ALGORITHM FOR ENERGY-EFFICIENT AND AUTOMATIC CONFIGURATION OF PEDESTRIAN NAVIGATION SERVICES: SMARTPHONES ON CLOUDS

Monir H. Sharker and Hassan A. Karimi
University of Pittsburgh, Geoinformatics Laboratory, School of Information Sciences, Pittsburgh, United States

Keywords: Pedestrian Navigation Services, Smartphone, Cloud Computing, Energy-Efficient Configuration, Load Balancing.

Abstract: Smartphones have been playing a major role as personal navigation aids to pedestrians, among other travelers. Pedestrian Navigation Services (PNSs) on smartphones face challenges such as limited battery power, relatively lower computational speed, limited storage capacities, and varying quality of available networks. Automatic configuration of PNS components is very important to address these challenges. In this paper, we present a novel approach for optimum configuration of components and computations of PNSs between smartphones and cloud platforms. An algorithm is developed based on three models, Minimum Computation (*MinComp*), Minimum Communication (*MinComm*), and Balanced Computation-Communication (*BalCC*), to allow smartphones to operate navigation services optimally in a distributed environment over cloud platforms. Each model was simulated by using various values for each parameter.

1 INTRODUCTION

Advancements of key technologies including geo-positioning sensors, wireless communications, mobile devices, and the internet have been paving the way for development of Pedestrian Navigation Services (PNSs) on smartphones (Karimi, 2011). To better understand what PNSs are and how they work, we make a distinction between navigation systems and navigation services where the former is a reference to stand-alone systems, with all data stored locally and all computations performed in a centralized manner, and the latter is a reference to services with data and computation distributed over a network of computers. We define PNSs as those navigation services available on smartphones that can assist pedestrians to move from one place to another and can meet their specific needs and preferences.

While from a user's perspective, little differences can be observed between navigation systems and services, from a developer's perspective, there are challenges in developing navigation services, in particular PNSs. In PNSs, challenges are stemmed from dynamic characteristics of mobile devices such as limited battery power, relatively lower computational speed, limited storage capacity, and varying quality of available networks signal strength. PNSs are based on client-server architectures, where clients are increasingly smartphones and servers are service providers. Optimum distribution of data and computation over client-server environments, e.g., energy efficient services, requires a new algorithm that can automatically configure service components between the clients and the server. In this paper, we present an algorithm that allows PNSs on smartphones to be configured dynamically and automatically based on specific criteria where servers are cloud computing platforms. Service providers are increasingly resorting to cloud computing platforms primarily to address scalability of their services and to minimize services costs.

Unique aspects of PNS components and features require special attention to configure those in a manner aiming for energy-efficient and compute-efficient services. For example, some existing navigation systems and services provide navigational assistance to pedestrians by using road networks, as they are used for assisting drivers. Road networks are unsuitable and ineffective for assisting pedestrians with their mobility as

H. Sharker M. and A. Karimi H..
ALGORITHM FOR ENERGY-EFFICIENT AND AUTOMATIC CONFIGURATION OF PEDESTRIAN NAVIGATION SERVICES: SMARTPHONES ON CLOUDS.
DOI: 10.5220/0003953400850090
In *Proceedings of the 1st International Conference on Smart Grids and Green IT Systems* (SMARTGREENS-2012), pages 85-90
ISBN: 978-989-8565-09-9
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

pedestrians usually walk on sidewalks and not on road center lines. For effective navigation assistance to pedestrians, PNSs must contain pedestrian paths in the underlying navigation environments. The core data in a pedestrian path database is a network of sidewalks which contains geometry, topology, and attributes of the pedestrian path (Kasemsuppakorn and Karimi, 2009). Geometrical data in a pedestrian path database include coordinates of decision points—a decision point is a juncture whereby a decision must be made by the pedestrian (e.g., an intersection)—and of intermediate points on sidewalks. Topological data in a pedestrian path database include data about connectivity between decision points and sidewalks. Attribute data in a pedestrian path database includes data such as sidewalk name, sidewalk width, and sidewalk type.

Given all these PNS components and features, the limitations of mobile devices, diversity of user requirements, and the environmental factors, a dynamic configuration scheme is a key requirement for providing efficient services. For example, given a complex computational task and a mobile device with low battery power, it is best to have the task performed on the server side. On the other hand, a simple computational task that is required frequently is best to be performed in the client side to minimize communication overheads. Many combinations, considering device constraints and computations at hand, among other factors, are possible for which an efficient solution is desirable.

The contribution of the paper is an algorithm for automatic configuration of data and computation components in PNSs. The algorithm addresses many situations where minimizing energy consumption is the highest priority. The rest of the paper is organized as follows. In Section 2, the main components of PNSs and possible model configurations are discussed. Section 3 describes the algorithm. The models and the simulation of the algorithm are explained in Section 4. Simulation results are discussed in Section 5. Section 6 highlights related works and Section 7 concludes the paper and discusses future research.

2 COMPONENTS AND MODELS

The main components and functions of PNSs include sidewalk network map database, geocoding, map matching, routing, smartphone (GPS-enabled), and cloud computing server and/or service provider. Sidewalk network map databases are the main data source for PNSs. Geocoding is the process of assigning geographic coordinates (lat/long) to a given place by comparing its description to the descriptions of location-specific elements in the reference datasets (Goldberg, 2007). Map matching is a technique for matching pedestrian trajectories to correct pedestrian paths in the sidewalk network (Karimi et al., 2006). Routing computes a preferred path from any given origin to destination (Kasemsuppakorn and Karimi, 2009). Proposed models of PNSs are explained bellow in brief.

2.1 Minimum Computation (*MinComp*)

For a resource poor client device, the *MinComp* model is considered where most computations are submitted to the cloud. In the *MiniComp* model, with the assumption that the device battery is low, its computation capability is limited, and its available storage is low, all computational tasks are submitted to the cloud. In this model, the client is mainly responsible for submitting the user request to the cloud, receiving the response from the cloud upon task completion, and presenting the results to the user.

2.2 Minimum Communication (*MinComm*)

In the *MinComm* model, most computations are performed at the client device provided that the client has sufficient battery power, computational capability, and storage capacity. This model is suitable in situations where the device can handle the computational load and the user needs fast response. This model is also suitable in situations where network availability is limited and network quality is poor. The main responsibilities of the cloud in this model are map storage, map rendering, and map data transfer.

2.3 Balanced Computation-communication (*BalCC*)

The third model, *BalCC*, is the most suitable model for many clients since client devices fall into this category most often. This is basically a trade-off between the *MinComp* and *MinComm* models. In situations where the client device has an average battery power, average computation capability, moderate amount of storage capacity, and network availability is in moderate level, the *BalCC* model provides the most optimal solution. In this model, some of the computations are performed in the client side while some in the cloud maintaining a balance

Table 1: PNS model specifications, decision criteria along with their pros and cons.

Model	Functions on Smart phones	Functions on Cloud Servers	Adaptation Criteria	Pros	Cons
Minimum-Computation	• GPS data acquisition • Get user input source/destination /options • GPS data/user input transfer (send) • Map data transfer (receive, cache) • Map match data transfer (receive) • Map data update	• GPS data transfer (receive) • GPS data map match • Map extent retrieval • Geocode user data • Route computing • Map rendering • Map data transfer (send)	• Power/energy consumption • Cost: pay per usage • Computation capability of client	• Minimum storage • No burden of updates on Smart phones	• Privacy compromised • Subject to Internet speed and interruption • Frequent GPS data update
Balanced Communication-Computation	• GPS data acquisition • User input source/destination /options • Map data transfer (receive, cache) • Geocode user data • GPS data map match • Map match data transfer (send) • Map data update	• Map match data transfer (receive) • Route computing • Map extent retrieval • Map rendering • Map data transfer (send) • GPS Data update	• Faster • Increased utilization	• Loads are balanced • Faster response	• Synchronization could be an issue • Frequent/Moderate GPS data update
Minimum-Communication	• GPS data acquisition • GPS data transfer(send) • Map data transfer (receive) • GPS data map match • User input source/destination /options • Geocode user data • Map extent retrieval • Route computing • Map rendering	• GPS data transfer (receive) • Map retrieval • Map data transfer (send)	• Minimum Internet interruption and disconnection • Cost: One-time fee for installation	• No/Limited privacy compromised • Infrequent map data download	• High computing demand on Smartphones • Large storage on smartphones

between the client and the server. The flexibility with the *BalCC* model is that it chooses the optimum load balance both for the client device and the cloud based on the dynamic status of the resources on the fly.

3 ALGORITHM

The algorithm has a decision engine where the decision, about which model to choose on the fly, is made. To make a decision, the algorithm (see Figure 1) consults parameters status such as battery power (BP) level, storage capacity (SC) level, processor speed (PS) of the client device, and the environmental factors like network availability (NA).

Battery power (BP) is the most important parameter considered by the algorithm. The objective is to minimize energy consumption. However, once the algorithm recognizes that the client device has high level of energy, it takes the other parameters into account. In situations where power is in medium or low range, the algorithm gives BP the highest priority to minimize energy consumption.

Each device's parameter responds by providing its level of strength by means of a score in accordance with the probability of the status. By accumulating all the response scores, the algorithm generates a scaled score and sends the score to the decision engine. The decision engine decides the model, with the highest score, to be executed.

The algorithm has the following steps.

1. Obtain specifications of the models.
2. Acquire the client device status probabilities along with a scaled score for all parameters (BP, SC, PS, and NA).
3. Monitor continuously the current status of the device and feed the latest status score.
4. Calculate the overall score (expected value) for each model taking all the possibilities

into consideration.

5. Select the model with the highest score.
6. Continue the process upon request.

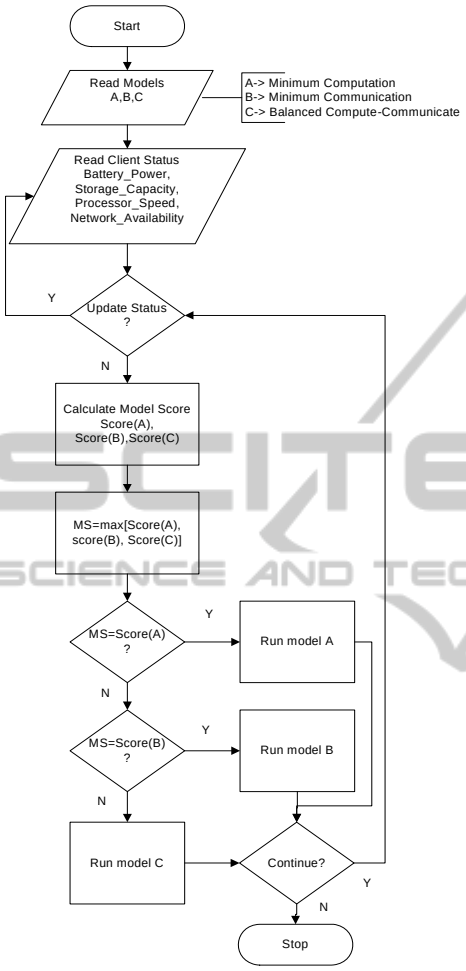

Figure 1: Flowchart of the algorithm.

4 MODELS AND SIMULATION

4.1 Models

To calculate the overall score (expected value) for each model, for one set of probabilities, the following equations are used:

$$Score(MinComp) = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} P_i(BP) \times P_j(SC) \times P_k(PS) \times P_l(NA) \times ScaleVal(A) \quad (1)$$

$$Score(MinComm) = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} P_i(BP) \times P_j(SC) \times P_k(PS) \times P_l(NA) \times ScaleVal(B) \quad (2)$$

$$Score(BalCC) = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} P_i(BP) \times P_j(SC) \times P_k(PS) \times P_l(NA) \times ScaleVal(C) \quad (3)$$

where, p_i(bp) is the probability of battery power of level i, i= 1,3 (high, medium, low), p_j(sc) is the probability of storage capacity of level j, j= 1,3 (high, medium, low), p_k(ps) is the probability of processing speed of level k, k= 1,3 (high, medium, low), and p_l(na) is the probability of network availability of level i, i= 1,3 (excellent, good, poor). The scale value (ScaleVal) is determined through a Bayesian belief network based on the parameters in each model.

The above score calculation, for each model, is only for one set of probabilities of the parameters. All possible combinations of probability set are then generated and the score for each model is calculated for all possibilities. Thus, comparing the overall score generated for the models for a given set of probability, the model with the highest score is chosen.

4.2 Simulation

We used influence diagram, as shown in Figure 2, to simulate the algorithm. The decision variable is set as *Model Decision* having three values *MinComp*, *MinComm*, and *BalCC*. Three variables, *Battery_Power*, *Storage_Capacity*, and *Processor_Speed*, at three levels of probabilities, *High (0.6)*, *Medium (0.3)*, and *Low(0.1)* for each, are considered. A fourth variable, *Network_Availability*, also has three levels of probabilities as *Excellent (0.6)*, *Good (0.3)*, and *Poor (0.1)*. The levels of variables could be chosen to any level with appropriate scaling factor. We simulated all possible outcomes for each model for all possible input sets. Then comparing the model outcomes for each input set, the decision model with the highest score suitable for the given situation was selected. We used GeNIe (GeNIe 2011), to create and evaluate a decision theoretic model for PNSs.

5 RESULT AND DISCUSSION

We used three levels of probabilities for each variable, which means 3!=6 sets of probabilities for each variable. In total, there are 6x6x6x6=1296 possible different decision points, considering all four variables together. The score for each model was calculated and a decision was made. The possibilities of Storage_Capacity (SC) are shown in Figure 3. We used a scale value ranging from 2.0 to 10.0 for each model. The final score calculated for each model hence ranges from 2.0 to 10.0.

For one example set of combination, i.e., BP[High(0.1), Medium(0.3), Low(0.6)], SC[High(0.1), Medium(0.3), Low(0.6)],

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

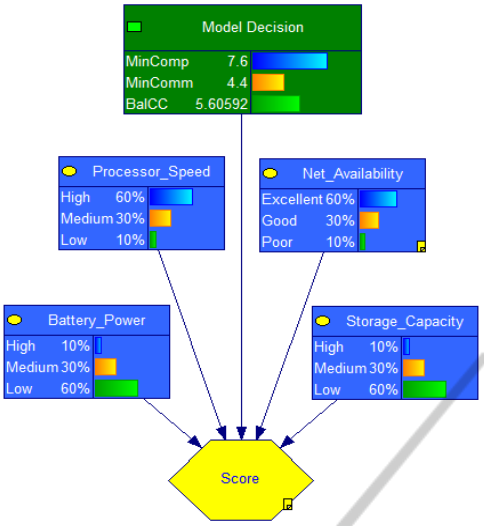

Figure 2: Influence diagram showing simulation result for an example set of combination.

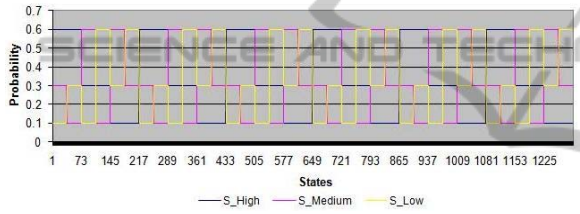

Figure 3: Possible states of the chance variable, Storage_Capacity.

PS[High(0.6), Medium(0.3), Low(0.1)], and NA[Excellent(0.6), Medium(0.3), Low(0.1)], the calculated scores are: *MinComp* = 7.6, *MinComm* = 4.4, and *BalCC* = 5.61. The result for this specific case is verified by Genie (Bayesian Network) as shown in Figure 2. The complete simulation result is plotted in Figure 4 for all the 1296 decision alternatives. This result gives an idea about how the decision switches between the models over the change of probabilities of variables. We will conduct the experiment on a real PNS in next phase.

6 RELATED WORKS

Fundamental challenges for mobile computing, due to limited resources on mobile devices, poor wireless network, and limited energy sources are outlined by Satyanarayanan (1996). The state-of-the-art cloud computing paradigm is analysed by Buyya et al. (2009, 2011) where cloud computing is recognized as the technology to offer computing services anywhere in the world on demand. In a distributed mobile computing framework, to minimize the power consumption for the client, computation loads are shared among the nodes such that the required energy can be fairly used as contributed by Liang et al. (2007). Without a priori knowledge of the application, the optimal deployment, that minimizes CPU usage at the mobile device keeping the used bandwidth minimal, is calculated (Verbelen et al., 2010). Banavar et al. (2000) proposed a new application model for pervasive computing that gives techniques such as device-specific rendering, application apportioning, and application adaptation. Hao-Hua Cho et al. (2004) describe the challenges of constructing a new, wireless, web-service architecture, based on a client model.

Stoer and Wagner (1997) describe an algorithm that finds the minimum cut to divide a graph in two partitions. The approach here is that to calculate the optimal partitioning of the components, it represents the services as a weighted graph, and transforms the deployment problem into a graph partitioning problem. Han et al (2008) present graph partitioning algorithms aimed specifically at the problem of partitioning software for mobile devices. One of the key considerations is an adaptor which monitors system-level states, such as CPU load or network bandwidth. They use the adaptors to adjust application parameters or activate functional reconfiguration actions based on a fuzzy control model.

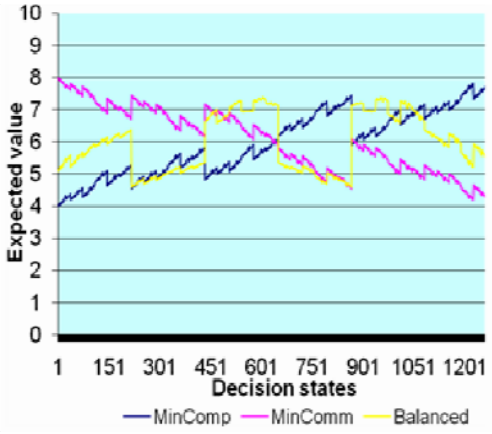

Figure 4: Simulation result for all alternatives.

However, none of these research projects takes all the parameters mentioned in this paper into account and considers all possible combinations. Our proposed algorithm contributes by using a Bayesian belief network (influence diagram) to find an optimal configuration. For any combination of

parameters, the influence diagram generates an overall score (see Figure 2 for an example combination) based on which the decision is made. Accordingly, each function in PNSs is executed either on the client or on the server depending on which provides the most optimal performance for the situation at hand. This algorithm also utilizes resources on user's device optimally and more effectively.

7 CONCLUSION AND FUTURE WORK

This paper describes different configurations explaining all possible computations and communications models in PNSs using smartphones (clients) and clouds (servers). An algorithm that analyses these models, in order to automatically decide a suitable configuration, is presented. The algorithm was simulated and the result shows that the algorithm provides the most optimum model given the environmental and device parameters of PNSs. The algorithm is dynamic and reconfigurable. Once the functions, specifications of power usage, storage capacity, processing capabilities of the mobile device, and the network provider information are available, this algorithm will automatically decide on the most suitable model for the scenario. Future work will include implementation and deployment of the algorithm on mobile devices and cloud computing platforms.

REFERENCES

Banavar, G., Beck, J., Gluzberg, E., Munson, J., Sussman, J., and Zukowski, D. (2000). Challenges: An Application Model for Pervasive Computing. *6th ACM/IEEE International Conference on Mobile Net-working and Computing*, Boston, USA: ACM Press, pp. 266-274.

Buyya, R, Yeo, C. S., Venugopal, S. Broberg, J., and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599.

Buyya, R., Broberg, J., and Goscinski, A. (2011). Cloud Computing: Principles and Paradigms. John Wiley & Sons, Inc., NJ.

GeNIe (Graphical Network Interface) (2011). http://genie.sis.pitt.edu/about.html

Goldberg, D.W., Wilson, J.P. and Knoblock, C.A., (2007). Fromtext to geographic coordinates: the current state of geocoding. *Journal of the Urban and Regional Information Systems Association*, 19, 33–46.

Han, S, Zhang, S, Cao, J., Wen,Y., and Zhang Y. (2008). A resource aware software partitioning algorithm based on mobility constraints in pervasive grid environments. *Future Generation Computer Systems*, 24(6):512-529.

Hao-hua, C., Chuang-wen, Y., and Chao-ming, T., (2004). Challenges: wireless Web services. *10th International Conference on Parallel and Distributed Systems (ICPADS 2004)*, IEEE Computer Society, pp. 657.

Karimi, H. A., Conahan, T., and Roongpiboonsopit, D. (2006). A Methodology for Predicting Performances of Map-Matching Algorithms. *6th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2006)*. Hong Kong, December 4-5.

Karimi, H. A. (2011). Universal Navigation on Smartphones. Springer.

Kasemsuppakorn, P. and Karimi, H. A. (2009). Personalized Routing for Wheelchair Navigation. *Journal of Location Based Services*, Vol. 3, No. 1, pp. 24-54.

Liang, W.Y., Hsieh, Y.M., and Lyu, Z.Y. (2007). Design of a dynamic distributed mobile computing environment. *International Conference on Parallel and Distributed Systems, IEEE Explore.*

Satyanarayanan, M. (1996). Fundamental Challenges in Mobile Computing. *15th ACM Symposium on Principles of Distributed Computing*, pp. 1-7.

Stoer M. and Wagner F., (1997). A simple min-cut algorithm, *J. ACM*, 44(4):585-591.

Verbelen, T., Hens, R., Stevens, T., and De Turck, F. (2010). Adaptive online deployment for resource constrained mobile smart clients. *3rd International ICST conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications(Mobilware).*