Embedded Database for Remote Process Management System

Kalyani N. Satone
M.E(C.S.E) ,S.D.C.O.E,
Wardha

Madhuri M .Pal
M.Tech(2^{nd} Year Embedded) ,
S.D.C.O.E,
Wardha

ABSTRACT
This paper investigates embedded databases for the MicroBaseJ project. The paper aims at the development of an integrated database and a user interface for a typical 3G mobile phone with Java MIDP2 capability. One of the key objectives is to target a generic solution.
A number of commercial sectors could benefit from this solution such as horticulture, building management, pollution/water management, industry etc.
Four embedded databases based on J2ME have been investigated. Two of the four have been evaluated and analyzed. The size of processed data is limited to 20000 records when using the wireless toolkit simulator and 11000 records when using a mobile phone**.**

1. INTRODUCTION
It's a global market where mobile phone user base in excess of 2.8 billion, the current market for information based management processes may be based on at least 75000 applications in the Australian market in the various field like horticultural and water/irrigation management sectors [1]. Further generic opportunities exist in chilled assets, industry, energy, pollution and security related applications. Mobile applications have covered the area of medicine, education, household appliances control, plant control and spatial information service. However, little has been seen in the horticultural sector.
There has been a number of research work related to the use of cell phone as a remote monitor. Most of them focused on telemedicine, education and plant control. Ravi, Chathish and Prasanna [2] proposed the WAP (wireless access protocol) protocol to develop an alarm management program for providing alert signals when any received data exceeds a preset value for the selected process variables. An implementation of a WAP-based telemedicine system was developed by Hung, Zhang et al.[3]. They utilized WAP devices as mobile access PCs for common inquiry and patients' common data. Authorized users can view the patient's data, monitor blood pressure and electrocardiogram on WAP equipment in store-and-forward mode. Nikolova, Meijs and Voorwinden [4] developed a technique for interconnecting home and mobile networks to enable the control of home appliances, connected in a home
network, from a remote mobile phone or a web pad. The remote control functions include remote mobile programming of VCR, remote mobile control of heating thermostats, remote mobile monitoring using security cameras, etc.
Rahman and Bhalla proposed to use a user-friendly language developed for a Relational Database Management System (RDBMS), Query-By-Example (QBE), to support spatial queries on mobile devices [5]. Most applications of mobile phone remote management utilize J2ME (Java 2 Platform Micro Edition) as a tool for user interface implementation. Yang and Kou [7] presented two techniques that uses J2ME to monitor and control PC clusters from mobile phones. Nichols and Myers [8] presented to generate a smart phone interface generator using Microsoft's Windows CE-based Smartphone platform. These interfaces allow users manage each appliance's full functionality and are consistent with other interfaces of the phone.
There are many techniques for communication, including Bluetooth, Zigbee, RFID (Radio frequency Identification), WAP and SMS (Short message service) that relate to the distance between the client and server. Siegemund and Florkemerier [9] presented an interaction pattern using RFID labels to pass messages. An integrated system based on WEB/WAP framework for remote monitoring and control of industrial processes was proposed by Nikolakopoulos, Koundourakis and Tzes [10]. User could access to the system using a WEB browser or a WAP-enabled mobile phone.

2. EMBEDDED DATABASE
Databases have provided efficient information retrieval engines for good number of applications for decades. However, there are few applications that have embedded the database into cell phone due to the limitation of resources within the mobile phone. This includes the power, network connection and memory storage. In recent years, with the development of technology, the price of hardware has dropped significantly.
The mobile phones have become more functional and powerful. The embedded database can now be realized in handsets. Currently, there are few embedded databases available for the handsets, which includes Perst Lite [11], PointBase [12], db4o [13]. RMS (Record Management System) is the basic API of J2ME which can store, retrieve and delete records.
Table1 provides the comparison of Perst Lite, RMS, PointBase and db4o based on different features. Perst [14] is a simple and object oriented embedded database. It is easy to use and provides high performance. Perst also provides fault tolerant support (ACID (Atomicity, Consistency, Isolation, Durability) transactions) and concurrent access to the database.
Tight integration with programming language is the main benefit of Perst. Perst stores objects directly without packing/unpacking code (which has to be written for traditional relational databases), so there is no gap between database and application data models.

Table 1: Database Comparison

	Pointbase Micro	Perst Lite	db4o	RMS
Size	45K for J2ME MIDP, 90K for J2ME CDC	30K~300K	250K	Default component of cell phone
Support language	JAVA	JAVA & .net	JAVA & .net	JAVA
platforms	J2EE, J2SE, MIDP, personal JAVA	J2ME, J2SE	JDK 1.1 or later	All J2ME
Application platforms	PDA, mobile handset, PC	PDA, mobile handset	PDA, PC	PDA, mobile handset, PC
Database engine	Relational	Object-oriented	object-oriented database	None
Query language	Subset of SQL92	QBE, perst search method	QBE OR SOND	None
JDBC	JDBC subset	none	none	none
utilities	Console, others	none	Defragment (not GUI)	none
reflection	yes	none	no	none
Ease of use	good	Very good	Very good	none
flexibility	Cross-platform	Cross-platform	Cross-platform	none
Performance	good	good	poor	Very poor
Open source	none	yes	yes	yes
security	encrypt	encrypt	none	no
strength	Super-small footprint	Easy installation	Easy installation	simple
weakness	SQL and JDBC subset	Non-standard query mechanism	Non-standard query mechanism	Without index, slow search engine
cost	US$299	US$2000 for per application per year	US$100 for personal, US$ 1000 for corporate	none

PointBase Micro [12] is a platform-independent Java relational database optimized to run on the Java 2 Micro Edition (J2ME CDC and CLDC/MIDP) and J2SE platforms[12]. It has an ultra-compact footprint (<45K for J2ME MIDP) and can be easily embedded within a Java application, making it transparent to users from the time of deployment. PointBase Micro is ideal for notebooks, PDAs and emerging Java-enabled devices. It provides effective data management for rapid and efficient mobile enterprise applications created by software vendors, systems integrators and corporate IT departments.

db4o [13] is the open source object database that enables Java and .NET developers to slash development time and costs and achieve unprecedented levels of performance. The unique design of db4o's native object database engine makes it the ideal choice to be embedded in equipment and devices, and in real-time control systems - in brief: in all Java and .NET environments without database administrator (DBA).

Perst Lite is chosen as the database embedded into handsets. The Perst Lite J2ME database has the simplicity in design and is a high performance database within the resource limit of most intelligent mobile and embedded devices. Perst Lite retains most of the features of the open source Perst since 2003 [14]. These include B-tree, Patricia Trie, Bit index, T-Tree and R-Tree indexes as well as List, Relation, and Set collections, all protected by transactions supporting the ACID properties (Atomicity, Consistency, Isolation and Durability). Perst Lite also offers additional features of multithreaded access, data encryption and asynchronous replication.

3. ARCHITECTURAL DESIGN OF MICROBASEJ

A. Microbuses Architecture

MicroBaseJ project investigates integrated database and user interface for 3G mobile phone with Java MIDP2 capability. It consists of the Cellular Sentinel, mobile phone user interface, server and public wireless network. The architecture of MicroBaseJ is shown in figure 1.

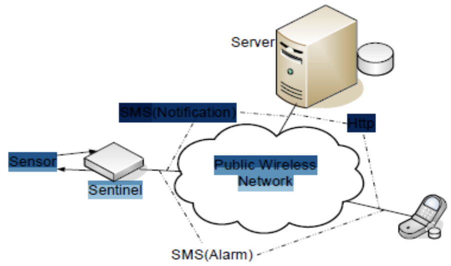

Fig 1: Architecture of MicroBaseJ

The Cellular Sentinel collects the information about monitoring and control, and sends these data to the server or mobile phone. The server / mobile phone will receive notification and alarm information from the Cellular Sentinel and store them into database, including temperature, humidity, alarm, GIS (geographic Information System), weather, telemedicine and remote vision, etc. The alert message will present on the mobile phones immediately.

The mobile phone provides a user interface for the subscriber to enquire or update the database, saves the message into local memory and plots the figure locally. Users can access data in the server database through a friendly user interface. The efficiency of different techniques and protocols for mobile phone interface will be compared and analyzed using metrics of response time, integrity element and cost. Notifications will be sent to server and saved as backups. And the mobile phone can access the server data through the network. This paper focuses on point-to-point interface between the sentinel and the mobile phone user. Figure 2 shows the block diagram of MicroBaseJ by highlighting the interface between the sentinel and mobile phone. Data Request asks the data from sensors at a sample rate (for example every hour) and data is saved in the object and ready to send.

Data Transfer sent the object of collected data to the user's mobile phone. The remote phone receives the notification from Sentinel as background in Data receive function and Data Storage function stores it into the Perst Lite database. The specified data will be look up from the database in terms of the user requirements and ready to present in Data Present function. Sample Rate Setting provides a menu to user to reset the sample rate and send to the Sentinel. The sentinel will store these data in Sample Rate Storage function.

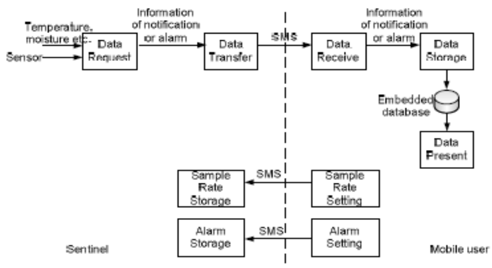

Fig 2: The logical Diagram of MicrobaseJ

B. MicroBaseJ Architecture

The Cellular Sentinel is one of the pocket based informatics for asset management and remote control solution of preference. The DS80C400 includes 1M RAM, 1M flash, 144-Pin SODIUM, and networking includes 1-Wire, CAN, serial, 10/100 Ethernet connections. The TINI provides physical connectors to interface the TINI with other device such as Ethernet network, a serial device and a 1-Wire network. It provides a friendly interface for remote asset management and control systems.

The developers can choose different language, such as Java, C or even coding in 8051. The Wavecom FASTRACK modem provides instant wireless capabilities. Housed in a rugged metallic casing, the FASTRACK modem is built to withstand the toughest environments. Its open interfaces and AT commands can embed and run the application right on the WISMO platform and it is GSM/GPRS enabled.

The Sentinel is the product of Mobile Control Solutions (MCS) Ltd. N.Z. which uses this platform and others to collect information from electronic sensors and send to the mobile user. MicroBaseJ intends to use Short Message System (SMS) as communication protocol to transmit collected data to user handset.

C. IDE and Software

Eclipse, an open source IDE (Interactive development environment) is chosen as the development environment. This paper is focused on creating an extensible development platform, runtimes and application frameworks for building, deploying and managing software across the entire software lifecycle, including enterprise development, embedded and device development, rich client platform, rich Internet applications, application frameworks, application lifecycle management and service oriented architecture.

The Eclipse version of this paper is Eclipse DSK (Development Software Kit) 3.2.2. The developed language, Java Platform Micro Edition (J2ME), provides a flexible and robust environment for applications based on mobile and other embedded devices— mobile phones, PDAs (personal digital assistants), TV settop boxes, and printers. J2ME contains elastic user interfaces, robust security, built-in network protocols, and support for networked and offline applications that can be downloaded dynamically. J2ME applications can be portable across many devices.

The Sun Java Wireless Toolkit (formerly known as J2ME WTK) is a set of tools for generating Java applications that run on devices obedient with the Java Technology for the Wireless Industry (JTWI, JSR 185) specification and the Mobile Service Architecture (MSA, JSR 248) specification. It includes built-in tools, utilities, and a device emulator. This paper adopts WTK2.5. The Sun Java Wireless Toolkit 2.5 contains all advanced development features found in version 2.2, 2.3 Beta, 2.5 Beta 2

4.DATABASE PERFORMANCE EVALUATION AND ANALYSIS

The key area of this paper is the implementation of embedded database. Insert function, search function, delete function and modify function are the major features in database. These articles focuses on the search function and insert function. The search function includes sequence and random search. While comparing Perst Lite and RMS, it is based on both simulation and real device. The comparison is based on the Java platform jre1.6.0_01 and Eclipse DSK 3.2.2.

A. Embedded Database Performance comparison based on simulation

Comparison is based on Sun Java Wireless Toolkit 2.5 DefaultColorPhone. It compares Perst Lite to RMS based on Insert function, Search function. The executing time is the average for each record during the whole evaluation. Within the Insert function, the code sets Perst Lite pool size as 64K, 100 records as one transaction. Figure 3 shows that the Perst Lite used around one tenth running time of RMS for Insert Function. The major reason is that Perst Lite used 100 records per transaction, this can really reduce the Input/Output (I/O) overhead. The performance of Perst Lite is better than RMS.

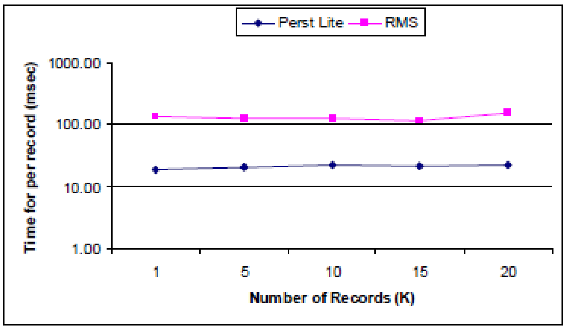

Fig 3: The Insert Function Performance Comparison

Figure 4 compares Perst Lite and RMS in Sequence Search. RMS put all records into enumerator and search from the beginning to the end. Perst Lite set all records into iterator and go through them. The result shows that the performance of Perst Lite is better than the RMS. The searching time for RMS keep increase, while Perst Lite keep stable when record number less than 16000, after that the searching execution time rises significantly.

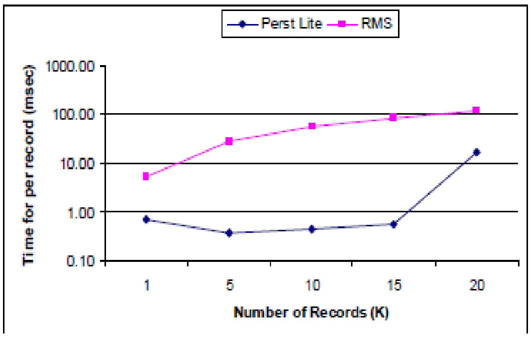

Fig 4: The sequence Search Performance Comparison

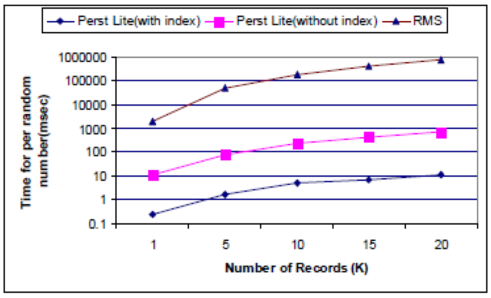

Fig 5: The Sequence Search Performance Comparison

There are 500 random numbers generated by the program in random search. RMS retrieve all records by using the random numbers. The run time is the average for each random number. Perst Lite did the same search with index and without index. Figure 5 shows that Perst Lite need less time to locate the random number than RMS. Furthermore, the search without index takes about 10 times more than that with index.

Table 2 Database Size Comparison

Record numbers (K)	Perst Lite (KB)	RMS (KB)
1	282	111
5	1225	556
10	2530	1112
15	4024	1677
20	5141	2241

Table 2 compares the database size of Perst Lite to RMS based on Sun Java Wireless Toolkit 2.5 DefaultColorPhone simulation. The record length is about 110 bytes. The data size of Perst Lite need twice of RMS. It means that Perst Lite needed more than 100 bytes in each record for the database overhead.

B. Embedded Database Performance comparison based on cell phone

The second part evaluation is based on the mobile phone, Nokia N73 with 42M memory and extra 1G miniSD card. The sized of processed data were limited to 11000 records. Figure 6 demonstrates the comparison of Insert Function between Perst Lite and RMS. It shows that both Perst Lite and RMS take more running time with the increase of record number. Perst Lite reflects better performance than RMS.

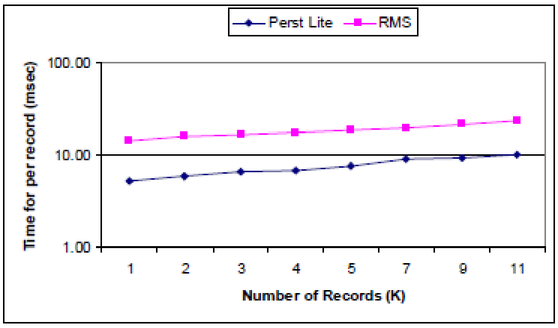

Fig: The Insert Function Performance Comparison

The Comparison of sequence search is illustrated in Figure7. It seems that Perst Lite and RMS performance data are quite close. There is a significant difference with the result based on the simulation. The reason may be that the database was stored into the mobile phone memory rather than the disk such as, it can reduce the I/O operation significantly.

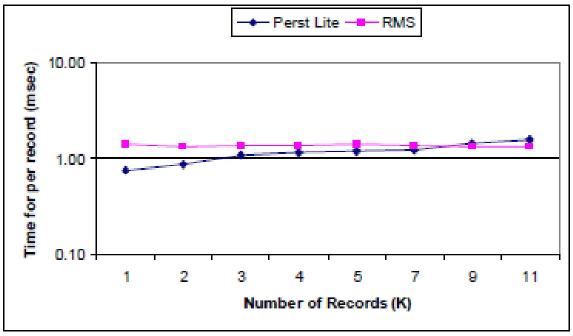

Fig 7: The Sequence Search Performance Comparison

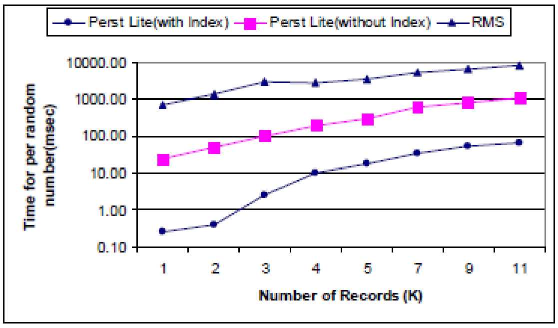

Fig8: The Sequence Search Performance Comparison

Figure 8 illustrates the comparison of random search performance of RMS and Perst Lite with index and without index. The code generates 1000 random numbers and calculates the average timing for each random search. It shows that Perst Lite performance is better than RMS although all data stored in memory. The random search with index needs less than one-tenth time of the search without index.

C. Performance Comparison Summary

Through simulation, Perst Lite performance is much better than the RMS in insert function, sequence search and random search. The Perst Lite has index which can locate records quickly. Secondly, the Perst Lite uses 64K as pool size. These can really reduce I/O operations then decrease the access time. In contrast, on Nokia N73, the performance of RMS improves markedly, sequence search of Perst Lite are similar to RMS. Insert function of RMS is just more than twice slower than that of Perst Lite. The major reason might be that the database is stored in the phone memory rather than extended memory. However, on Nokia N73, the performance difference on random search with and without index between Perst Lite and RMS is still significant. Table 3. Perst Lite and RMS timing for per (based on 5000 records, msec)

Case			Perst Lite	RMS
Simulation	Insert function		20.15	127.69
	Sequence search		0.39	27.53
	Random search	with index	2.2	
		without index	72.1	44943.99
Nokia N73	Insert function		7.49	18.26
	Sequence search		1.17	1.4
	Random search	with index	18.12	
		without index	286.34	3305.15

5. CONCLUSION

This paper reviews the mobile computing with embedded databases. The focus is on the Point-to-Point mobile embedded database. Four embedded databases are explored. These are Perst Lite, PointBase, db4o and RMS. Both the Perst Lite and RMS performance are evaluated. The evaluation covered evaluating the performance of the Insert function, Sequence Search and Random Search. Perst
Lite reflects good performance and supersedes RMS in all of the above functions.

6. REFERENCES

[1] Mobile and World, "Subscriber statistics end Q1 2007," 2007.

[2] S. Ravi, M. Chathish, and H. Prasanna, "WAP AND SMS BASED EMERGING TECHNIQUES FOR REMOTEMONITORING AND CONTROL OF A PROCESS PLANT," in *ICSP'04. 2004 7th International Conference*, 2004, pp. 2672-2675.

[3] K. Hung and Y. Zhang, "Implementation of a WAP-Based Telemedicine System for Patient Monitoring," *Information Technology in Biomedicine,* vol. IEEE Transations on vol. 7, pp.101-107, June 2003.

[4] M. Nikolova, F. Meijs, and P. Voorwinden, "Remote Mobile Control of Home appliances," *Consumer Electronis,* vol. 49, pp. 123-127, Feb 2003.

[5] S. Rahman and S. Bhalla, "Supporting Spatial Data Queries for Mobile Services," in *ACM International Conference on Web Intelligence*, 2005.

[6] V. Lee, K. Lam, and W. Tsang, "Transaction Processing in

Wireless Distributed Real-time Databases."

[7] C. Yang and M. Kou, "Remote Monitoring and Control of PC Clusters Using Mobile Phones with J2ME," 2005.

[8] J. Nichols and B. Myers, "Controlling Home and Office Applances with Smart Phones," *PERVASIVE computing,* pp. 60-67, 2006.

[9] F. Siegemund and C. Florkemerier, "Interaction in Pervasive Computing Settings using Bluetooth-Enabled Active Tags and Passive RFID Technology together with Mobile Phones," in *The First IEEE International Conference on Pervasive Computingand Communications*, 2003.

[10] G. Nikolakopoulos, M. koudourakis, and A. Tzes, "An Integrated System based on WEB and/or WAP Framework for remote Monitoring and control of Industrial Processes," in *International Symposium on Virtual Environments, Human-Computer Interfaces, and Measurement Systems,* Lugano, Switzerland, 2003.

[11] McObject, "McObject Releases Perst Lite, an Open Source Embedded Database For Intelligent Devices on Java's J2ME Platform," 2006.

[12] DataMirror, "POINTBASE MICRO: The Ultra-Compact Database for Enterprise Mobility," 2004.

[13] db4objects, "db4o Open Source Object Database."

[14] McObject, "Perst Documentation," 2006.

[15] Maxim, "Getting Started with TINI."