New systems old tools – How to cope with future HPC computing systems?

Stefan Wesner[1]

[1]High Performance Computing Centre Stuttgart (HLRS)
Universität Stuttgart
Nobelstrasse 19, 70569 Stuttgart, GERMANY
wesner@hlrs.de

The talk reflects on two major developments imposing challenges on the available tools for HPC computing systems. The first angle is given on system management tools and the challenges that need to be addressed based on future developments expected from the hardware side, but similarly based on potentially new operational models supporting different type of users. As an example of our related research activities the German national research project TIMaCS (Tools for Intelligent System Management of Very Large Computing Systems) is shortly presented and how it approaches the challenges outlined before.
The second part of the talk discusses the need for new development environments in particular supporting new and inexperienced user communities entering the HPC domain driven by the global trend for parallel computing based on multi- and manycore chips. As an example the Eclipse Parallel Tools Platform is shortly presented and identified topics that need further investigation from an HLRS viewpoint are provided.

New systems old tools – How to cope with future HPC computing systems?

Stefan Wesner
Deputy Director, High Performance Computing Centre Stuttgart

Outline

- System management viewpoint
  - Relevant trends for future computing systems
  - Context of potential future compute service provision
  - Identified Challenges and potential approach
- How to support application developers in the future?
  - Different type of users
  - The Eclipse PTP project
  - Identified research topics for a Parallel Development Environment based on Eclipse PTP

SYSTEM MANAGEMENT VIEWPOINT

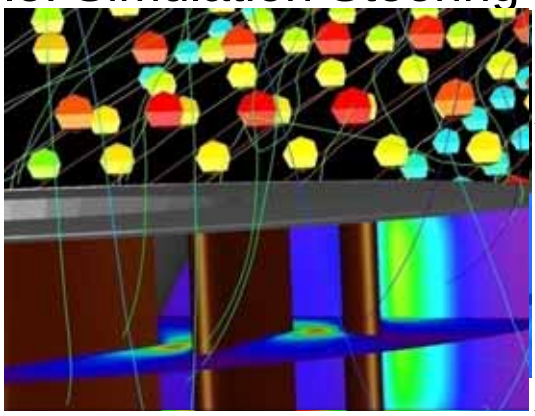
Relevant trends for future computing systems

- Soon expected multi petaflops systems will have a very large number of cores
- More complex and sophisticated interconnects
- Hybrid computing systems
  - Combining different architectures in a single system
  - Conceptually support an evolutionary upgrade policy
    - E.g. extend the hybrid system with a new architecture
    - Replace only a part of the system
  - "Team Model" each part does what it can do best (e.g. computing or IO or …)
- Time to market for new products becomes shorter and shorter

Relevant trends for future computing systems (continued)

- Different operational models in parallel
  - Classical „ssh“ based access (interactive/batch)
  - Grid based access using different types of middleware (batch)
    - UNICORE, GT4, etc. including portal based access
  - Supporting „urgent computing“ e.g. using SPRUCE and HARC from TeraGrid
    - Privileged access e.g. driven by external SLAs
    - Useful for mission critical simulations e.g. from industrial customers or for medical simulation supporting the preparation of a treatment
  - Interactive Simulation Steering and Virtual Prototyping
  - Different operating systems for different jobs on the compute nodes
  - Virtualisation technologies (e.g. XEN)

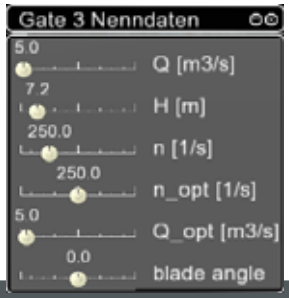
Example: Simulation Steering

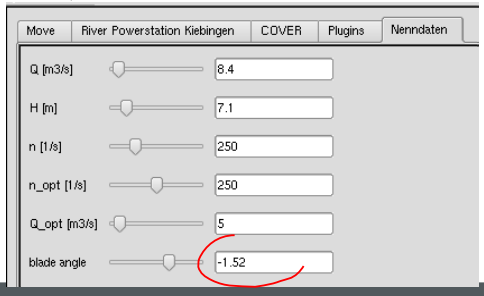


Simulation

Visualisation

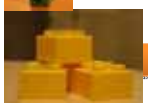
Interaction





Potential future compute service provision model




Provider A (e.g. Computing)

Provider B (e.g. Computing)

Provider C (e.g. Visualization)

Provider D (e.g. Data Storage)

Provider E (e.g. Accounting)

Provider F (e.g. Identity)

Provider G (e.g. Knowledge)

Provider H (e.g. Simulation Services)

Customer

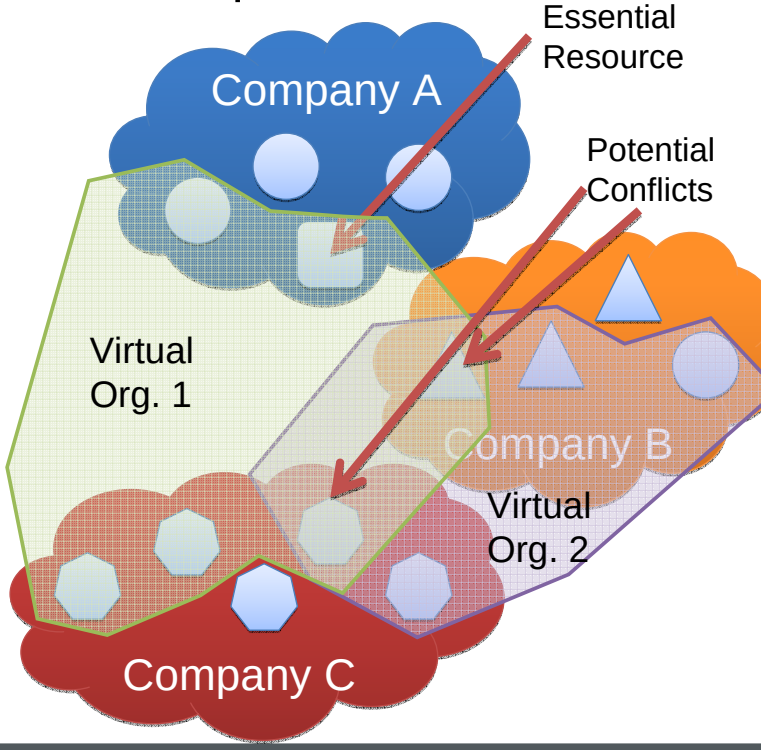
Relationship is described using Service Level Agreements

Same symbol represents rich service description allowing integration

Information provided about the "bricks" may vary from customer to customer

Potential future compute service provision model

- Aggregation of resources from different providers require
  - Define fulfillment criteria using Service Level Agreements
  - Need adaptive management of resources
  - Demand for automated and customer driven management



Identified Challenges

- Existing monitoring solutions do not scale to the size of future computing systems
- More holistic approach to get a system status is needed covering hardware, network, middleware, applications, licenses, ...
- No common information model available
- Management of systems must be automated but system administrators must retain full control
- Different level of abstractions are needed to provide a picture on system health, operational quality and alignment with business/operational objectives
- Variety of operational models and SLA driven service provision needs to be reflected in modifications of rules and behaviour of the system

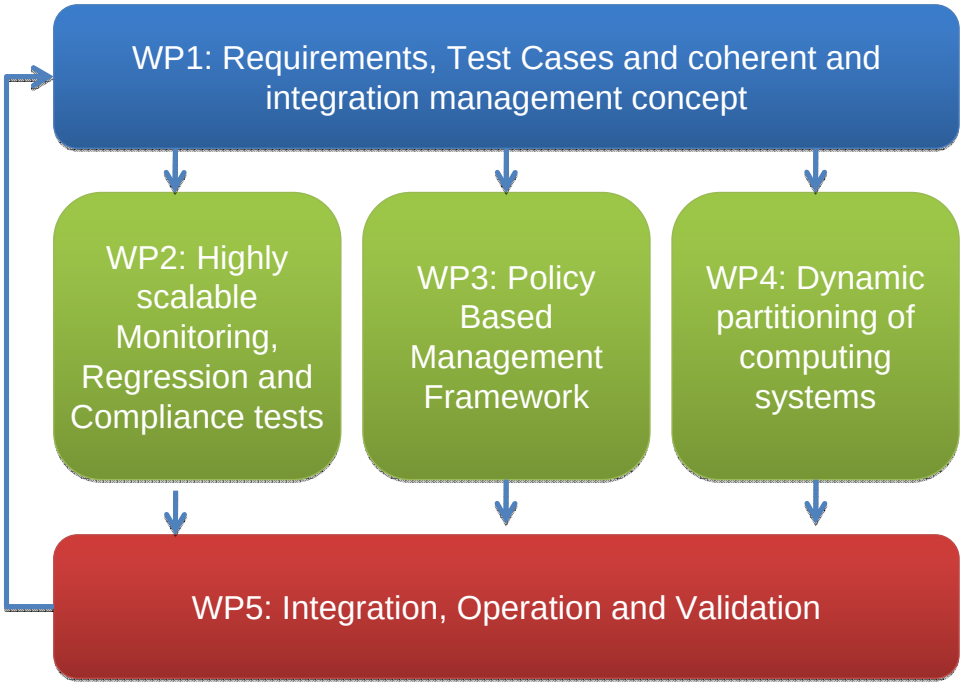
TOWARDS A SOLUTION – THE TIMACS PROJECT

TIMaCS – Key Facts

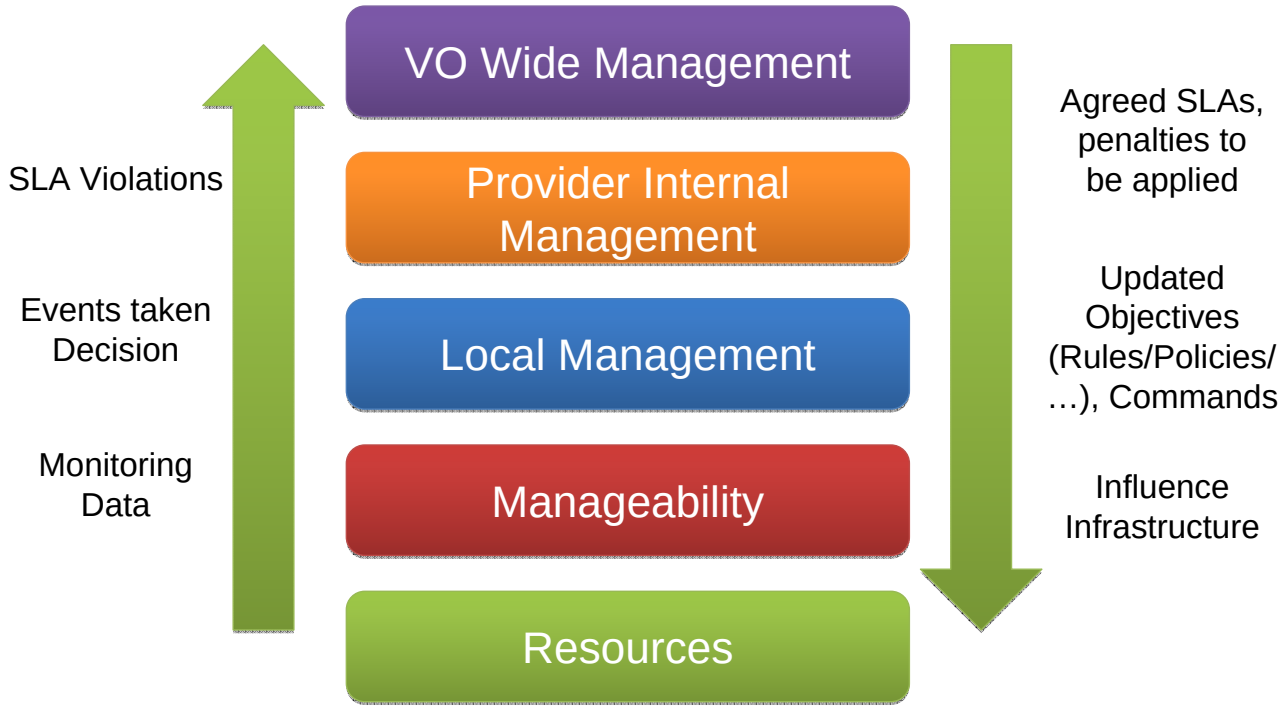
- TIMACS - Tools for Intelligent System Management of Very Large Computing Systems
- Funded by the german ministry for research (BmBF)
- Partners
  - NEC Germany
  - Science & Computing AG
  - Universität Marburg
  - Universität Dresden
  - HLRS (coordinator)
- Overall Funding 1,5 M€
- Duration 3 years.

- Key Objectives in a nutshell
  - Concept and realisation of a highly scalable monitoring solution for very large computing systems
  - A dynamic cluster partitioning system based on virtualisation
  - A knowledge based Management Framework allowing automated application of human defined policies
  - Pre-emptive and proactive system management.

TIMaCS – Project Organisation



TIMaCS – Layered Management Model



TIMaCS – Achievements of the first 3 months

- Survey of and experiments with existing monitoring tools as potential basis
  - Nagios, Quattor, Self-Developed software
- Analysis of existing information models for describing systems status or alerts
  - Common Information Model (CIM) from DMTF
  - GLUE schema defined at the Open Grid Forum
  - Information model of proprietary tools
- Initial experiments with XEN for the dynamic partitioning of the compute system

HOW TO APPROPRIATELY SUPPORT APPLICATION DEVELOPERS?

Experienced HPC users

- Less than 1% if the software development community?
- Used to rely on low level access and “command line tools”
- Large amount of existing codes written in Fortran, C, …
- Acceptance of new programming models such as UPC, CAF?
- Acceptance of new languages only if interoperable with existing codes written in Fortran/C
- Vi, emacs and makefiles are their major tools

New HPC users

- Parallel computing will be a general skill for software developers in the near future
- Large amounts of code written in Java, C#, C++, C, …
- Expect availability of rich development environments such as Visual Studio or Eclipse
- Significantly reduced hardware awareness
- Likely more open for new programming languages such as X10, Chapel, …

H L R S

An example community for new HPC users: LarKC

- The aim of the EU FP 7 Large-Scale Integrating Project LarKC is to develop the Large Knowledge Collider, a **platform for massive distributed incomplete reasoning that will remove the scalability barriers of currently existing reasoning systems for the Semantic Web.**
- Several hundreds of thousands lines of Java Code
- Zero experience in (massively) parallel computing
- Used to rely on rich set of development tools such as
  - Integrated development environment
  - Syntax and Semantic validation of code while typing
  - Rich support for version control systems, profiling, debugging, etc. right from the editor environment
- Modelling approaches e.g. based on the Unified Modelling Language (UML) generating programme code is common practic

H L R S

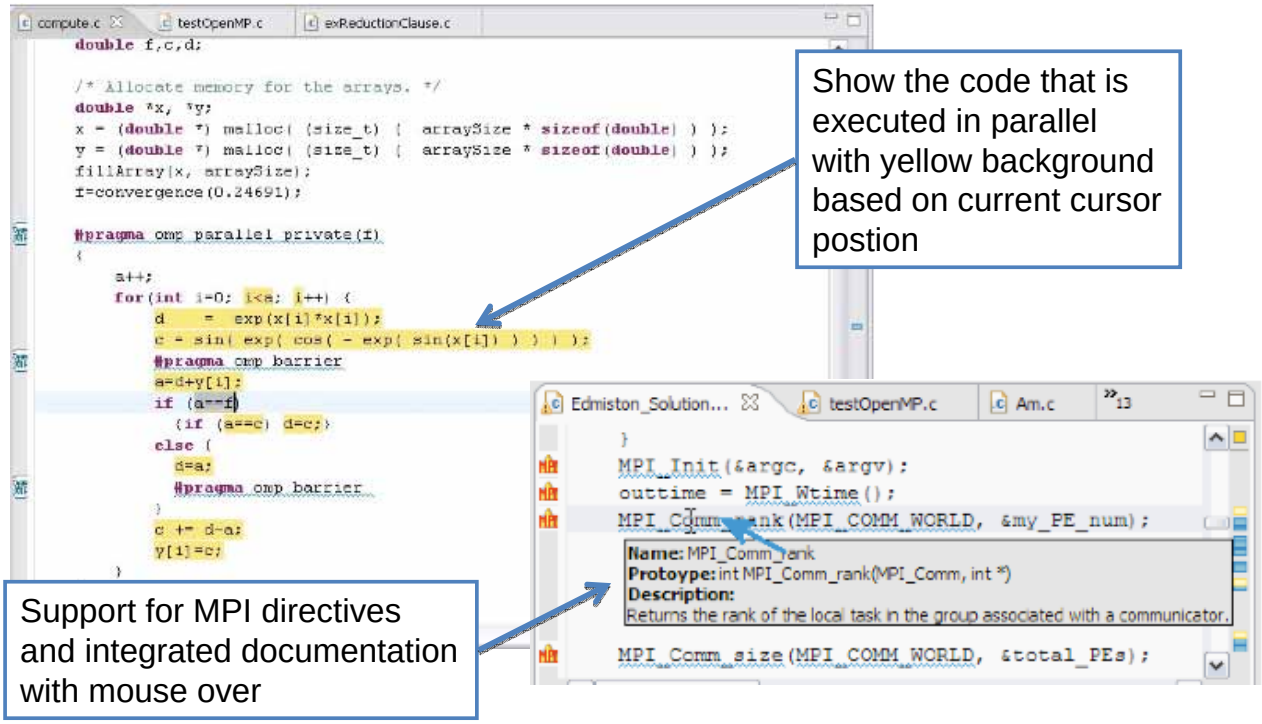
How to support such different user communities?

- “Parallel Computing newbies” need something close to their familiar environment to minimise the gap
  - Integrated Development environment specifically supporting Parallel Computing (SMP, MPP)
  - Support for new programming languages e.g. X10, Chapel, UPC,
  - Extension of modelling approaches e.g. UML with specific elements to model data distribution and performance related aspects
- “Heroic” application developers used to an HPC environment
  - continued support for command line tools and detailed profiling information
  - Potentially slow adoption of the more convenient environment?

An example the Eclipse Parallel Tools Platform

- Eclipse PTP is an open source effort http://www.eclipse.org/ptp
- Designed to support a variety of different programming models
  - Sequential, threaded, Message Passing (MPI), OpenMP, global address space
- Aims to hide the complexity of parallel programming environments by realising integrated support for
  - Different Runtime systems, Job schedulers, etc.
- Target group for the platform are
  - Experienced users seeking for a more convenient development environment
  - Eclipse developers starting parallel programming

Example Eclipse PTP: Specific editor extensions



Status of Eclipse PTP@HLRS

- Initial experiments using C and MPI are promising
- Goal is to investigate further into the tools platform as part of research projects
- Identified tasks
  - Assess applicability of the platform for different programming languages including (A)PGAS languages
  - Investigate of applicability of modelling approaches for Parallel Computing
    - Develop a UML profile for parallel computing based on existing profiles for Realtime UML
    - Integration with the Eclipse Modeling Framework Project (EMF)
  - Collect experiences if the tool is able to reduce the gap for non parallel programming experts
  - Validate applicability of provided tools such as the parallel debugger and the runtime support

Conclusions

- System Management needs to evolve to be more automated to cope with
  - Increased complexity of the hardware
  - New operation models
  - Enable faster responses
  - Ultimately realise a Business goals and IT alignment
- Support for Application Developers
  - Need to cover different types of users
  - Support experienced users with existing and emerging programming models and corresponding tools
  - Additionally exploit potential new user communities with
    - An environment familiar for them e.g. Eclipse
    - Abstraction of the environment minimising the entry gap
    - New programming languages

THANK YOU

Dr. Stefan Wesner, wesner@hlrs.de