FFMK: A FAST AND FAULT-TOLERANT MICROKERNEL-BASED SYSTEM FOR EXASCALE COMPUTING

Phase 1 Results: Summary
- First L4-based prototype
- Several source-compatible MPI applications ported
- Tested on small island of real HPC cluster
- Gossip scalability and resilience modeled, simulated, and measured
- Erasure-coded in-memory checkpoints with XtreemFS, tested on Cray XC40
- 2 SPPEXA Workshops

Phase 2 (2016 – 2017)

FFMK System Architecture
- L4 microkernel on every node
- Programming paradigms provided as library-based runtimes
- Performance-critical parts of MPI, InfiniBand, and checkpointing run directly on L4
- Non-critical support functionality reuses Linux (e.g., XtreemFS MRC-OSD, MPI startup-control)
- Gossip algorithms disseminate info for platform management
- Linux compatibility via virtualization
- Optional application hints can improve decision making
- GROMEX, COSMO-SPECS+FD4, CP2K, benchmarks, mini apps, ...

Second L4-based Prototype: Decoupled Execution
- Avoids operating system noise by sidestepping Linux
- HPC Applications are ordinary Linux processes, but its threads moved to compute core controlled by L4
- Communication via InfiniBand via direct hardware access
- Linux System calls: Move thread back into Linux, handle operation on service core, return to compute core
- L4 system calls: faster scheduling, threads, memory, ...

Dynamic Platform Management
- Consider CPU cycles, memory bandwidth, and other resources
- Classification based on memory load (“memory dwarfs”) to optimize scheduling and placement
- Prediction of resource usage using hardware counters and application-level hints (e.g., number of particles, time steps)

Fault Tolerance
- Application interfaces to optimize or avoid CAI (e.g., hints on when to checkpoint, ability to recover from node loss)
- Node-level fault tolerance: Multiple Linux instances, micro-rebooting, proactive migration away from failing nodes

Scientific Network (Selection)
Center for Advancing Electronics Dresden (TA Dresden Excellence Cluster)
Highly Adaptive Energy-Efficient Computing (SFB128)
ASTEROID (SPP1620)

Frank Bellas
Karlsruhe Institute of Technology
Gerhard Hader
UNSW, NICTA
Michael Bossmann
Heidelberg University
Dresden Fossendorf

Laxmi Kant V. Kale
University of Illinois at Urbana-Champaign
Vijay Saraswat
IBM Research, Zürich, X10
Eric Van Hensbergen
IBM Research Austin

Yutaka Ishibashi
University of Tokyo
Frank Mueller
North Carolina State University

Related Projects
Ango / Hobbies / mDS
Argonne / Sandia / Intel

SPPEXA: ESSEX / GROMEX
Gerhard Wellein / IceCube

Profs. Alexander Reinefeld
Zuse Institute Berlin
Thomas Steinke
Thorsten Schuett
Florian Wende

Prof. Amnon Barak
Hebrew University of Jerusalem
Amnon Shvish
Ely Levy
Tal Ben-Nun
Alexander Margolin
Michael Sutton

Load Balancing

Prof. Wolfgang E. Nagel
Technische Universität Dresden, ZIH
Matthias Lieber

MPI and Performance Analysis
- VAMPIR: Visualization and Analysis of MPI Resources, Supercomputer 63, X11(69-70), 1996

Critical App
Non-critical App
Linux