2^{nd} HIPERFIT Workshop 2011

Mathematical Finance meets Programming Languages and Systems

Call for Participation

December 1-2, 2011
Copenhagen, Denmark

Information online: `http://hiperfit.dk/workshop-12-2011.html`

HIPERFIT

Today, the financial sector faces daunting computational challenges, involving both an increasing demand for performance and higher transparency requirements. In addition, time-to-market is more and more important for financial applications, both by commercial software developers and in-house. The HIPERFIT[1] research center at the University of Copenhagen aims to solve these simultaneous challenges of high transparency, high computational performance and high productivity in an integrated approach of declarative, domain specific and high-level functional programming languages.

HIPERFIT fosters cooperation between researchers from three different university departments (Computer Science, Mathematics, and Physics), and major industrial partners. Researchers and practicioners contribute cutting edge knowledge from their respective fields to bear on computational and algorithmic problems that transcend their own field of expertise.

2^{nd} HIPERFIT **workshop, December 1^{st} and 2^{nd}, 2011:**

In view of our interdisciplinary spirit, HIPERFIT organises biannual workshops to encourage experts' exchange and discussions. Our 2^{nd} workshop, which will take place in Copenhagen on December 1-2, features 10 invited presentations by our academic partners and invited guests, combined with two special talks (DIKU talk and COPLAS talk series).

Our eleven speakers are international researchers from the different HIPERFIT areas, ranging from modern mathematical finance, via programming language technology and modern approaches to parallel programming, to high-performance systems. Interested researchers and practicioners of all related fields are invited to participate in this exciting event in Copenhagen, to meet and discuss with our international speakers and guests.

[1] HIPERFIT is the research center for Functional High-Performance Financial Information Technology at the University of Copenhagen, funded by the Danish Council for strategic research. In cooperation with CFIR (Copenhagen Finance IT Region), the center joins researchers in mathematical finance, programming languages, and systems, and four major Danish banks (Danske Bank, Nordea, Jyske Bank, Nykredit) and two companies in Financial Software (SimCorp, Lexifi). For more information see `http://hiperfit.dk`.

Organisation and Venue:

The HIPERFIT 2011 workshop will be held at the University of Copenhagen, North Campus, in the HCØ building – next to the Department of Computer Science (DIKU). A map and travel information can be found on the department web pages `http://www.diku.dk/english/contact/findvej_kopi_kopi/`.

There will be no published proceedings, the meeting is intended to be an interdisciplinary forum for discussion and networking. For more information, including presentation abstracts and the most recent schedule information, please visit the workshop web pages `http://hiperfit.dk/workshop-12-2011.html`.

Participation is free, but for organisational purposes, a workshop registration is required. To register, please fill out the web form accessible from the workshop pages.

Looking forward to seeing you in Copenhagen!
Jost Berthold and Fritz Henglein – HIPERFIT Research Center

Thursday 1st December:

- Claudio Albanese, Imperial College/Global Valuation Ltd
  High Throughput Portfolio Processing on Heterogeneous Boards
- Enrico Biffis, Imperial College
  Collateral Flows, Funding Costs, and Counterparty-Risk-Neutral Swap Rates
- Don Syme, Microsoft Research Redmond
  Strongly-Typed Programming in the Information Rich World
- Niels Nygaard, University of Chicago
  Optimal Discretization of Random Variables and Option Pricing Using F#
- Satnam Singh, University of Birmingham
  Data-parallel GPU/FPGA Programming with Accelerator
- Manuel Chakravarty, University of New South Wales
  Shared Data Structures in Nested Data Parallelism
- Mary Sheeran, Chalmers University of Technology
  (DIKU talk) Domain-Specific Languages: Past, Present and Future

Friday 2nd Dec.:

- Robert Harper, Carnegie-Mellon University
  On Teaching Parallelism in Introductory CS Courses at CMU
- John Reppy, University of Chicago
  Portable Parallelism in Diderot
- Clemens Grelck, University of Amsterdam
  Declarative Array Programming with Single Assignment C
  - Language Design and Compiler Technology -
- Thomas Neumann, Technical University Munich
  HyPer-sonic Combined Transaction AND Query Processing
- Robert Harper, Carnegie-Mellon University
  (COPLAS talk) Canonicity for Two-Dimensional Type Theory

Thursday 1st December

Claudio Albanese, Imperial College/Global Valuation Ltd

High Throughput Portfolio Processing on Heterogeneous Boards

9:00

Heterogeneous computing boards combining 2-8 CPUs and 4 or more GPUs open new scenarios and pose new challenges to financial modelling. On the one hand, capable boards allow one to load entire bank portfolios on a single node and bypass the network bottleneck. On the other, the memory bottleneck is extremely tight and needs to be overcome at the algorithmic level and when designing the software architecture.

We demonstrate the modelling strategies at which we arrived with a case-study involving coherent global simulations of global portfolios of netting sets.

Claudio Albanese graduated at ETH Zurich and held full time professorships at the University of Toronto and Imperial College. He currently leads Global Valuation Limited and teaches at King's College London

Enrico Biffis, Imperial College

Collateral Flows, Funding Costs, and Counterparty-Risk-Neutral Swap Rates

9:45

We study the impact of bilateral default risk and collateral rules on the marking to market of swaps, and show how swap rates must be determined endogenously from the collateral flows and funding costs associated with the marking-to-market procedure. We consider the possibility of using stochastic approximation algorithms to compute equilibrium swap rates, and examine the impact of asymmetric credit quality and different collateral and close-out rules on the pricing of OTC instruments.

The talk is based on joint work with Damiano Brigo (King's College) and Lorenzo Pitotti (Algorithmics/Imperial College London).

Enrico Biffis is Assistant Professor in Actuarial Finance at Imperial College Business School. His main research interests lie in the areas of insurance and risk management, with a focus on asset-liability management, valuation of insurance and pension liabilities, optimal design of risk transfers for long term risks and catastrophe exposures. He has written extensively on market-consistent accounting standards for insurers, longevity risk management and securitization. Prior to joining Imperial College London in 2007, Enrico held positions at Bocconi Milan, Association of British Insurers, and Cass Business School. Enrico holds degrees in Statistics (BSc & MSc), Actuarial Management (MSc), and Mathematics for Economic Decisions (PhD).

Don Syme, Microsoft Research Redmond

Strongly-Typed Programming in the Information Rich World

11:00

The world is information rich, but why are our programming languages so information sparse? Especially our strongly typed functional languages, given that they are such good information processing environments! In this talk, I'll give an overview of the challenges of strongly-typed information-rich programming against web data markets, semantic-web ontologies, content-management systems and databases. Are information spaces just libraries? Can we gives types to "everything", and if so, should we? We'll demonstrate what F# 3.0 offers in the area of Information Rich Programming, and look at how information-richness makes us reconsider language design and evaluation, as well as touching on the research problems and engineering challenges that come with the territory.

Don Syme is a Principal Researcher at Microsoft Research, Cambridge, researching for Microsoft in programming languages, especially the design and implementation of F#, but also C#, Visual Basic and other .NET languages.

As a researcher, his area is programming language design and implementation, with emphasis on making functional languages that are simpler to use, interoperate well with other languages and which incorporate aspects of object-oriented, asynchronous and parallel programming. He is interested in programming language perspectives on type inference, concurrency, reactivity, pattern matching and language-oriented programming. In the past he has worked in formal specification, interactive proof, automated verification and proof description languages. Don holds a PhD from the University of Cambridge and is a member of the WG 2.8 working group on functional programming.

Niels Nygaard, University of Chicago

Optimal Discretization of Random Variables and Option Pricing Using F#

11:45

It is well known that probability distributions form a monad, an elegant formulation of this is to express the functor as a monad transformer of the Writer monad by the List monad where the Writer monad is defined by the monoid of probabilities i.e. real numbers in [0,1]. We extend this construction to define a Random Variable monad as a monad transformer of the Writer monad with monoid $[0,1]\times\mathbb{R}$ where the monoid structure is given by $(p,x)\otimes(q,y)=(pq,x+y)$. The functorial structure of the RandomVar monad is slightly different from the obvious one and is based on the conditional expectation of a Random Variable with respect to a sigma-algebra. We use a Gaussian Quadrature algorithm to discretize continuous Random Variables and apply this to the transform method of pricing vanilla options under stable distributions. We also show how the RandomVar monad makes it simple to build trees to price path dependent options. The algorithms are implemented in F#.

Niels O. Nygaard is Professor of Mathematics at the University of Chicago, the Founding Director of the Financial Mathematics Master's Program and the Stevanovich Center for Financial Mathematics.

Satnam Singh, University of Birmingham

Data-parallel GPU/FPGA Programming with Accelerator

13:15

This presentation will illustrate the design of an embedded domain-specific language for stencil-style computations which can be dynamically compiled to GPUs and multicore SSE3 vector instructions and off-line compiled to FPGA circuits. The system is language independent and raises the level of abstraction of data-parallel programming to achieve an usual level of productivity versus effort. The Accelerator system provides a way to map a single computation onto quite different processing elements; however we show that performance portability is still very difficult to achieve.

Satnam Singh is researching techniques for heterogeneous parallel programming using FPGAs, multicore vector processors and GPUs. His work has focused on the application of high-level concepts from functional programming languages to the design of low-level architectures. He has worked at the University of Glasgow (UK), Xilinx (San Jose, California), Microsoft (Redmond, Washington and also Cambridge, UK), British Telecom (UK), VLSI Technology (Sophia Antipolis, France) and currently holds the Chair of Reconfigurable Systems at the University of Birmingham (UK). In January he will join Google (Mountain View, California).

Manuel Chakravarty, University of New South Wales

Shared Data Structures in Nested Data Parallelism

14:00

It is well known that Blelloch's flattening transformation duplicates shared data structures across multiple parallel operations, thus increasing the work and space complexity of certain classes of programs. Palmer et al. proposed a work-efficient variant of Blelloch's transformation for first-order programs, but unfortunately their approach does not extend to higher-order programs.

In this talk, I will explain the importance of Blelloch's algorithm and the problems created by the duplication of shared structures. Moreover, I will introduce a novel method to achieve a work-efficient transformation for higher-order functional programs and will summarise our experience with implementing this new method in the Glasgow Haskell Compiler.

Manuel M T Chakravarty is an Associate Professor at the University of New South Wales, Sydney. His main research interests are in functional programming languages, novel compiler technology, and parallel programming. He graduated from the University of Karlsruhe and received a doctoral degree from the Technical University of Berlin.
He contributed to Haskell's foreign function interface, the theory and implementation of type families, and the design and realisation of Data Parallel Haskell, an implementation of nested data parallelism in the Glasgow Haskell Compiler. He receives his inspiration from combining theory with practice.

Mary Sheeran, Chalmers University of Technology

(DIKU talk) Domain-Specific Languages: Past, Present and Future 15:00

I have worked on Domain-Specific Languages (DSLs) for all of my research career, and have seen how their popularity ebbs and flows. In hardware design, researchers who work on the use of modern programming languages or, worse still, on "correct by construction" methods almost always end up being ignored. The standard hardware description languages System Verilog and VHDL reign supreme, and we are left with a daunting post-hoc verification problem. We went wrong somewhere!

Maybe things will work out better on the software side? At the Usenix Conference on Domain Specific Languages in 1997, Paul Hudak spoke on "The Promise of Domain-Specific Languages". He argued that "a well-designed DSL should be the ultimate abstraction for a particular application domain, capturing precisely the semantics of an application, no more and no less". In software development, DSLs are hot once again, both in computer science and in software engineering. Research funding is easier to get and industrial collaborators are open to the use of strange specification and programming languages. The Hiperfit Centre in Copenhagen and a well-funded project on Resource Aware Functional Progrmming at Chalmers are examples of this positive trend. This time we have to deliver!

In this talk, I will look at current research in DSLs that may help us to deliver. I will also point to areas where we need to collaborate more effectively if we are to succeed this time. The talk can be viewed as a call to arms, and I am hoping that some members of the audience will answer the call.

Mary Sheeran is Professor in Computer Science at Chalmers University of Technology, Göteborg.
In the early 1980s, she worked on a domain specific language (DSL) for hardware design, based on Backus' FP. She even studied patterns or combinators for building hardware, but failed to call them design patterns. In recent years, she has been working with Ericsson on a DSL for digital

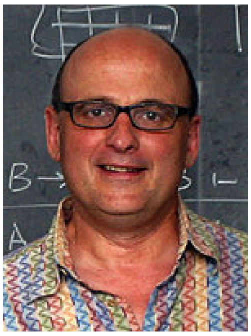
signal processing algorithm design and implementation (in software). Building on this and on their expertise in testing, the Functional Programming Group at Chalmers has recently obtained substantial funding to work on the use of DSLs in Resource Aware Functional Programming. There are many parallels with the HIPERFIT Centre, and so great opportunities for scientific collaboration.

Friday 2nd Dec.

Robert Harper, Carnegie-Mellon University

On Teaching Parallelism in Introductory CS Courses at CMU

9:00



(This might be a shorter talk, as Robert Harper will give a COPLAS talk in the afternoon).

Robert Harper is a Professor of Computer Science at Carnegie Mellon University, where he has been a member of the faculty of the Computer Science Department since 1988. His main research interest is in the application of constructive type theory to the design and implementation of programming languages and to the development of systems for mechanization of mathematics. He is probably best known for his work on the design and implementation of Standard ML, for the introduction of the LF logical framework, and for the development of the concept of a type-directed certifying compiler based on typed intermediate languages.

John Reppy, University of Chicago

Portable Parallelism in Diderot

9:45

Portable parallelism has long been the desire of parallel language designers, but it appears difficult, if not impossible, to achieve in a general-purpose language. An alternative is Parallel Domain-Specific Languages (PDSLs), which, by restricting the programming model, can achieve good performance on a wide range of hardware platforms.

In this talk, I report on Diderot, which is a PDSL for advanced image analysis and visualization algorithms. Diderot supports a very-high-level mathematical computation model coupled with a batch-synchronous parallelism model. Diderot is designed to both enable rapid prototyping of new image analysis algorithms and high performance on a range of parallel platforms. I will give an overview of the design of Diderot and early examples of its use. I will then describe aspects of its implementation, including preliminary work on generating code to run on GPUs.

Diderot is joint work with Gordon Kindlmann, Charisee Chiw, Lamont Samuels, and Nick Seltzer.

John Reppy is a Professor of Computer Science and a Senior Fellow of the Computation Institute at the University of Chicago. He received his Ph.D. from Cornell University in 1992 and spent the first eleven years of his career at Bell Labs in Murray Hill NJ. He has been exploring issues in language design and implementation since the late 1980's, with a focus on higher-order, typed, functional languages. His work includes the invention of Concurrent ML and work on combining object-oriented and functional language features. His current research is on high-level languages for parallel programming, including the Manticore and Diderot projects.

Clemens Grelck, University of Amsterdam

Declarative Array Programming with Single Assignment C - Language Design and Compiler Technology -

11:00

As an array programming language SAC (for Single Assignment C) focusses on multidimensional arrays as the primary data structure. Based on a formal array calculus SAC supports declarative array processing in the spirit of interpreted languages such as APL. This programming style treats multidimensional arrays in a holistic way and is thus quite different from the notion of indexed collections of data in main stream imperative languages. Functions map argument values into result values like in mathematics, rather than being simple subroutines. At the same time, the syntactic similarity with C and the semantic equivalence between the SAC semantics based on context-free substitution and the C semantics based on step-wise modification of state facilitate the conversion of imperative programmers to functional array programming in SAC.

SAC is a high-productivity language for all application domains that deal with large collections of data in a computationally intensive way. At the same time SAC also is a high performance language competing with low-level imperative languages through compilation technology. The abstract view on arrays combined with the functional semantics support far-reaching program transformations. A highly optimised runtime system takes care of automatic memory management with an emphasis on immediate reuse. Last not least, the SAC compiler exploits the state-free semantics of SAC and the data-parallel nature of SAC programs for fully compiler-directed acceleration on contemporary multi- and many-core architectures. Following an overview on the language design, we sketch out selected aspects of the associated compilation and parallelisation technology and report on some comparative runtime experiments.

Dr Clemens Grelck is a Senior University Lecturer at the University of Amsterdam, the Netherlands. Before that, he had academic positions with the Universities of Kiel and Luebeck in Germany and the University of Hertfordshire, United Kingdom. His research interests are in the areas of programming language design, compilation technology and runtime resource management with the joint aim to advance software engineering for parallel systems. He is a key figure behind the functional array language SAC and the stream-oriented coordination language and component technology S-Net.

Thomas Neumann, Technical University Munich

HyPer-sonic Combined Transaction AND Query Processing

11:45

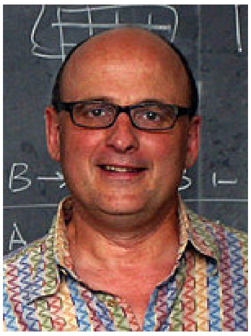
In this presentation we want to convince the community that it is - against common belief - indeed possible to build a main-memory database system that achieves world-record transaction processing throughput and best-of-breed OLAP query response times in one system in parallel on the same database state. The two workloads of online transaction processing (OLTP) and online analytical processing (OLAP) present different challenges for database architectures. Currently, users with high rates of mission-critical transactions have split their data into two separate systems, one database for OLTP and one so-called data warehouse for OLAP. While allowing for decent transaction rates, this separation has many disadvantages including data freshness issues due to the delay caused by only periodically initiating the Extract Transform Load-data staging and excessive resource consumption due to maintaining two separate information systems. We present an efficient hybrid system, called HyPer, that can handle both OLTP and OLAP simultaneously by using hardware-assisted replication mechanisms to maintain consistent snapshots of the transactional data. HyPer is a main-memory database system that guarantees the full ACID properties for OLTP transactions and executes OLAP query sessions (multiple queries) on arbitrarily current and consistent snapshots. The utilization of the processor-inherent support for virtual memory management (address translation, caching, copy-on-write) yields both at the same time: unprecedentedly high transaction rates as high as 100000 per second and very fast OLAP query response times on a single system executing both workloads in parallel. The performance analysis is based on a combined TPC-C and TPC-H benchmark.

Thomas Neumann studied Computer Science from 1997 until 2001 at the Univ. Mannheim, Germany. He received his doctoral degree in 2005 from the same university with a thesis on query optimization. Thereafter, he worked as a senior researcher at the Max Planck Institute for Informatics (MPI) in Saarbrücken. In 2009 he was visiting researcher in the SQL Server group of Microsoft, before being appointed as a Professor at the Techn. Univ. München (TUM) in 2010. His research focus is on database system performance optimization. He covers a broad range of topics from formal and algorithmic work on logical query optimization to database systems engineering. Before joining TUM, he engineered the RDF-3X system at MPI, an open-source high-performance triple store for RDF data.

Robert Harper, Carnegie-Mellon University

(COPLAS talk) Canonicity for Two-Dimensional Type Theory

13:15



Higher-dimensional dependent type theory enriches conventional one-dimensional dependent type theory with additional structure expressing equivalence of elements of a type. This structure may be employed in a variety of ways to capture rather coarse identifications of elements, such as a universe of sets considered modulo isomorphism. Equivalence must be respected by all families of types and terms, as witnessed computationally by a type-generic program. Higher-dimensional type theory has applications to code reuse for dependently typed programming, and to the formalization of mathematics. In this paper, we develop a novel judgemental formulation of a two-dimensional type theory, which enjoys a canonicity property: a closed term of boolean type is definitionally equal to **true** or **false**. Canonicity is a necessary condition for a computational interpretation of type theory as a programming language, and does not hold for existing axiomatic presentations of higher-dimensional type theory. The method of proof is a generalization of the NuPRL semantics, interpreting types as syntactic groupoids rather than equivalence relations.

This talk reports about joint work with Daniel R. Licata.

Robert Harper is a Professor of Computer Science at Carnegie Mellon University, where he has been a member of the faculty of the Computer Science Department since 1988. His main research interest is in the application of constructive type theory to the design and implementation of programming languages and to the development of systems for mechanization of mathematics. He is probably best known for his work on the design and implementation of Standard ML, for the introduction of the LF logical framework, and for the development of the concept of a type-directed certifying compiler based on typed intermediate languages.
