An Autonomic Approach To Resilence In Sensor Network Query Processing

Alan B. Stokes
School of Computer Science, University of Manchester, Manchester M13 9PL, United Kingdom

Abstract. Sensor network query processors make numerous assumptions when compiling a query for running on a wireless sensor network, i.e. consistent network topology, reliable data transmissions, etc; some of these assumptions do not hold in real life deployments, like a consistent network topology, as nodes can fail unpredictably. We propose that by using the autonomic computing principles of self healing, self protection, self configuration and self optimisation in the form of an autonomic component which interacts with the running query execution plan on the wireless sensor network, the "useful lifetime" of a wireless sensor network can be increased beyond what is achieved by non-adaptive query execution plans.

1 Introduction

As Wireless Sensor Networks (WSNs) become applicable to more applications, the assumptions of the sensor network query processors (SNQPs) used to generate query execution plans (QEPs) become one of the limiting factors in this process. One key assumption which SNQPs like TinyDB [7] and SNEE [2] use is that the condition of the WSN during the entire query life time is constant. This assumption does not hold in real life deployments, where sensors can malfunction, fail, or be destroyed without warning. This is also true for the energy levels of the motes, which are guaranteed to diminish with time and can cause motes to become non-functional.

To compensate for this assumption, some kind of adaptive behaviour needs to be employed which can adjust the QEP to the new condition of the WSN. We plan to introduce adaptivity to a SNEE QEP by the use of autonomic computing principles (self healing, self protecting, self configuration and self optimisation) [6]. We aim to explore how the effect of using an autonomic manager which takes the QEP inside the WSN as its managed artefact has on the measured ‘useful lifetime’ of a QEP. Hopefully we will discover that the use of such a component will extend the useful lifetime beyond the current level given by QEPs generated by SNEE.
2 Background to Approach

Autonomic computing tries to emulate the behaviour exhibited by autonomic systems in nature, for example our autonomic nervous system which regulates the behaviour within our bodies [1]. In Computing, and in the case of WSN’s in particular, this system can be emulated by a component which monitors the QEP to detect changes to the QEP.

When a change is detected the autonomic manager would use knowledge learned or programmed into its knowledge base, to decide if any adjustment is required. If an adjustment is required, the autonomic manager would then determine what change is to be made and how. The final phase of the autonomic cycle is execution, where the autonomic manager communicates with the QEP, making the alterations. The QEP would then acknowledge that the changes have taken place, and the cycle repeats. This cycle is illustrated in Figure 1 which represents the MAPE control loop.

3 Current Progress

Currently most of our focus has been aimed at the analysis stage of the MAPE control loop. Here we have been developing cost models which measure the effect of losing a node in the WSN which participates in the QEP in relation to
cardinality and accuracy (by the means of confidence intervals defined in [5]).

As any logical operator can be scheduled to execute in more than one physical node, being able to determine what operators are running on a specific node is essential, do allow us to do this we have implemented a data structure called an Instance Operator Tree (IOT) which places instances of operators on each site and encapsulates communication between QEP fragments and execution sites [3]. This allows us to track what data flows through each node more effectively.

We are also currently developing a notation which defines what local changes are possible to adapt to a failed node. These changes are expected to be only local changes, but can have temporal effects on operators further down the QEP.

Finally we are trying to develop a equivalence relation which can determine at the time of compilation if 2 motes are functionally equivalent. This is to try to exploit the concepts of clustering mentioned in [4] the goal in our case will be to enable, at runtime, the dynamic assessment and characterization of cluster members so as to allow the autonomic manager to choose the new cluster head when required.

4 Foreseeable Challenges

There are many aspects which have not been confronted yet, e.g., developing code which can be embedded onto the motes whilst still allowing enough space to place the QEP of the mote in question. This code should allow the QEP to respond to requests and transmissions from the autonomic manager.

There is also the problem of getting the changes from the autonomic manager to the motes in question within the WSN as efficiently as possible and hopefully without affecting the rest of the motes within the QEP.

Finally we mention the problem of collecting current metadata from the WSN during runtime. We are considering either piggy backing the metadata on the end of the tuple transmissions between nodes, or else allowing a specific agenda execution cycle to be for metadata collection.

5 Conclusion

We conclude that even though we are still early in the research cycle, we are hopeful that by introducing a autonomic manager component which communicates with the QEP, the ‘useful lifetime’ of a SNEE QEP can be extended beyond what is capable of a non-adaptive SNEE QEP. We have broken down the autonomic manager into its separate phases and have defined what progress we have had in developing tools which the autonomic manager can use to analysis the results of the QEP and the evolving state of the WSN. We finally listed
a few of the challenges we have yet to face.

Acknowledgements Thank you to my supervisors Alvaro A. A. Fernandes and Norman W. Paton for all their help, advice and guidance.

References