QuantQuote
historical intraday stock data

QuantEDGE

Overview

September 2011

Document Number: 3062-1

Contents

Revision History

Version	Description	Date
001	Initial release	September, 2011

QuantEDGE – Overview

Summary

QuantQuote's QuantEDGE libraries are a C++ toolkit designed to facilitate data I/O with QuantQuote's proprietary compression format. By default, QuantQuote second and tick resolution datasets come in QuantEDGE binary files which reduce disk space usage by up to 92% compared to raw text format.

Advantages

With the advent of multi-core processing, disk I/O has become one of the most significant bottlenecks for high performance computing. Especially for tick data, which can approach up to 100GB per day for uncompressed quote data, storing and processing files can be a significant computational challenge. QuantEDGE reduces file sizes by an order of magnitude allowing for extensive storage cost savings.

QuantEDGE does decompression on the fly entirely in memory for maximum performance. No temporary files are created at any time in the process so there are no extra read or write operations. Because of the smaller filesizes and binary format, reading QuantEDGE files is usually even faster than reading an uncompressed text file, despite the added overhead of decompression.

QuantEDGE libraries are written in C++ and completely self contained, no external libraries need to be present or installed on the system. QuantEDGE code is platform independent and should work on all Unix/Linux/OSX distributions. QuantEDGE has also been used successfully on Windows platforms even though QuantEDGE is not officially supported on Windows (please contact sales@quantquote.com for more details). A standalone command line QuantEDGE application is also available for bulk decompression of QuantEDGE files into regular text files.

Usage

QuantEDGE can be seamlessly integrated into any C++ program by including the correct header files and linking the libraries. It is also fully compatible with QuantQuote's innovative TickMAP framework and TickMAP is automatically incorporated into QuantEDGE. Sample codes and Makefiles are available through our customer support page or by contacting us at support@quantquote.com).

Once the appropriate header and library files are installed, QuantEDGE files can be read by simply calling the appropriate function such as:

```
get_second_data(std::string permtick, int date, std::vector<struct> &v);
```

where permtick is the TickMAP permanent ticker, date is the requested date, and vector<struct> v is a STL vector that will be filled with the data from the requested trading day. Each element of the vector (a struct) represents a different tick and the struct could be something like:

```
struct SecTradeStruct
{
   int time, open, high, low, close, volume;
   bool suspicious;
};
```

Thus, the volume of the first tick could be extracted using the following code:

```
v.at(0).volume;
```

Performance Testing

Comparative performance tests were carried out to compare the speed of the QuantEDGE libraries versus STL C++ in loading data. Tests were done on a Dell PowerEdge R710 server with the following specs:

Dell 11G R710
2x Intel Xeon E5620 Processors
12x2GB Triple Channel DDR3 Memory @ 1066Mhz
6x500GB Seagate Constellation ES (ST3500514NS) in RAID 10 on Dell PERC 6/I
RHEL 5.8

The input trade data has 352,968 entries. QuantEDGE was tested by loading the QuantEDGE format file using load_file() from the QuantEDGE library. STL C++ was tested by loading a CSV file containing the same information into a STL std::vector <std::vector <string> > to replicate the structure available in the QuantEDGE std::vector<struct>. The reading of the CSV file was done using getline(). The same level of compiler optimization was used in both codes – both were compiled using g++ 4.1.2 with the –O3 flag.

From a filesize perspective, the QuantEDGE compressed binaries stored the data using 841,363 bytes (841Kb) compared to 17,321,136 bytes (17.3MB) for a CSV file, a compression factor of over 95%. The read results averaged over 10 tests are shown below in Figure 1. QuantEDGE has a clear advantage.

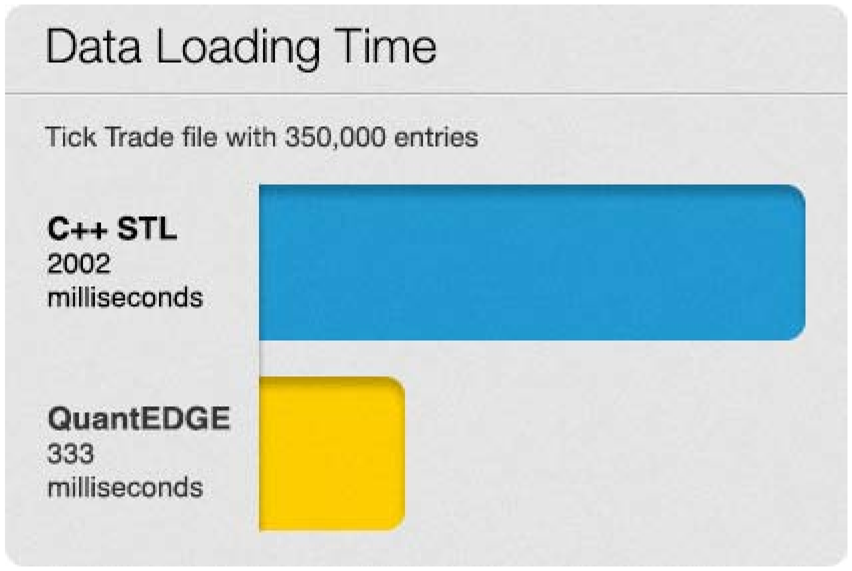

Figure 1: QuantEDGE performance compared to STL C++ with uncompressed text input.