Parallel Programming Done Right with OTL and PPL

Primož Gabrijelčič
About me

- Pascal programmer since 1984 (HiSoft pascal on ZX Spectrum)
- First contact with Borland: Turbo Pascal 3 (on CP/M)
- Programming highly responsive 24/7 applications since 1997
- Writer: The Delphi Magazine, Blaise Pascal, Monitor (Slovenia)
- Blogger: http://thedelphigEEK.com
- Contact me: http://primoZ.gabrijeLcic.org
Multithreading
Multithreading is hard

“New programmers are drawn to multithreading like moths to flame, with similar results.”

- Danny Thorpe
Solution

- Extract all hard parts into a boilerplate code.
- Test it. Test again. Test repeatedly.
- Reuse as much as possible.
- Test again. Don’t stop testing.

– or –

- Use existing library.
- Continue testing.
When to do it?

- **Unblocking GUI**
  - Long calculations
  - Synchronous APIs
    - File system
    - (Serial) communication

- **Speeding up the computation**
  - Faster calculation
    - More/less appropriate tasks (algorithms)
  - Serving more than one client at once
Patterns
Adapt algorithm to the pattern

- Don’t write the code for your algorithm
- Decompose the algorithm into patterns
  - Use those patterns in the code
- When everything fails, go low-level
- Tasks first, threads last
Frameworks

- **PPL**
  - Parallel Programming Library
  - XE7+, all platforms, RTL license
  - patterns: For, Future, Join

- **OTL**
  - OmniThreadLibrary
  - 2009+ (patterns), 2007+ (tasks), Windows (VCL/console/service) only, OpenBSD license
  - patterns: Async[/Await], Background worker, For, Fork/Join, Future, Join, Map, Parallel task, Pipeline
  - [http://www.omnithreadlibrary.com/](http://www.omnithreadlibrary.com/)
Dish of the day

- Async/Await
  - Fire asynchronous tasks

- Future
  - Execute long calculation in background

- For
  - Use all of available CPUs when processing large data

- Map
  - Converting data in parallel

- TimedTask
  - Just like TTimer, but running in a thread
Async/Await

Async(code1). Await(code2)

code 1

code 2
Future

Future (code)

.Value

Result

code
For(first, last, code)
Map
TimedTask

TimedTask. Every(interval). Execute(code)
Other OmniThreadLibrary patterns
ForEach
ParallelTask
Join
Pipeline
Fork/Join
Get more information

- [http://www.omnithreadlibrary.com/tutorials.htm](http://www.omnithreadlibrary.com/tutorials.htm)
- “Parallel Programming with OmniThreadLibrary”
  - [https://leanpub.com/omnithreadlibrary](https://leanpub.com/omnithreadlibrary)
Keep in mind
Important Facts We Learned Today

- Don’t write boilerplate code – use patterns
- Be careful when accessing shared data
- Never access the GUI from a background thread!
Q & A