Virtual Infrastructures for eScience

2012-09-16

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

eScience

- Definitions
  - "computationally intensive science carried out in highly distributed network environments" (Wikipedia)
  - "any type of science that requires and / or benefits from use of distributed computational resources" (colloquial)
- Distributed computational resource environments
  - High-Performance Computing (HPC)
  - High-Throughput Computing (HTC)
  - Grid Computing
  - Cloud Computing
  - Hybrid environments

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

Virtual Infrastructures

- Distributed software-based systems that
  - virtualize, (inter)connect, and abstract resource systems
  - focus on computational capabilities as utilities
  - provide non-trivial qualities of service
- Characteristics
  - distributed systems
  - decentralized algorithms
  - autonomous mechanisms
  - heterogeneous resources
  - complex security models
  - multiple administrative domains
- Aim
  - generic, reusable infrastructure capabilities
  - domain-specific toolkits for advanced science

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

Sample Research Areas (Group)

- Programming models for distributed computing
- Distributed and federated grid and cloud architecture
- Proactive elasticity control for cloud computing resources
- Placement (scheduling) of virtual machines and data
- Live migration of large-scale virtual machines
- Distributed and hierarchical fairshare scheduling
- Distributed resource brokering and storage systems
- Theory and algorithms for distributed resource allocation
- Decentralization models for distributed systems
- Energy efficiency in distributed computing

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Sample Project: Stochastic Simulation

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

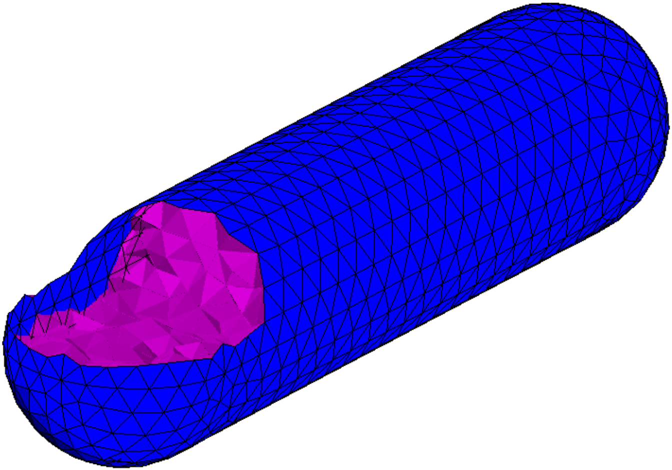
- Stochastic simulation of reaction-diffusion kinetics
- Applications in molecular systems biology, e.g., modeling protein interactions on the hull of bacteria
- Collaboration between UmU, UU, and UCSB
- Project focus
  - scale an existing computational model to distributed computing environments
  - develop a generic tool that allows domain experts to use distributed resources transparently

URDME-StratUm

- Hybrid system developed for the project
- COMSOL for geometric modeling
- Matlab-based software package for computational modeling
- Native (C++) solvers generated for experiments
- Java-based system for distribution and management of computations

Geometrical Model

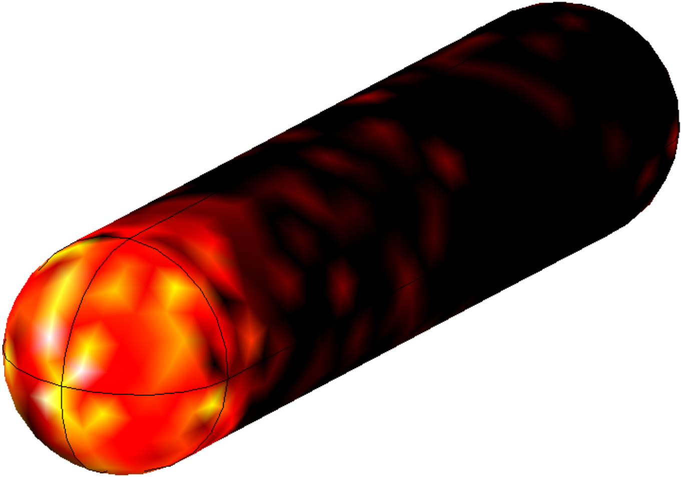
Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation



P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Stochastic Modeling

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation



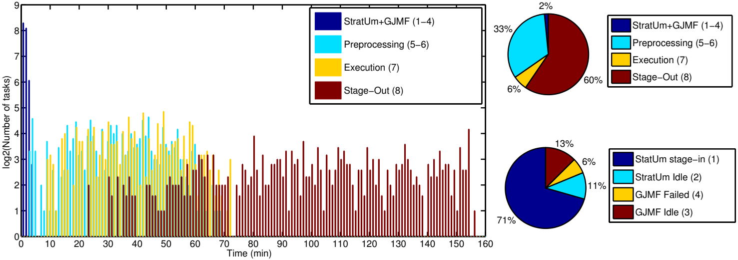
P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Parallelization

- Modeling is (coarsely) embarrassingly parallel
  - Many simulations can be run in parallel
  - Simulations operate on (mainly) shared data
- Simulation computations relatively small
  - Execution time depends on location in parameter space
  - Typical execution time on the order of 5-10 minutes
- Simulation data relatively large
  - Parameter data on the order of 10 MB
  - Resulting data on the order of 10-100 MB
- Large amounts of simulations required
  - Thousands of simulations required to reach statistical significance
  - Hundreds of thousands required to do fine-grained biological analysis

Performance

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation



StratUm

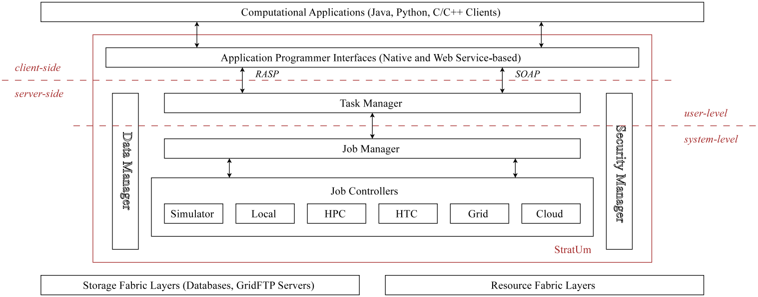
Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

- StratUm - a toolkit for distributed computing
  - security (credentials) management
  - data management
  - computation (job) management
  - client APIs (for ease of use and integration)
- Focus on distributed job and resource management
- Aims to facilitate the first steps into distributed computing
- Main approach: abstraction and automation

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

SAMPLE PROJECTS – Stochastic Simulation

StratUm Architecture



P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

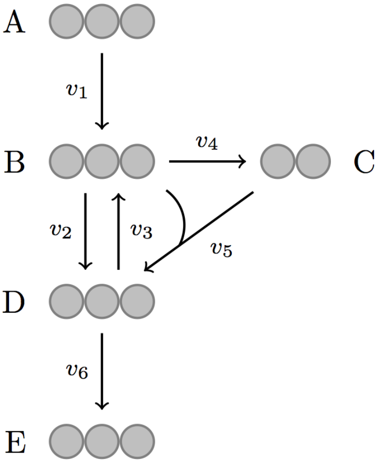
Sample Project: Flux Estimation

- Metabolic flux estimation
- Simulation of human cellular metabolism mechanisms combined with biological experiments
- Collaboration between UmU and KI
- Project focus
  - develop a proposed computational model and scale it to distributed environments
  - develop computational tools that can collaborate with biological experiments

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Metabolic Networks

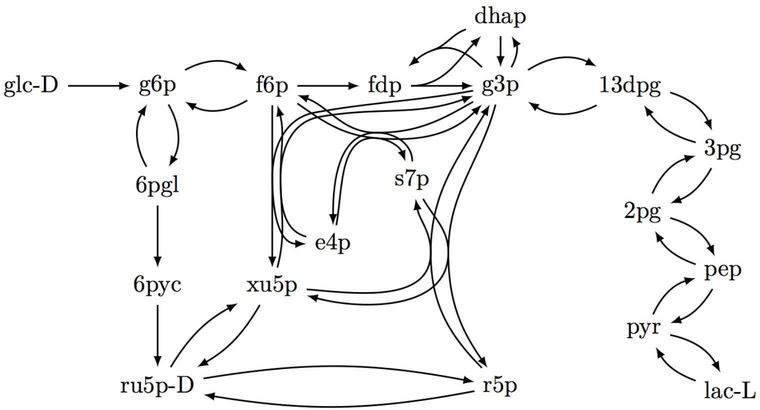
Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation



P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Metabolic Network

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation



Project Contributions

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

- Work in progress
- Linear algebra improvements to the computations
- New optimization model for the modeling
- Problem-based computation distribution algorithm
- Computation management interfaces and tools

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Fair Resource Allocation

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

- Automated distributed fair resource allocation
- Decentralized system that allows stakeholders to recursively allocate resource capacity in projects
- Fair in the sense that all users / projects / VOs get their allocated share of available capacity
- Approach taken: fairshare scheduling
- Project focus
  - develop a recursively delegatable resource allocation model
  - develop a scalable decentralized distributed system for fairshare prioritization
- Prototype system (Aequus) scheduled for testing in SweGrid (fall of 2012)

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

Aequus

- A decentralized, distributed system for distributed computing policy allocation enactment
- Builds on injection of a fairshare job prioritization mechanism in local scheduling decisions
- Enforces a global (infrastructure-wide) fairshare allocation scheme through local computations on distributed data
- Builds on three contributions
  - policy model
  - algorithm
  - architecture
- Main approach: decentralization and precomputation
  - usage policies defined by stakeholders
  - usage information propagated in a distributed system
  - decentralized local mechanisms operate on global data

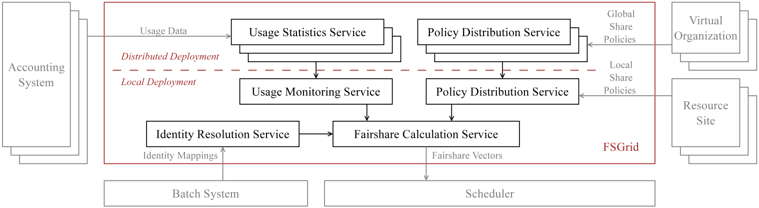
P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

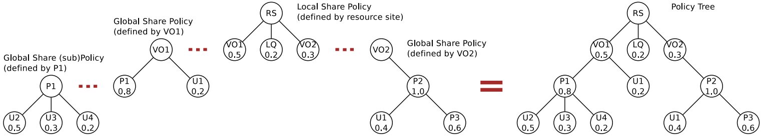
Contributors (not ordered)

- Erik Elmroth, Umeå University
- P-O Östberg, Umeå University
- Daniel Espling, Umeå University
- Sverker Holmgren, Uppsala University
- Andreas Hellander, UC Santa Barbara
- Linda Petzold, UC Santa Barbara
- Brian Drawert, UC Santa Barbara
- Gonzalo Rodrigo, Umeå University
- Luis Tomas, Universidad de Castilla-La Mancha
- Roland Nilsson, Karolinska Institutet
- Jesper Tegner, Karolinska Institutet
- Lars Karlsson, Umeå University
- Eddie Wadbro, Umeå University
- Blanca Caminero, Universidad de Castilla-La Mancha
- Carmen Carrion, Universidad de Castilla-La Mancha
- Mikael Öhman, Umeå University
- Niclas Lockner, Umeå University
- Sebastian Gröhn, Umeå University
- Anders Häggström, Umeå University

Aequus Architecture



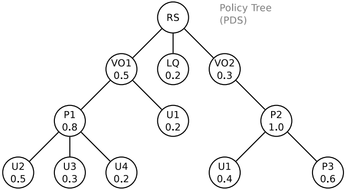
Aequus


Global Share (sub)Policy
(defined by P1)
P1
U2 0.5
U3 0.3
U4 0.2
Global Share Policy
(defined by VO1)
VO1
P1 0.8
U1 0.2
RS
Local Share Policy
(defined by resource site)
VO1 0.5
LQ 0.2
VO2 0.3
VO2
Global Share Policy
(defined by VO2)
P2 1.0
U1 0.4
P3 0.6
Policy Tree
RS
VO1 0.5
LQ 0.2
VO2 0.3
P1 0.8
U1 0.2
P2 1.0
U2 0.5
U3 0.3
U4 0.2
U1 0.4
P3 0.6

Aequus Fairshare Calculation

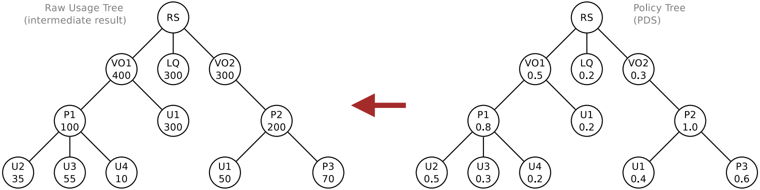
Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

- Usage policies compiled from multiple sources



Aequus Fairshare Calculation

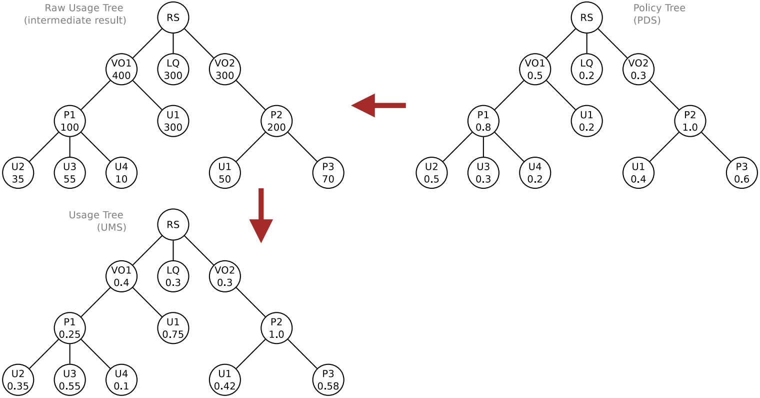
- Usage trees constructed from usage summaries



P-O Östberg
Dept. Computing Science, Umeå University, Sweden

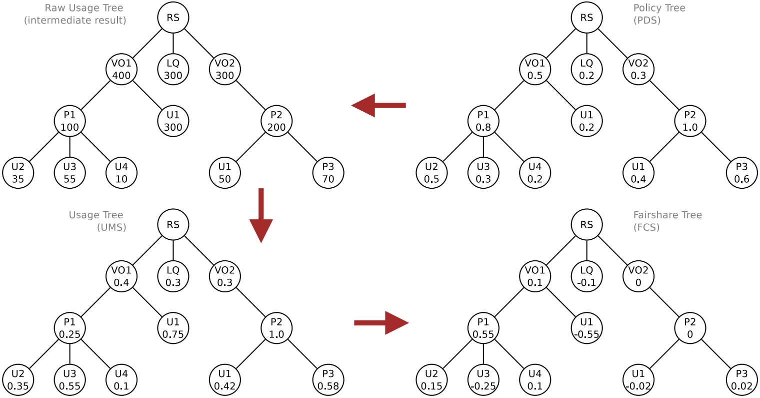
Aequus Fairshare Calculation

- Usage trees normalized to enable policy comparison



Aequus Fairshare Calculation

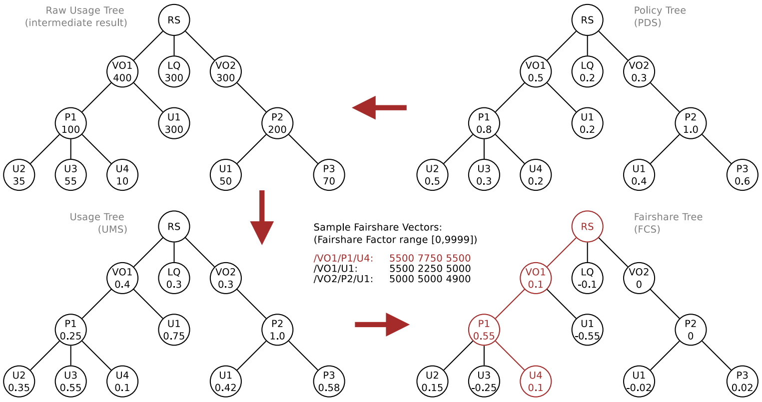
- Fairshare trees constructed from policy and usage trees



P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Aequus Fairshare Calculation

- Fairshare vectors defined by paths in fairshare trees



Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation

Aequus Fairshare Operator

Comparison of policy and usage elements is performed by (configurable) fairshare operators defined as

$$d = kd_a + (1-k)d_r$$

where

$$d_a = p - u$$

$$d_r = \begin{cases} \left(\frac{p-u}{p}\right)^2 & \text{for} \quad u < p \\ 0 & \text{for} \quad u = p \\ -\left(\frac{p-u}{u}\right)^2 & \text{for} \quad u > p \end{cases}$$

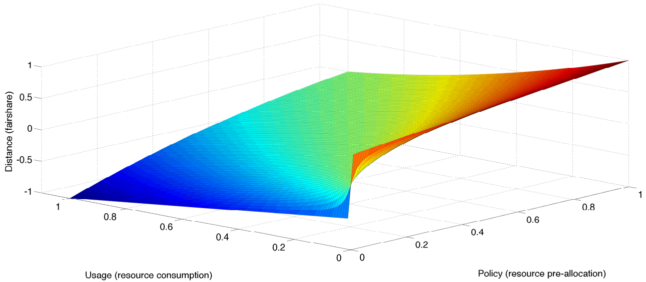
$$k, p, u \in [0, 1]$$

$$d, d_a, d_r \in [-1, 1]$$

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Fairshare Value Space

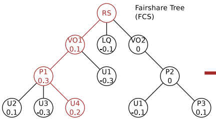
- The fairshare value space is spanned by basis vectors formed by unit policy and usage vectors
- Absolute and relative fairshare operators provides different fairshare balance measurements



P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Fairshare Vector Extraction

- Fairshare vectors are extracted from fairshare tree paths
- Vector elements are translated to integer format
- Vectors are ordered lexicographically or arithmetically
- Users and jobs are prioritized by vector ordering
- Entire process is precomputable



Node Value Fairshare Vectors
(Padded with 0.0 if applicable)

/VO1/P1/U2	0.1	0.3	0.1
/VO1/P1/U3	0.1	0.3	-0.3
/VO1/P1/U4	0.1	0.3	0.2
/VO1/U1	0.1	-0.3	(0.0)
/LQ	-0.1	(0.0)	(0.0)
/VO2/P2/U1	0.0	0.0	-0.1
/VO2/P2/P3	0.0	0.0	0.1

Integer Value Fairshare Vectors
(Value range [0,9999])

/VO1/P1/U2	5499 6499 5499
/VO1/P1/U3	5499 6499 3499
/VO1/P1/U4	5499 6499 5999
/VO1/U1	5499 3499 4999
/LQ	4499 4999 4999
/VO2/P2/U1	4999 4999 4499
/VO2/P2/P3	4999 4999 5499

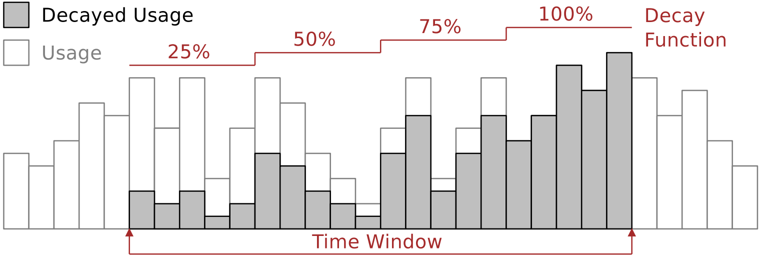
Ordered Fairshare Vectors
(Value range [0,9999])

/VO1/P1/U4	5499 6499 5999
/VO1/P1/U2	5499 6499 5499
/VO1/P1/U3	5499 6499 3499
/VO1/U1	5499 3499 4999
/VO2/P2/P3	4999 4999 5499
/VO2/P2/U1	4999 4999 4499
/LQ	4499 4999 4999

P-O Östberg
Dept. Computing Science, Umeå University, Sweden

Aequus

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation



P-O Östberg
Dept. Computing Science, Umeå University, Sweden

URDME-StratUm Integration

Introduction
Sample Projects
Stochastic Simulation
Flux Estimation
Fair Resource Allocation