# High frequency data with NIO and The DBReader Framework

# Basic NIO

- NIO (new input output) is Java's fast data reading / writing library
- It gives us super fast access to raw binary data
- Obviously, speed is very useful to us in the context of the simulation infrastructure we've been talking about

# NIO – Reading from channel to buffer

- For example, in the case of a read (code on next slide):
  - We fill a *ByteBuffer* with raw binary data from an open file channel
  - As we fill the buffer, its internal pointer is incremented
  - When we're ready to process data from this buffer, we call its *flip* method to return its internal pointer to the beginning of the buffer
  - We can access that data using convenience methods of *ByteBuffer* such as *getByte*, *getInt*, *getFloat*, and others
- The same functionality is available for writing

# Reading example

```
RandomAccessFile aFile = new RandomAccessFile( "data.txt", "rw" );
FileChannel inChannel = aFile.getChannel();
ByteBuffer buf = ByteBuffer.allocate( 48 );
int bytesRead = inChannel.read( buf );
while ( bytesRead != -1 ) {
  System.out.println( bytesRead );
  buf.flip();
  while(buf.hasRemaining()){
    System.out.print((char) buf.get());
  }
  buf.clear();
  bytesRead = inChannel.read(buf);
}
aFile.close();
```

# From *ByteBuffer* documentation

| | |
|---|---|
| abstract char | **getChar**()<br>Relative *get* method for reading a char value. |
| abstract char | **getChar**(int index)<br>Absolute *get* method for reading a char value. |
| abstract double | **getDouble**()<br>Relative *get* method for reading a double value. |
| abstract double | **getDouble**(int index)<br>Absolute *get* method for reading a double value. |
| abstract float | **getFloat**()<br>Relative *get* method for reading a float value. |
| abstract float | **getFloat**(int index)<br>Absolute *get* method for reading a float value. |
| abstract int | **getInt**()<br>Relative *get* method for reading an int value. |
| abstract int | **getInt**(int index)<br>Absolute *get* method for reading an int value. |
| abstract long | **getLong**()<br>Relative *get* method for reading a long value. |

Some of the methods for interpreting raw binary data

# The DBReader framework

- In a previous lecture, we described a strategy for reading data from files that may differ in format with time stamps that may not necessarily line up
- That approach is implemented in the DBReader framework, the code for which has been uploaded to the group web site
- The framework is comprised of DBManager and three interface classes – I_DBClock, I_DBReader, and I_DBProcessor

# DBManager

- This is the class that manages reading data from potentially many DBReaders
- First the list of readers is sorted by their next time stamp – the time stamp of the next single record that will be read from these readers – so that the reader at the top of the list is chronologically first in line for reading
- The manager then obtains the next time stamp (sequence number) it should read up to from its clock
  - If we want to read every record, the clock would simply return the time stamp obtained from the next reader. However, the clock is useful in reading data in a less granular way – for example, in five minute buckets

# DBManager (contd)

- The manager then asks all readers to read all records up to and including the time stamp it obtained from the clock
- When it encounters the first reader that has a time stamp higher than the time stamp obtained from the clock, it stops the reading cycle and starts the processing cycle
- In the processing cycle, iterates through its processors – which, in our case, are simulations – passing to each its list of readers
- The readers are then sorted again, a new time stamp is obtained from the clock, and the loop continues.

# Advantages

- DBReader doesn't know anything about the actual file types of the readers
- It works entirely with the interfaces previously mentioned, which are relatively simple
- The readers implement these interfaces so they can be easily incorporated into the DBReader framework
- For the sake of example, I have uploaded the concrete implementations of TAQ quotes and trades DBReaders and related files