

21 May 2008, Distributed Computing Workshop

Dr Oleg Soloviev

oleg.soloviev@econophysica.com

Developing Distributed Financial Applications

Econophysica Ltd

Econophysica Ltd

- A small team in London and a larger development team in Tomsk, Russia (Siberia).
- Serving major financial institutions for 8 years.
- Projects on
 - Financial derivatives
 - Risk management
 - Algorithmic trading

Agenda

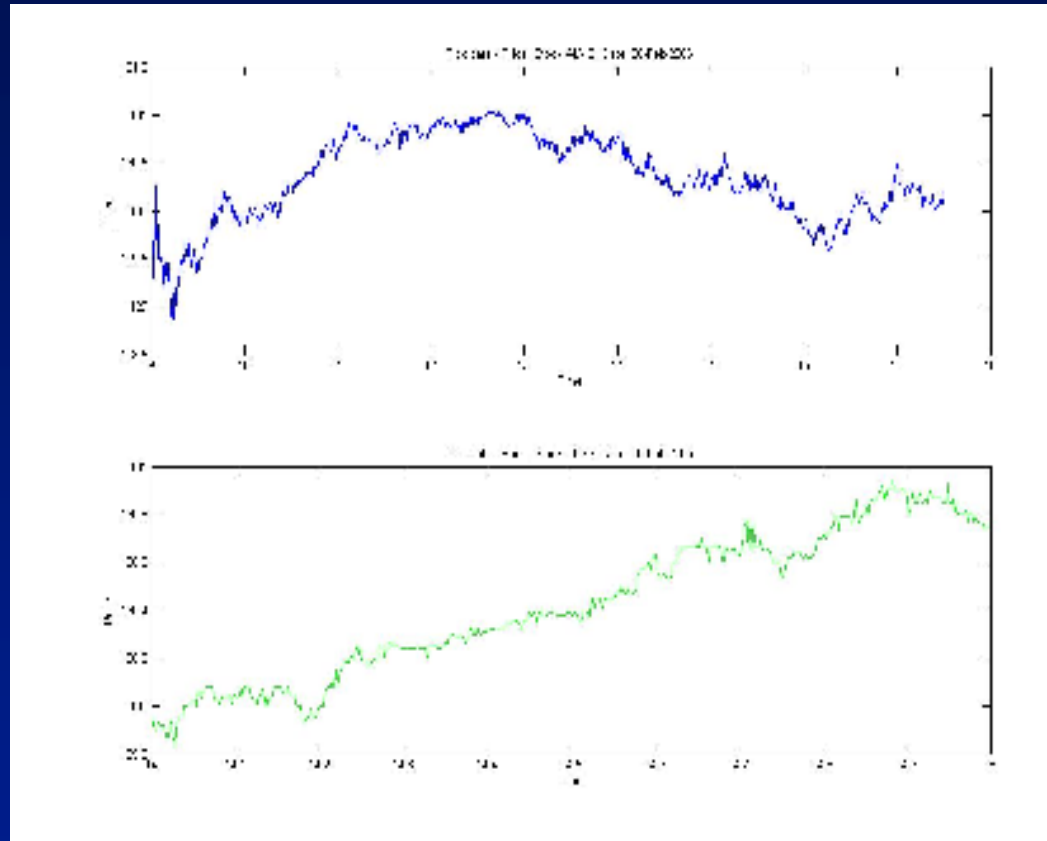
- Computational Finance (Main players)
- Examples of distributed financial computations
- Which technologies appear to be suitable?
- Problems
- Conclusion

Computational Finance

- Investment banks
 - Derivative desks
 - Pricing models
 - Algo-trading
 - Execution algorithms
 - Forecasting algorithms
 - Risk-management
- Hedge funds
 - Pricing Models
 - Execution/Forecasting algorithms
- Brokers
 - Execution algorithms
- Insurance companies
 - Extreme evens forecasting
- Regulating authorities

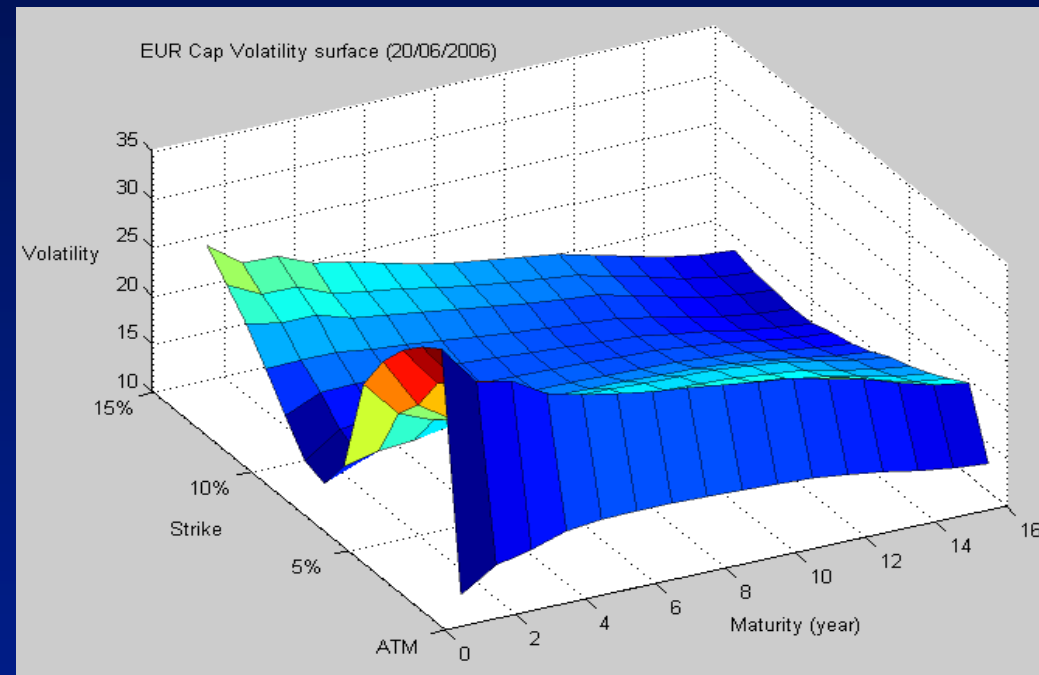
Example: Stock Movement Forecasting

- We have to analyse stock movements over hundreds of days. One day involves tens of thousands of data points. One day may take several minutes to analyse.
- Then we run tens of different forecasting algorithms for tens of different stocks.



Example: Option Pricing

- Calibration: repetitive search for global minimum from different starting points. On one computer, this can take up to an hour.
- Monte-Carlo simulations. This involves hundreds of thousands of repetitive calculations.



Example: Cash Flow at Risk

- Companies with massive financial operations (e.g. telecom companies) have to routinely optimise their cash flows.
 - Thousands of scenarios
- Financial regulating authorities have to monitor hundreds of financial activities taking place at the same time.
 - Change point detection
 - Pattern recognition

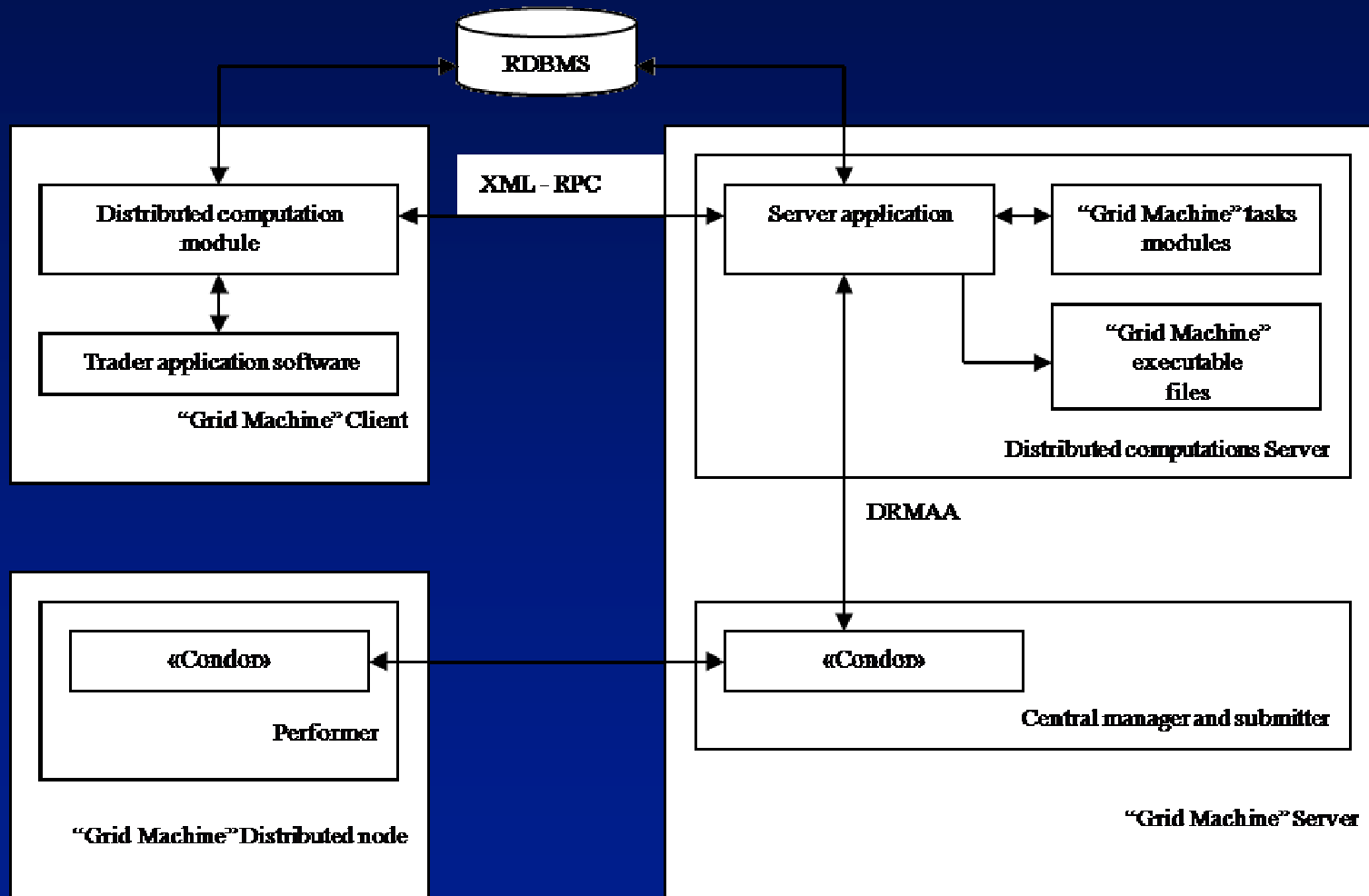
Distributed Technologies: Condor->gLite

- University of Wisconsin-Madison Project (13 years)
 - High throughput computing
 - Distributed resource management system
 - GRID-style computing environment
 - Cross-platform
 - C
- Fully compatible with gLite workload management system.
- Ideal for locally distributed computing.
- GPL (free license)

Current Project

- Development of financial mathematical algorithms for distributed computing.
- Local Grid machine (based on Condor).
- In combination with gLite -> Global Grid.
- Significant speed increase for calibration procedures (of fixed-income multi-factor BGM models) on LGM.
- American Monte-Carlo under development.

EcoCondor Grid Machine Architecture



Other GRID Technologies

- Rack-mounted clusters (e.g. by Dell)
- Utility computing engines (e.g. by Sun)
- Portable batch systems.
- Lots and lots of other hardware, software solutions.

Common Issues with Distributed Computing

- Algorithms written for single-processor computations cannot be automatically distributed. Huge software libraries have to be re-written.
- There is no automatic compatibility between different distributed algorithms.
- Certain tasks cannot be parallelized.
- Grids may increase maintenance costs.

Conclusion

- High performance computing is a must for many financial applications. In many areas business growth depends vitally on computational performance.
- GRID computing, in general, and gLite, in particular, represent a unique opportunity for knowledge transfer between traditionally very reluctant financial institutions and academia. The main reason is that academics can talk to financial IT people who are essentially ex-academics themselves.
- Also distributed computing on its own does not give away any sensitive financial information except for the performance level.
- On implementation side, it would be a huge benefit to come up with cross-technology distributed solutions.
- Big thanks to STFC for a very timely support of a technology transfer project on distributed computing.