Context-Aware Optimal Assignment of a Chain-Like Processing Task onto Chain-Like Resources in M-Health

Hailiang Mei and Ing Widya

Architecture and Services for Networked Applications Group, Department of Computer
Science, University of Twente, The Netherlands
{H.Mei, I.Widya}@utwente.nl

Abstract. This paper focuses on the optimal assignment of a chain-structured medical task onto a chain of networked devices, a need identified in context-aware mobile healthcare applications. We propose a graph-based method to compute the assignment which has the minimal end-to-end process and transfer delay. In essence, the method transforms the assignment problem to a shortest path problem in a graph representing all possible assignments. Compared to earlier work, our method relaxes a so-called contiguity constraint, which is a necessity in the earlier case but unnecessarily constraining in the healthcare applications. The proposed method reduces the time and space complexity in case it is adapted to include the contiguity constraint.

Keywords: graph-based, task assignment, contiguity constraint, M-health.

1 Introduction

In recent years, mobile healthcare (M-health) applications have received more and more attentions due to its ability to satisfy today's society demands, like patient self-management, continuous and anywhere tele-monitoring and tele-treatment ([1-3]). In M-health, sensed biosignals of mobile patients need to be processed and transferred to healthcare professionals via several handheld or body-worn devices. On the other hand, a biosignal processing task often consists of cascaded processing units, which have to be configured onto the previously mentioned devices. A challenge in M-health is therefore to study the dynamic assignments of the processing units to devices such that the performance of the processing task can be guaranteed despite the resource changes or fluctuations in the mobile environment. In this paper, we address the optimal assignment to minimize the *end-to-end process and transfer delay*. This measure suits our intended M-health applications which require timely responses.

This assignment problem is similar to an industrial case in the area of parallel computing studied earlier by Bokhari ([4]). In order to find the optimal assignment of m task units connected in a chain onto n processors also connected in a chain, Bokhari presented a graph-based solution using a layered assignment graph to find an optimal assignment with minimal *bottleneck processing time*. In this assignment graph, any path connecting two distinguished nodes represents a possible assignment. Therefore, the assignment problem is translated into a path-search problem which can be solved

with time complexity of $O(m^3n)$ and space complexity of $O(m^2n)$ ([4]). Some improved algorithms are further reported in [5-7].

Since the objective is to minimize the bottleneck processing time, the previous solutions have to obey a so-called *contiguity constraint*, i.e. any device should be assigned with at least one task unit. However, this constraint unnecessarily rules out several potential solutions for the M-health applications. In this paper, we propose a more general graph-based method to compute the optimal assignment, where it is possible that some devices are not assigned with a processing unit but only forward the processed biosignal data from the precedent device to the next device, for example to save battery power or to avoid complex processing on the particular device. With a slight modification, the proposed method is also applicable to problems with contiguity constraint while reducing the time and space complexity by a factor of m^2 and m respectively compared to Bokhari's original approach.

This paper is organized as follows: Section 2 describes a context-aware M-health application scenario for predicting epileptic seizures. Section 3 gives the problem formulation. Section 4 presents the graph-based method on finding the optimal assignment. Section 5 illustrates the result of performance measurement. Section 6 concludes the work.

2 Background

An epilepsy prediction task developed in ([8]) consists of six biosignal processing units (BSPU) as shown in Fig. 1. First, the patient's raw Electro Cardio Gram (ECG) data is filtered to remove signal artifacts and environment noise. Thereafter, beat-to-beat heart rate (HR) is derived and heart rate variability (HRV) in the frequency domain is calculated. The frequency spectrum of the HRV is then used to calculate the probability of an upcoming or occurring seizure. To reduce the chance of false alarms, the patient's activity information is monitored as well and correlated with the analyzed spectrum in the final stage.

Fig. 1. Epilepsy prediction task consisting of six BSPUs

The studied M-health platform ([9]), which facilitates the sensing, processing and transporting of the biosignals, consists of the following devices: a set of body-worn sensorboxes, a handheld Mobile Base Unit (MBU), a back-end server and an end-terminal (Fig. 2). The sensorboxes collect the biosignals of the patient, sense 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 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.

Like other applications running in a mobile environment, the usability of M-health applications are seriously affected by the fluctuations and scarcity of the environment's resources, e.g. network bandwidth, battery power and computational power of handhelds, etc. In order to deliver dependable healthcare services, M-health applications should embed context-aware features. For example, if a patient is outdoor and GPRS performance severely suffers from the GSM calls in the cell, transfer of raw ECG data to the back-end server may be impeded unacceptably if seizure forecasting or detection is completely processed at the server. Dynamically reassigning the processing units to the available devices may solve the problem: If the QRS detection can be assigned to the MBU, the GPRS channel bandwidth may be sufficient for smooth transfer of the derived heart rate signal, the patient's activity data and optionally one of the adequately downsampled ECG signal to conform to medical protocols. To realize this desired adaptation, two major topics should be addressed: computation of a better assignment of BSPUs and distribution of BSPUs across the devices according to this new assignment. We focus on the first topic in this paper and formulate the research problem in the next section.

Fig. 2. M-health platform

3 Problem Formulation

We formulate the problem of optimally assigning a BSPU chain onto a device chain as:

- **Given:** a directed BSPU chain containing m BSPUs increasingly indexed by $i \in \{1, 2, \dots, m\}$, a directed device chain containing n devices increasingly indexed by $j \in \{1, 2, \dots, n\}$ and a performance measure τ , where τ denotes the end-to-end process and transfer delay for a particular configuration of BSPUs across devices.
- **Goal:** To find an assignment function Ψ_{opt} among all possible $\Psi: \{i\} \rightarrow \{j\}$ such that $\tau(\Psi_{opt})$ is the minimal one among all $\tau(\Psi)$.
- **Subject to:** the following *ordering-* and *local-constraints*.

The *ordering constraint* is that for every pair of BSPU i and BSPU i' , if $i < i'$, then $\Psi(i) \leq \Psi(i')$. This constraint therefore reflects the consistency of directions of the two chains. It is a more relaxed constraint compared to the *contiguity constraint* discussed earlier (cf. [5-7]). The *local constraint* is that for every BSPU i , its hosting device $\Psi(i)$

should satisfy the BSPU i 's system requirement (e.g. CPU speed and library support) and user preference (e.g. this BSPU i has to run on device $\Psi(i)$).

We define the computation time for BSPU i to process one frame of biosignal data at device j as $p_{i,j}$ and the communication time for transferring one frame of BSPU i 's output biosignal data over the link between device j and $j+1$ as $c_{i,j}$. The variable $c_{0,j}$ denotes the communication time for transferring one frame of raw sensor output over the link between device j and $j+1$. We neglect intra-device communication time, e.g. if both BSPU i and $i+1$ are located on the same device. We further assume there is no additional computation time overhead for executing several BSPUs on the same device. Since the values of $p_{i,j}$ and $c_{i,j}$ could be known a priori, e.g. based on analytical benchmarking or task profiling ([10]), the end-to-end delay τ of a particular configuration of BSPUs across devices can be calculated. In Fig. 3, we illustrate a 6-BSPU-4-device assignment of the epilepsy prediction task on the networked M-health devices. In this example, BSPU 1 and 2 are assigned to device 1, BSPU 3, 4 and 5 are assigned to device 3, etc. Based on our definitions, its end-to-end process and transfer delay can be calculated as:

$$\tau = p_{1,1} + p_{2,1} + p_{3,3} + p_{4,3} + p_{5,3} + p_{6,4} + c_{2,1} + c_{2,2} + c_{5,3}$$

Fig. 3. An example of assigning a directed BSPU chain ($m=6$) onto a directed device-chain ($n=4$)

4 Optimal Assignment Computation Procedure

In order to find the optimal assignment of m BSPU (indexed by i) onto n devices (indexed by j), we first build a layered assignment graph consisting $mn+2$ nodes (Fig. 4). In this graph, each row (excluding nodes $\langle S \rangle$ and $\langle T \rangle$) corresponds to a BSPU and each column corresponds to a device. The label $\langle i, j \rangle$ on each node corresponds to a possible (i.e. satisfying the local constraint) assignment of BSPU i to device j . A node labeled $\langle i, j \rangle$ is connected by edges to all nodes $\langle i+1, j \rangle$, $\langle i+1, j+1 \rangle \dots \langle i+1, n \rangle$ in the layer below (ordering constraint). All nodes in the first (last, respectively) layer are connected to the node $\langle S \rangle$ ($\langle T \rangle$, respectively). Therefore, any path connecting nodes $\langle S \rangle$ to $\langle T \rangle$ corresponds to an assignment of BSPUs to devices fulfilling the earlier mentioned constraints (e.g. the thick path in Fig. 4 which corresponds to the assignment shown in Fig. 3).

Each edge of this layered assignment graph is then labeled with the sum weight of computation time and communication time appropriately: (1) the edges connecting node $\langle S \rangle$ to node $\langle 1, 1 \rangle$ till $\langle 1, n \rangle$ are weighted with the communication time of transfer the raw sensor data as 0, $c_{0,1}$, $c_{0,2} \dots c_{0,n-1}$. (2) In layer i (except for the last layer), each edge connecting node $\langle i, j \rangle$ to the layer below is first weighted with the computation time for BSPU i to process one frame at device j , i.e. $p_{i,j}$. Then, to the

weight on the edge from $\langle i, j \rangle$ to $\langle i+1, j+1 \rangle, \langle i+1, j+2 \rangle \dots \langle i+1, n \rangle$ in the layer below, a communication time required to transfer BSPU i 's output data, $c_{i,j}, c_{i,j}+c_{i,j+1}, \dots, c_{i,j}+c_{i,j+1}+\dots+c_{i,n-1}$ is added respectively. (3) For the last layer, each edge connecting node $\langle m, j \rangle$ to node $\langle T \rangle$ is first weighted with the computation time for BSPU m to process one frame at device j , i.e. $p_{m,j}$. Then, to the edge from $\langle m, j \rangle$ (except for $\langle m, n \rangle$) to $\langle T \rangle$, a communication time required to transfer BSPU m 's output data, $c_{m,j}+c_{m,j+1} \dots +c_{m,n-2}+c_{m,n-1}$ is added. As an illustration, the weights associated with the thick edges are shown in Fig. 4.

Fig. 4. The 6-BSPU-4-device assignment graph without contiguity constraint

In the last step, by applying a shortest-path search algorithm (e.g. Dijkstra algorithm), we can identify a path connecting $\langle S \rangle$ and $\langle T \rangle$ in this weighted graph which corresponds to an optimal assignment (Ψ_{opt} with the minimal τ).

The space complexity (defined as the number of nodes in the assignment graph) of this method is $O(mn)$. Thus the time complexity of shortest-path search step is $O(m^2n^2)$ if the Dijkstra algorithm is used. If a labeling method as proposed by Bokhari ([4]) is used for search, the time complexity can be reduced to $O(mn)$.

It is easy to convert our procedure to tackle the assignment problem with the contiguity constraint. We only need to remove all the edges which are “cutting a vertical edge”, i.e. the assignment of two adjacent BSUs onto two non-adjacent devices and all the edges connecting nodes $\langle S \rangle$ and $\langle T \rangle$ except for the ones connecting $\langle 1, 1 \rangle$ and $\langle m, n \rangle$. This simplified assignment graph is shown in Fig. 5. It is obvious that the two triangle areas (the grey area in Fig. 5) can not be part of the optimal path. Therefore, they can be removed from the assignment graph as well in order to reduce the space complexity. Following the similar weighting and search steps, the optimal assignment with the contiguity constraint can be obtained. This new assignment graph has some sort of relation with an earlier proposed assignment graph or ([5, 11, 12]): the nodes matrix in our solution is the transpose of the nodes matrix proposed earlier with the contiguity constraint.

Fig. 5. The 6-BSPU-4-device assignment graph with contiguity constraint

5 Performance

We implement the proposed assignment method in Java to get more in-depth understanding. This is, to the best of our knowledge, the first experimental study based on series of theoretical studies on chain-to-chain assignment problem. We record the CPU time of two computation steps in this method: the weighted assignment graph construction and the shortest-path search. For the “search” step, we use an open source library¹ which implements the Dijkstra shortest-path algorithm. The program is tested on a Windows XP machine with Intel Pentium 2.4G CPU and 1.5G RAM. Different pairs of m and n , e.g. (20,10), (40,10), ... (100,20), are tested and the number of assignment graph nodes ranges from 200 to 2000. Both the assignment with the contiguity constraint and the assignment without are tested (Fig. 6).

Fig. 6. The percentage of CPU time spending on each of two steps in assignment with contiguity constraint (a) and assignment with ordering constraint (b)

¹ <http://rollerjm.free.fr/pro/graphs.html>

One observation we could make is that the “construction” step is more time consuming than the actual “search” step in most of cases, which suggests that the evaluation on this kind of graph-based method should be based on the time complexity of the entire method instead of the “search” step only ([4-7]).

On the other hand, we also implement a program based on the optimal assignment approach proposed by Bokhari ([4]) and tested against our method with the contiguity constraint. Again, the CPU time of graph construction and shortest-path search are measured separately. Several pairs of m and n are tested and the results are shown in Table 1. Due to the reduced time and space complexity, our method outperforms the original Bokhari’s method. For a m -BSPU- n -device assignment, our method requires an assignment graph with $mn+2$ nodes compare to a Bokhari’s graph with $(m-n+2)*(m-n+1)*(n-2)/2+(m-n+1)*2+2$ nodes. For example, for the 60-BSPU-30-device assignment, an assignment graph with 14952 nodes will be constructed if Bokhari’s method is used. Because the adjacent matrix is used to represent the graph, which implies heavy memory requirement, the program runs out of memory before the completion.

Table 1. CPU time (in millisecond) comparison between Bokhari’s original method and our proposed method on chain-to-chain assignment with the contiguity constraint

Setting		CPU time of Bokhari’s method			CPU time of our method		
# BSPU	# device	Construction	Search	Total	Construction	Search	Total
20	10	94	63	157	16	47	63
30	10	392	172	564	31	47	78
40	10	985	673	1658	32	46	78
30	20	250	94	344	63	78	141
40	20	1017	563	1570	94	78	172
50	20	3996	2690	6686	141	94	235
40	30	360	156	516	172	94	266
50	30	2112	939	3051	235	109	344
60	30	Out of memory	Out of memory	Out of memory	282	110	392

6 Conclusion

In this paper, we study the problem of assigning a chain-structured biomedical signal processing task across several devices connected in a chain based on the context information. We present a graph-based method to search for an optimal task assignment which minimizes the end-to-end process and transfer delay. Our solution differs from earlier work by means of a different objective and the relaxation of a so-called contiguity constraint, which is a necessity in the earlier case but unnecessarily constraining in the healthcare applications. The proposed method also reduces the time and space complexity in case it is adapted to include the contiguity constraint.

However, we are aware that more complex M-health tasks can not be modeled by a chain of processing units. On the other hand, the involvement of other devices may also complicate the model of M-health platform, e.g. adding a home gateway into the configuration (Fig. 2) can create a second path between the MBU and the backend

server. Therefore, in the future work, we plan to address a more general model, i.e. (Directed Acyclic Graph) DAG-tasks-to-DAG-resources assignment problem. Since, very likely, no algorithms with polynomial time complexity will be ever found to solve the general problem, our future work will be focused on heuristic approaches, e.g. *Branch-and-Bound* ([13]) and *Genetic Algorithms* ([14]).

Acknowledgement

The authors thank Dr. Bert-Jan van Beijnum for his valuable comments. This work is part of the Freeband AWARENESS Project. Freeband is sponsored by the Dutch government under contract BSIK 03025. (<http://awareness.freeband.nl>)

References

1. 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.
2. 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.
3. Halteren, A.v., et al., *Mobile Patient Monitoring: The MobiHealth System*. The Journal of Information Technology in Healthcare, 2004. 2(5).
4. Bokhari, S.H., *Partitioning problems in parallel, pipelined, and distributed computing*. IEEE Transactions on Computers, 1988. 37(1): p. 48-57.
5. 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.
6. Nicol, D.M. and D.R. O'Hallaron, *Improved Algorithms for Mapping Pipelined and Parallel Computations*. IEEE Transactions on Computers, 1991. 40(3): p. 295-306.
7. 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.
8. 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).
9. AWARENESS, *Freeband AWARENESS project webpage*. 2005, <http://www.freeband.nl/project.cfm?id=494&language=en>.
10. Braun, T.D., et al. *A Taxonomy for Describing Matching and Scheduling Heuristics for Mixed-Machine Heterogeneous Computing Systems*. in *17th IEEE Symposium on Reliable Distributed Systems*. 1998.
11. Hansen, P. and K.-W. Lih, *Improved Algorithms for Partitioning Problems in Parallel, Pipelined, and Distributed Computing (Correspondence)*. IEEE Transactions on Computers, 1992. 41(6): p. 769-771.
12. Yeh, C.-C. *On the Power-Aware Resource Allocation for Linear-Pipelined Real-Time Tasks*. in *19th International Conference on Advanced Information Networking and Applications*. 2005.
13. 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: p. 1223-1240.
14. Wang, L., et al., *Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-Based Approach*. Journal of Parallel and Distributed Computing, 1997. 47: p. 8-22.