1st Generation of CoE

EoCoE - Energy oriented Centre of Excellence

BioExcel - Centre of Excellence for Biomolecular Research

NoMaD - The Novel Materials Discovery Laboratory

MaX - Materials design at the eXascale

ESiWACE - Excellence in SImulation of Weather and Climate in Europe

E-CAM - An e-infrastructure for software, training and consultancy in simulation and modelling

POP - Performance Optimisation and Productivity

COEGSS - Center of Excellence for Global Systems Science

CompBioMed - A Centre of Excellence in Computational Biomedicine

CoE and EsD

General requirements:

- EsDs need to be stable enough to allow production runs
- Clear view on programming models and software stack developments
  - As much as possible, commonly used tools need to be supported (compilers, MPI, OpenMP, numerical libraries, etc.)
  - Novel technologies should provide standard interfaces where possible
- Applications need to be part of EsD design from the beginning
  - Needs significant engagement from the EsD projects for application porting and optimization

Open Questions:

- Support for extreme FLOP **AND** extreme scale data (analytics)
- How sustainable will novel technologies be (and thus the effort to port to them)?
- How will application porting and optimization be funded?
- Support for ensembles/workflows

CoE and SRA-3

Domain CoE: What are the key scientific challenges and what does it take to achieve them?

	BioExcel	CompBioMed	MaX	ESiWACE	E-CAM
HPC System Architecture and Components	Large width vector units, low-latency networks, high-bandwidth and large memory; fast CPU<->accelerator transfer rates, Heterogeneous acceleration, floating-point	Hybrid systems, GPUs, high-bandwidth memory	Scientific challanges: accurate computations requiring more powerful HPC systems, heterogeneous systems designed to optimize end-to-end workflows	High-bandwidth memory, networks, NVRAM	High memory bandwidth, large RAM
System Software and Management	Dynamic (task) scheduling, Support for workflows	Dynamic scheduling, urgent computing	Virtual Machine model supported	Dynamic scheduling, compilers	Cross compiling, archiving tools
Programming Environment	Standardization, portability, task parallelism, fast code driven by Python interfaces	Portability, ease of scale out, MPI extensions	support new MPI and OpenMP standards + new development tools like python	Fast standardization, DSL	Interactive testing, OpenCL support, sustainable support of standards
Energy and Resiliency	Distributed computing techniques to handle resiliency/fault tolerance	Reduced cost, improved fault tolerance	Energy optimized workflows: HPC systems including energy monitoring and profiling.	Fault handling, less precision	Energy aware algorithms

CoE and SRA-3

Domain CoE: What are the key scientific challenges and what does it take to achieve them?

	POP	CoEGSS	NOMAD	EoCoE
HPC System Architecture and Components	Throughput oriented devices (vectors), memory architectures and how to use them, architectural support for runtimes, mechanisms to monitor progress and notify runtimes in cases of resource preemptions	Convergence between HPC & HPDA, NVRAM, Fast networks	For extreme scale simulations: powerful compute nodes, high bandwidth memory, low-latency networks. For data analytics: convergence between HPC & HPDA	accelerator with shared memory FPGA Processor in memory NVRAM, large memory
System Software and Management	Dynamic, interactive use of available resources, tight and bidirectional communication/cooperation between job schedulers and runtimes	Dynamically scaling jobs, Integration of HPC & HPDA, Visualization (in situ), Data analytics (in situ) Programming Environment	Containers, scalable data bases, interactive access to subsets of resources, remote visualization, synchronization of big data sets between different computer centres in different countries and continents	complex, light and flexible workflow (notebooks based on python may be a solution comparing to other heavy solutions) Allowing an easy access to large HPC ressources Runtimes Heterogenous job handling for coupled codes/libraries
Programming Environment	Programming model and runtime support for malleability, asynchrony/out of order task execution, hide heterogeneity and tolerate latency and variability, powerful performance analytics in tools, tools for task dependencies and memory access patterns, programmers mindset from bottom-up latency dominated to throughput oriented mentality	Well-defined standards /and tools that implement the standards) for agent-based modeling, Compilers, Debuggers	MPI, OpenMP, GPU programming, Python, tools for profiling and optimization of scalable data bases	Task programming automatic optimization (as BOAST) domain specific language to test new algorithms more easily Performances analysis tools Automated, continual performance monitoring of production
Energy and Resiliency	Better integration between algorithmic based fault detection techniques and mechanism in the infrastructure from detected errors	Not important for CoeGSS	Energy aware system operation, algorithms and workflows	Fault tolerance Analysis in situ

CoE and SRA-3 (cont'd)

Domain CoE: What are the key scientific challenges and what does it take to achieve them?

	BioExcel	CompBioMed	MaX	ESiWACE	E-CAM
Balance Compute, I/O and Storage Performance	Post-processing on the fly, Data-focused workflows, handling lots of small files in bioinformatics	Post-processing on the fly, easy transfer between storage tiers	HT material science workload becomes quickly memory and I/O bound: systems with high IOPS and post posix data objects are required.	Post-processing on the fly, multi-tier software	Fast access of archive data, multi-platform workflows, multi-threaded applications for hybrid production/analysis applications
Big Data and HPC usage Models	Proximity of data generation and analysis/visualization resources, workflows, machine learning for analyzing simulation data, high-throughput sampling	Analytics of simulation outputs, visualisation	Workflows, intelligent data analytics	Recomputing, data analytics	Fast access to data bases, data mining
Mathematics and algorithms for extreme scale HPC systems	Multi-scale algorithms, task-parallel algorithms, Electrostatics solvers, ensemble sampling & clustering theory, ensemble simulations	Novel time stepping algorithms, automated implementation of multiscale computing patterns	New algorithms avoiding synchronous (unnecessary) data dependency and exploiting unreducible data dependency tree (nesting), to improve concurrency and locality	Disruptive numerical methods (discretization), data placement	Memory/cache aware algorithms, asynchronous algorithms, efficient handling of long range/collective correlations

CoE and SRA-3 (cont'd)

Domain CoE: What are the key scientific challenges and what does it take to achieve them?

	POP	CoEGSS	Nomad	EoCoE
Balance Compute, I/O and Storage Performance	Integration of asynchronous I/O interface in programming model, better integration of programming models/languages and persistent storage	Converged systems, Live data analytics, Strong data movement capabilities	Data-centric workflows, high I/O bandwidth, handling huge number of files	In situ data analysis coupled to dedicated I/O system High-band large capacity storage
Big Data and HPC usage Models	Better integration between programming model and storage interface, more dynamic, interactive supercomputing practices, make users/programmers aware of cost/benefit of each individual data and computation for better resource/storage scheduling	HPDA Platform support, Algorithms / Models for efficient HPDA	High performance data analytics of simulation outputs and derived data, faster analysis of complex and Big Data structures (map reduce, faceted search), fast and efficient data mining. Significant advancements in hard and software for handling Big Data more efficiently	Techniques of data analysis coming from Big Data (robust solution with fault tolerance) Reproducibility and ease of calculations based on notebooks High-throughput computing
Mathematics and algorithms for extreme scale HPC systems	Algorithm complexity (computation/communication). asynchrony and variability tolerance, algorithm based fault detection	Algorithms for efficient data analytics	Novel scalable algorithms for extreme scale simulations and for extreme scale data analytics	parallel dynamics (molecular or for climate) multiscale modelling so better coupling between codes as QM/MM linear algebra Parallel in time algorithms