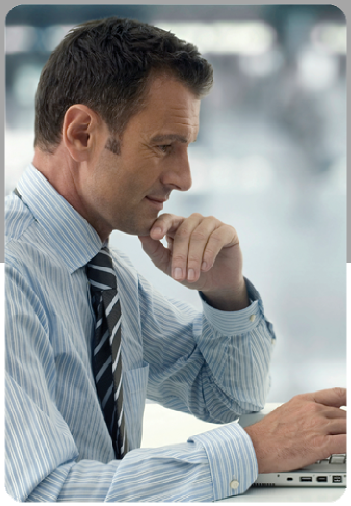

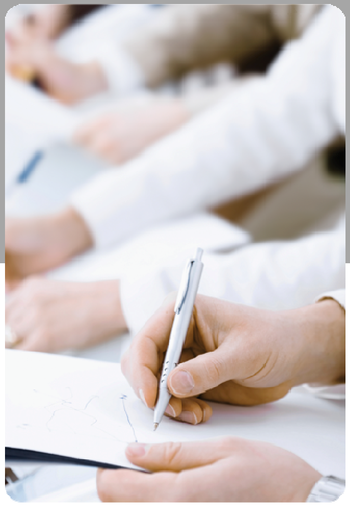

A quick note on threading

Jasper van Baten, AmsterCHEM
Richard Szczepanski, KBC/Infochem

September 2013

Changes in the Multiflash CAPE-OPEN DLL, 2013

- Enhanced multithreading support
  - Inspired by some interesting findings at BASF
- x64 support
  - Inspired by x64 support of the Multiflash core

Two ways of utilizing all nodes on a given computer:

- Multithreading
  - Thread safety issues
  - Performance bottlenecks

- Parallel processing
  - Separate process spaces, all works well
  - Inter-process communication may be limiting

What is the big deal with multithreading?

- Many ‘legacy’ implementations are not thread safe
  - Common blocks
  - Global variables
  - Static variables
  - Shared resources

- But do Apartment Threaded COM objects require this?
  - Yes!

KBC + Infochem

Three approaches:

- What was done in the past
  - Shield access to the unsafe DLL using synchronization
- What is done now:
  - Create only a single property package from each unsafe DLL
- What will be done in the future
  - Eliminate the unsafe factors from the DLL

KBC + Infochem

- What was done in the past
  - Shield access to the unsafe DLL using synchronization

Existing methods:

- MutEx (Mutually Exclusive) objects
  - Can be named, therefore can work across processes
- Critical Sections
  - Local, but faster

Clearly thermodynamic calculations cannot be performed concurrently

KBC + Infochem

- What is done now:
  - Create only a single property package from each unsafe DLL

No 2 DLLs with the same name can be loaded.

Create a copy of the DLL in the temp folder with unique name.

Dynamic linking: LoadLibrary, GetProcAddress

Initial delay due to loading DLL. Bigger memory footprint

KBC + Infochem

- What will be done in the future
  - Eliminate the unsafe factors from the DLL

This requires removing all COMMON blocks. F77 not very suitable. A much bigger project.

What cannot be done?

Heap memory allocations are synchronized.

No two threads can simultaneously allocate memory.

This impacts multithreading performance.

As long as COM is used, this cannot be helped. New object model should take this into account.

Conclusions

- Multi-threading improvements are customer driven
- The current implementation is intermediate
- The final answer is thread safe core libraries

- x64 is now supported