in mutually consistent states; and the application state invariants hold.

The development on component-based software engineering can support efficient implementation of dynamic reconfiguration system, e.g. CORBA [33], enterprise JavaBean [34]. The newly emerged OSGi technology[1] provides a service-oriented, component-based environment and offers standardized ways to manage the component lifecycle. Since the main targets are embedded and mobile systems, it is also lightweight and thus a candidate for implementation of BSPU distribution framework.

7 Conclusion

This paper presents the on-going PhD research on dynamic distribution of biosignal processing tasks across available nodes in m-health system. The hypothesis is that the m-health system can therefore adapt in such a way to overcome the demand and resource mismatch caused by patient's mobility or stringent resources. There are two main research topics: (1) computation of a suitable assignment of tasks in compile-time or run-time and (2) dynamic distribution of tasks across the nodes according to this new assignment in run-time. We outlined the research approach and presented some intermediate results.

Since the derived system model of m-health system also applies to other distributed stream processing applications, e.g in [18, 30, 31, 35], the research results obtained in this PhD work are expected to benefit those application areas as well.

[1] www.osgi.org

Acknowledgement

This work is part of the Freeband AWARENESS Project. Freeband is sponsored by the Dutch government under contract BSIK 03025. (http://awareness.freeband.nl). The author would like to thank Prof. Hermie Hermens, Prof. Boudewijn Haverkort, Dr. Bert-Jan van Beijnum and Dr. Ing Widya for their effort on guiding this PhD work.

References

1. TelemedicineAlliance. Telemedicine 2010: Visions for a Personal Medical Network. 2004 [cited; Available from: http://www.esa.int/esapub/br/br229/br229.pdf.
2. Hung, K. and Z. Yuan-Ting, Implementation of a WAP-based telemedicine system for patient monitoring. Information Technology in Biomedicine, IEEE Transactions on, 2003. 7(2): p. 101.
3. Yuan-Hsiang, L., et al., A wireless PDA-based physiological monitoring system for patient transport. Information Technology in Biomedicine, IEEE Transactions on, 2004. 8(4): p. 439.
4. Rasid, M.F.A. and B. Woodward, Bluetooth telemedicine Processor for multichannel biomedical signal transmission via mobile cellular networks. Information Technology in Biomedicine, IEEE Transactions on, 2005. 9(1): p. 35.
5. Halteren, A.v., et al., Mobile Patient Monitoring: The MobiHealth System. The Journal of Information Technology in Healthcare, 2004. 2(5).
6. Jones, V., et al., Future challenges and recommendations, in M-health Emerging Mobile Health Systems, R.S.H. Istepanian, S. Laxminarayan, and C.S. Pattichis, Editors. 2006, Springer. p. 267-270.
7. Satyanarayanan, M., Pervasive computing: vision and challenges. IEEE Personal Communications, 2001. 8(4): p. 10-17.
8. Badrinath, B., et al., A Conceptual Framework for Network and Client Adaptation. Mobile Networks and Applications (MONET), 2000.
9. AWARENESS, Freeband AWARENESS project webpage. 2005, http://www.freeband.nl/project.cfm?id=494&language=en.
10. Tönis, T., H.J. Hermens, and M. Vollenbroek-Hutten, Context aware algorithm for discriminating stress and physical activity versus epilepsy. 2006, AWARENESS deliverables (D4.18).
11. Norman, M.G. and P. Thanisch, Models of machines and computation for mapping in multicomputers. ACM Computing Surveys, 1993. 25(3): p. 263-302.
12. Ma, Y.-C., T.-F. Chen, and C.-P. Chung, Branch-and-bound task allocation with task clustering-based pruning. Journal of Parallel and Distributed Computing, 2004. 64(11): p. 1223 - 1240.
13. Pawar, P., et al. Context-Aware Task Assignment in M-health - A Genetic Algorithm Based Approach. in 2007 IEEE Congress on Evolutionary Computation (CEC 2007). 2007. Singapore.
14. Mei, H. and I. Widya. Context-Aware Optimal Assignment of a Chain-like Processing Task onto Chain-like Resources in M-Health. in 7th International conference on computational science. 2007. Beijing, China.

15. Mei, H., P. Pawar, and I. Widya. Optimal Assignment of a Tree-Structured Context Reasoning Procedure onto a Host-Satellites System. in 16th Heterogeneity in Computing Workshop (HCW 2007). 2007. Long Beach, California.
16. Bokhari, S.H., Partitioning problems in parallel, pipelined, and distributed computing. IEEE Transactions on Computers, 1988. 37(1): p. 48-57.
17. Cooper, K., et al. New grid scheduling and rescheduling methods in the GrADS project. in Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International 2004.
18. Pietzuch, P., et al. Network-Aware Operator Placement for Stream-Processing Systems. in 22nd International Conference on Data Engineering (ICDE'06). 2006.
19. Stone, H.S., Multiprocessor scheduling with the aid of network flow algorithms. IEEE Transactions on Software Engineering, 1977. 3: p. 85-93.
20. Lo, V.M., Heuristic algorithms for task assignment in distributed systems. Computers, IEEE Transactions on, 1988. 37(11): p. 1384.
21. Shen, C.-C. and W.-H. Tsai, A Graph Matching Approach to Optimal Task Assignment in Distributed Computing Systems Using a Minimax Criterion. Transactions on Computers, 1985. C-34(3): p. 197-203.
22. Lee, C.-H. and K.G. Shin, Optimal task assignment in homogeneous networks. Parallel and Distributed Systems, IEEE Transactions on, 1997. 8(2): p. 119-129.
23. Kafil, M. and I. Ahmad, Optimal task assignment in heterogeneous distributed computing systems. Concurrency, IEEE [see also IEEE Parallel & Distributed Technology], 1998. 6(3): p. 42.
24. Ajith, T.P. and C.S.R. Murthy, Optimal task allocation in distributed systems by graph matching and state space search. Journal of Systems and Software, 1999. 46(1).
25. Sheu, J.-P. and Z.-F. Chiang, Efficient allocation of chain-like task on chain-like network computers. Information Processing Letters, 1990. 36(5): p. 241 - 246.
26. Ashraf Iqbal, M. and S.H. Bokhari, Efficient algorithms for a class of partitioning problems. Parallel and Distributed Systems, IEEE Transactions on, 1995. 6(2): p. 170.
27. Woeginger, G.J. Assigning chain-like tasks to a chain-like network. in Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms. 2001.
28. Kramer, J. and J. Magee, Dynamic Configuration for Distributed Systems. IEEE Transactions on Software Engineering, 1985. 11(4).
29. Wegdam, M., Dynamic Reconfiguration and Load Distribution in Component Middleware. 2003, University Twente, The Netherlands.
30. Gu, X. and K. Nahrstedt, On Composing Stream Applications in Peer-to-Peer Environments. Parallel and Distributed Systems, IEEE Transactions on, 2006. 17(8): p. 824-837.
31. Amini, L., et al. Adaptive Control of Extreme-scale Stream Processing Systems. in 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06). 2006.
32. Moazami-Goudarzi, K., Consistency preserving dynamic reconfiguration of distributed systems, in Imperial College London. 1999.
33. Almeida, J.P.A., et al. Transparent dynamic reconfiguration for CORBA. in 3rd International Symposium on Distributed Objects and Applications (DOA '01). 2001.
34. Rutherford, M.J., et al. Reconfiguration in the Enterprise JavaBean component model. in IFIP/ACM Working Conference on Component Deployment. 2002.
35. Xing, Y., et al. Providing Resiliency to Load Variations in Distributed Stream Processing. in 32nd International Conference on Very Large Data Bases. 2006.