Benefits of Database for Embedded System and IoT Device Manufacturers

An ITTIA White Paper, October 2016

Introduction

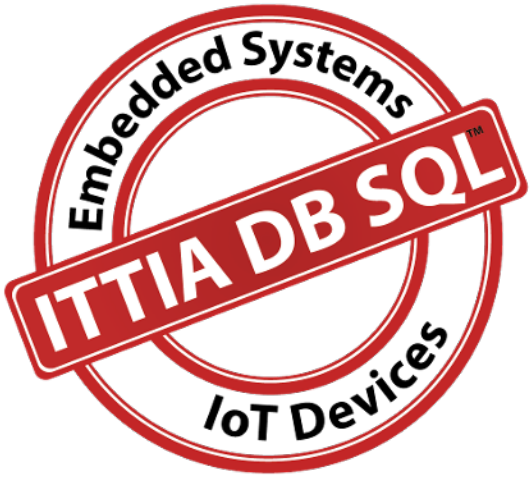
A significant number of connected devices are becoming responsible to capture, store and analyze data for the purpose of making decisions. On each device, information is collected and filtered before it is shared with other systems. Ultimately, the information originating from these devices will be essential to gain insight and inform people's decisions.

The availability of inexpensive, low-power processors with high-capacity solid-state storage media enables the creation of intelligent, connected devices for the Internet of Things. Software developers must write embedded applications that react quickly while continuously collecting information. Data is often relayed from sensors and gateways to other devices, systems, or the cloud, but other data is retained on the device to improve privacy and reduce bandwidth requirements.

Database technology gives software developers an advantage to cope with data growth, protect data integrity, and adapt to new hardware when creating IoT devices. Information appended to a flat text or binary file can only be searched sequentially, whereas a database library can index data inserted in any order and quickly locate specific records even as data volume grows to many gigabytes. With database transactions, multiple records can be updated in a single atomic operation.

When Does Embedded Database Make the Most Sense?

As a product's hardware evolves over time, the data model and data management layer of a database application are easily ported to new platforms. A wide variety of application processors and efficient microcontrollers can benefit from an embedded database. The type of hardware required depends on what database features the software will use and query complexity more than the volume of data, as depicted in the Recommended Minimum Specifications chart.



Recommended Minimum Specifications

Database Use Case	Processor or Microcontroller	Memory	Storage Media
Rapid search on flash media storage	100 Mhz 32-bit	512 KiB	10 MiB
Complex queries and high availability	500 Mhz 32-bit	10 MiB	100 MiB
High performance in-memory storage	1 Ghz 64-bit	1 GiB	1 GiB

Leverage Developer Experience and Industry Standards

The standard application program interface to a relational database for software developers is SQL, the structured query language. SQL statements are used both to modify a database and to gather information for reports. Relational databases follow standard integrity rules to ensure that data is always accurate and accessible. As a result, the data management platform delivers an excellent out-of-the-box experience:

- Standard SQL syntax and familiar terminology simplify design and planning.
- The device is capable of self-management and communicating data with other standards-based products.
- Queries rapidly express complex questions, empowering the device to make decisions on its own.

Development Convenience and Maintainability

The selection of relational database directly effects interoperability of the device and maintainability of the application. The impact of selecting an unsuitable design is difficult software maintenance and more expensive total cost of ownership. Important data management characteristics must be considered when building IoT and embedded applications.

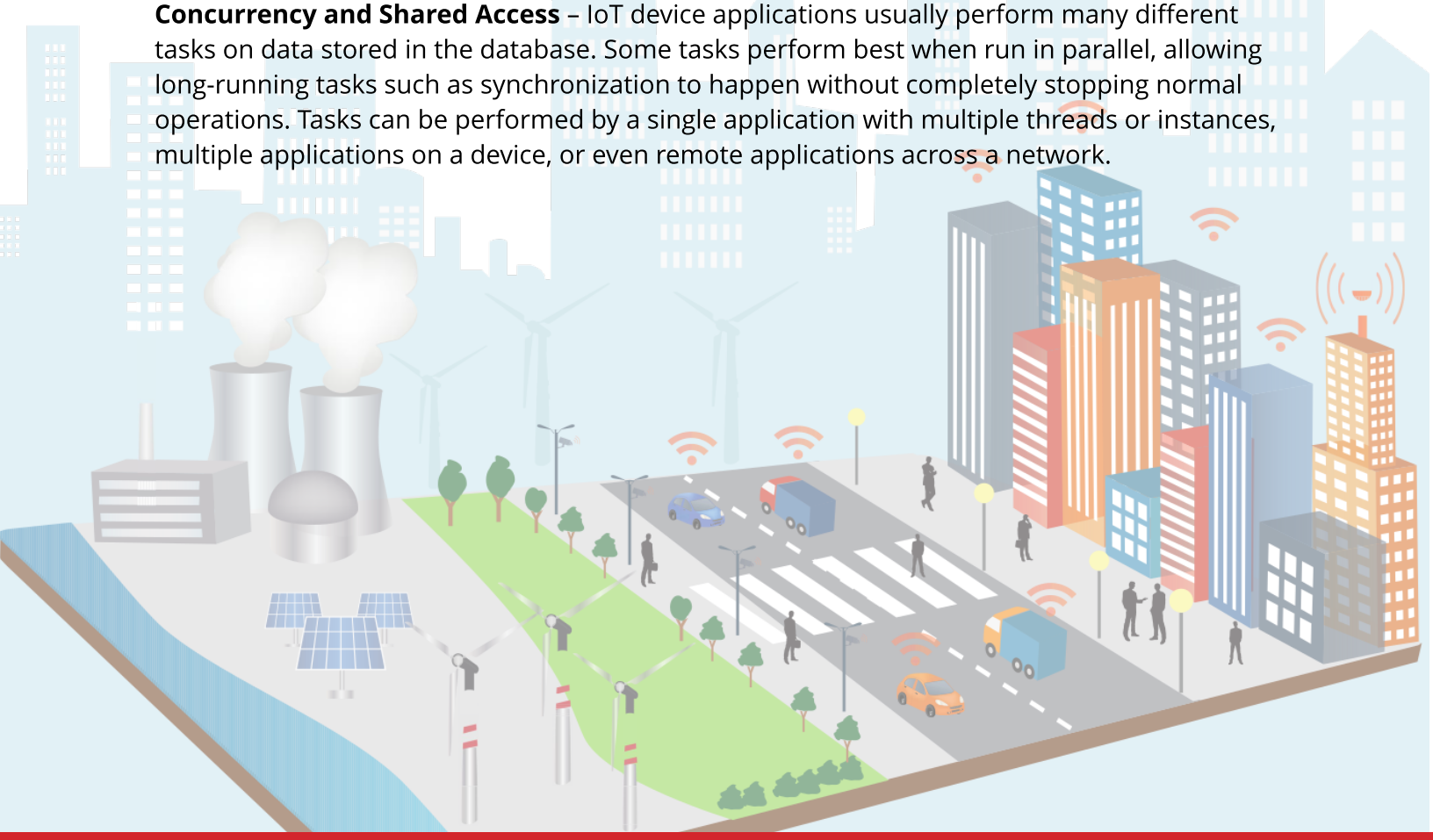
High Availability – When mission-critical data for the Internet of Things device becomes unavailable, the entire system is put at risk. Support for high availability maximizes the protection and availability of data.

Security – To prevent unauthorized access, database files on removable storage media should be encrypted. Remote client/server communications should also use an authentication method that does not reveal passwords over the network, such as SCRAM, or a fully encrypted TLS connection.

Cross-Platform Data Management – A robust data management technology scales to meet the resource constraints of embedded systems. A portable storage file format and platform-independent database API enable application source code to run on any operating system, RTOS, or file system with little or no modification. Even without an operating system, database files can be stored directly on raw flash media in ARM, MIPS, PowerPC, or x86 firmware.

Atomic Transactions – A database uses transactions to protect data from inconsistency and corruption. Each transaction combines one or more database operations into a single atomic commit. If an error occurs during a transaction, any incomplete changes are rolled back, ensuring that the data remains consistent. Even if the error results in a crash, committed transactions are automatically recovered the next time the database is opened.

Concurrency and Shared Access – IoT device applications usually perform many different tasks on data stored in the database. Some tasks perform best when run in parallel, allowing long-running tasks such as synchronization to happen without completely stopping normal operations. Tasks can be performed by a single application with multiple threads or instances, multiple applications on a device, or even remote applications across a network.



Practical In-Memory and On-disk Storage Logic – Performance is critical in embedded and IoT applications. For this reason, a database should provide different storage back-ends for in-memory and on-disk storage so that application developers can optimize performance for each type of record. Hybrid databases, which use a mixture of memory and disk tables, allow applications to seamlessly integrate both storage models.

Replication – Replication is used to share information between devices and across storage media to improve reliability, fault-tolerance, or accessibility. Replication makes it easy to keep devices up-to-date with each other, even when connections are unreliable.

Development tools – Embedded software is developed for specific target hardware, often starting with an off-the-shelf development board before moving to custom hardware. The operating system, relational embedded database, graphical user interface, compiler, file system, and other software components must all work together to support the target platform.

Development Challenges – Since the hardware in an Internet of Things device is usually limited in some way, the database library should be configurable to allow unused features and components to be excluded from the final firmware image.

Benefits and Intelligence Embedded with ITTIA DB SQL

Powerful embedded hardware makes sophisticated data processing possible, but the data must be stored in an efficient and accessible form. ITTIA DB SQL database software enables applications to perform SQL queries that search, analyze and report on information stored on devices with limited memory requirements. With close attention paid to standards during the design and implementation of ITTIA DB SQL, software developers have many options to interoperate with other systems.

ITTIA DB SQL offers superior interoperability and maintainability for medical devices, industrial automation, automotive, and other embedded systems. ITTIA DB SQL provides an easy path for the device to collect, process, and intelligently respond to real-world data and events. Ensuring accuracy and precision is very important before sharing that information.

With two different storage models, on-disk and in-memory, ITTIA DB SQL is able to meet the performance and durability needs of specific applications. Several locking models are also available to balance between high and low concurrency requirements.

Data Management and Connectivity

ITTIA DB SQL offers:

- Embedded library design
- Data modeling with relational tables
- Performance
- Predictable memory management
- Portable file format and API
- Multiple shared access models for concurrent transactions
- Remote access for read and write transactions
- Database replication to efficiently distribute data
- High availability across embedded systems, devices, and the cloud

Case Study 1

As an embedded database library, ITTIA DB SQL is invisible to end-users and requires no database administration. This lightweight technology provides strong concurrency with both multi-process and multi-threaded capability to the device's applications. Embedded developers who may be concerned about device constraints can now experience the freedom and flexibility typically reserved for enterprise databases with features such as client/server for remote access and replication for high availability.

Wasserbauer, a leading robot manufacturer, created industrial automation systems embedded with ITTIA DB SQL on Linux. The reliable storage engine in ITTIA DB SQL enabled its robots to monitor animal health by capturing important information and make decisions based on a large volume of collected data.

Markets

Industrial Automation

Medical Devices

Automotive

Building Automation

Solar/Energy

Consumer Electronics

ITTIA DB SQL can be used in bare metal firmware, on multitasking real-time operating systems, and within a virtual address space. It is suitable for low-footprint applications and is easily ported to new storage media, processors, C/C++ compilers, and operating systems.

Support Platforms:

- Embedded Android
- Linux
- Windows
- QNX
- ThreadX
- µC/OS Family
- INTEGRITY
- VxWorks
- Nucleus
- Custom Firmware

Minimize Total Cost of Ownership

By integrating an embedded database library, device manufacturers reduce development time and cost, especially the overall cost of maintenance. When ITTIA DB SQL is embedded in a device, the added value includes:

- Deliver a complete out-of-the-box development platform
- Empower developers with full relational data management
- Offer excellent ease-of-use
- Exceptional performance
- Much shorter development cycle
- Greatly reduced time to market
- Tremendous savings in development and cost

Case Study 2

The on-board computers in many solar power inverters collect a substantial amount of information about the behavior of the system and maintenance status. Processing that information in a local ITTIA DB SQL database reduces bandwidth requirements and improves data management.

When PV Powered recognized the importance of data management for their solar inverter products, they evaluated various databases and alternatives before selecting ITTIA DB SQL as the best fit. Great performance, ease of use, SQL standards, and the ability to share data with other systems lead PV Powered to select ITTIA.

Summary

The Internet of Things has been called the next Industrial Revolution and it will impact the way all businesses and consumers interact with the physical world. The forecast indicates that there will be 34 billion devices connected to the internet by the year 2020.

ITTIA has had the privilege of supporting customers building applications such as medical devices, industrial automation, portable data terminals, consumer electronics, network appliances, and automotive systems. ITTIA technology enables IoT and embedded developers to leverage enterprise data management features on embedded systems and devices across their product lines, whatever operating systems and architectures are used.

Engineers at ITTIA are ready to assist device manufactures to build prototype applications and case studies for specific industries and scenarios. With decades of experience in database design and implementation, the ITTIA engineering team shares its collected knowledge and insight openly with customers and partners, saving them significant development time and cost.

The reliable, scalable, and cost effective data management and connectivity solutions from ITTIA deliver an opportunity to tackle complex data management challenges on the Internet of Things and empower embedded system manufacturers to inject intelligence into new devices.

Disclaimer

Information in this document is provided solely to enable system and software implementers to use ITTIA products. No express or implied copyright license is granted hereunder to design or implement any database management system software based on the information in this document.

ITTIA reserves the right to make changes without further notice to any products described herein. ITTIA makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does ITTIA assume any liability arising out of the application of or use of any product, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Statistics and parameters provided in ITTIA white papers and data sheets can and do vary in different applications and actual performance may vary over time. All operating parameters must be validated for each customer application by customer's technical experts. ITTIA and the ITTIA logo are trademarks or registered trademarks of ITTIA L.L.C. in the U.S. and other countries. All other product or service names are the property of their respective owners.

 References in this document to ITTIA products and services do not imply that ITTIA intends to make them available in every country.