FEDERAL UNIVERSITY OF CEARÁ

High Performance Computing in LIA - UFC: Current Status and Future Directions

João Marcelo Uchôa de Alencar

4th Workshop of HOSCAR Project

15-19 September, 2014
Gramado, RS, Brazil

GREat
Grupo de Redes de Computadores
Engenharia de Software
e Sistemas

CENAPADUFC

PARGO

Agenda

- About UFC
- Activities developed by UFC Team
  - GREat
  - ParGO

UFC

- Ceará
  - Brazilian northeast state
  - 9 millions inhabitants
  - GDP: R$ 87,982 billions
- Federal University of Ceará
  - Founded in 1954
  - 42.443 students enrolled
  - 8 campuses across the state (Fortaleza, Quixadá, Sobral, ...)
  - Recently elected the 13th best university in Brazil by *Folha de São Paulo*
  - Patent applications increased 766% from 2008-2009 to 2010-2011

UFC

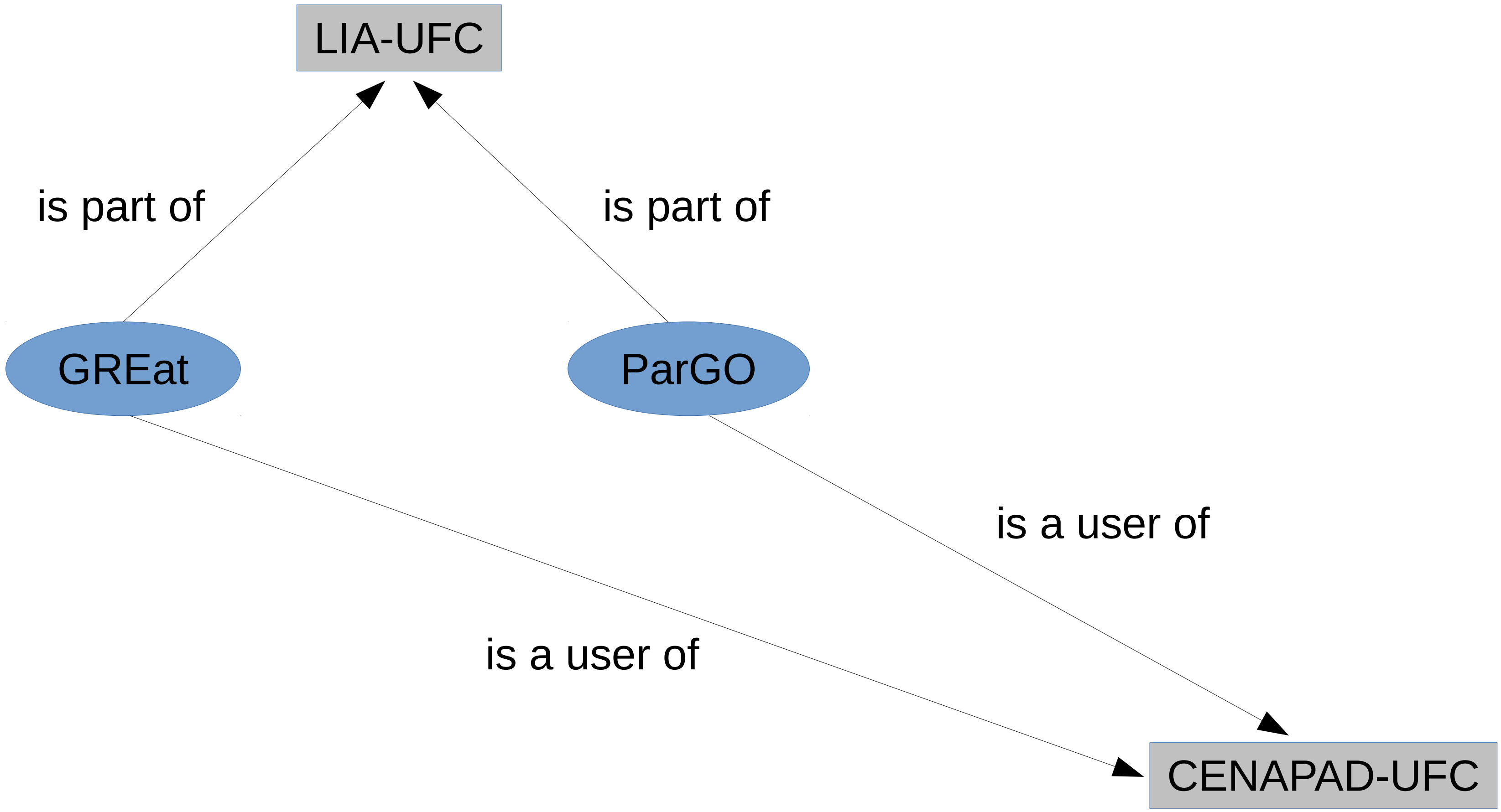
UFC Team involved in HOSCAR

- GREat
  - Prof. Dra. Rossana M. C. Andrade
  - Prof. Dr. José Neuman de Souza
  - Prof. Dr. Fernando A. Mota Trinta
  - Prof. Dr. Danielo Gomes
  - Prof. Dr. Miguel Franklin
  - Prof. Emanuel Ferreira Coutinho
  - Dra. Carina Teixeira de Oliveira
  - Ronaldo Lima
  - Felipe Anderson Maciel
  - Philipp B. Costa
  - Deborah Maria Vieira Magalhães
  - Prof. Paulo A. Leal Rego.
  - Jefferson Ribeiro
  - Renato Neto
  - Igor do Carmo
  - Samuel Soares
- ARIDA
  - José Antônio Fernandes de Macedo
  - Vinicius Pires
- ParGO
  - Prof. Dr. Heron de Carvalho
  - Prof. João Marcelo Uchôa Alencar
  - Prof. Jefferson Silva
  - Cenez Rezende
  - Wagner Al-Alan
  - Anderson Boettge
  - Neemias Gabriel

UFC

- LIA – UFC
  - Is the global lab of the Computer Science Department
  - www.lia.ufc.br
- GREat
  - The Group of Computer Networks, Software Engineering, and Systems
  - www.great.ufc.br
- ParGO
  - Paralelism, Graphs and Otimization
  - www.lia.ufc.br/~pargo
- CENAPAD-UFC
  - National Center for Supercomputing – UFC
  - www.cenapad.ufc.br, @cenapadufc
  -

UFC


LIA-UFC
is part of
is part of
GREat
ParGO
is a user of
is a user of
CENAPAD-UFC

CENAPAD-UFC

- Mission
  - To provide on-demand High Performance Computing (HPC) services to universities, research institutes and other public or private instititutions
  - It is a national center that focuses on meeting the needs of research groups in the north and northeast.
  - SINAPAD

- Computational Cluster
  - Cluster Bull
    - 48 nodes, each with 12 cores e 24 GB RAM
    - Total: **576 cores e 1152 GB RAM**
  - GPUs Nvidia
    - 3 nodes, each also with 16 CPU cores e 96 GB RAM
    - 6 k20 boards
  - Storage
    - 145 TB

- Computational Cloud
  - 5 nodes:
    - Intel Core 2 Duo 2 cores and 2 GB RAM (controller)
    - 2x Intel Core i7 8 cores and 8 GB RAM
    - Intel Xeon com 4 cores and 16 GB RAM
    - Intel Xeon com 12 cores and 32 GB RAM
  - Total: **34 cores and 66 GB RAM**
  - Toolkit: OpenNebula

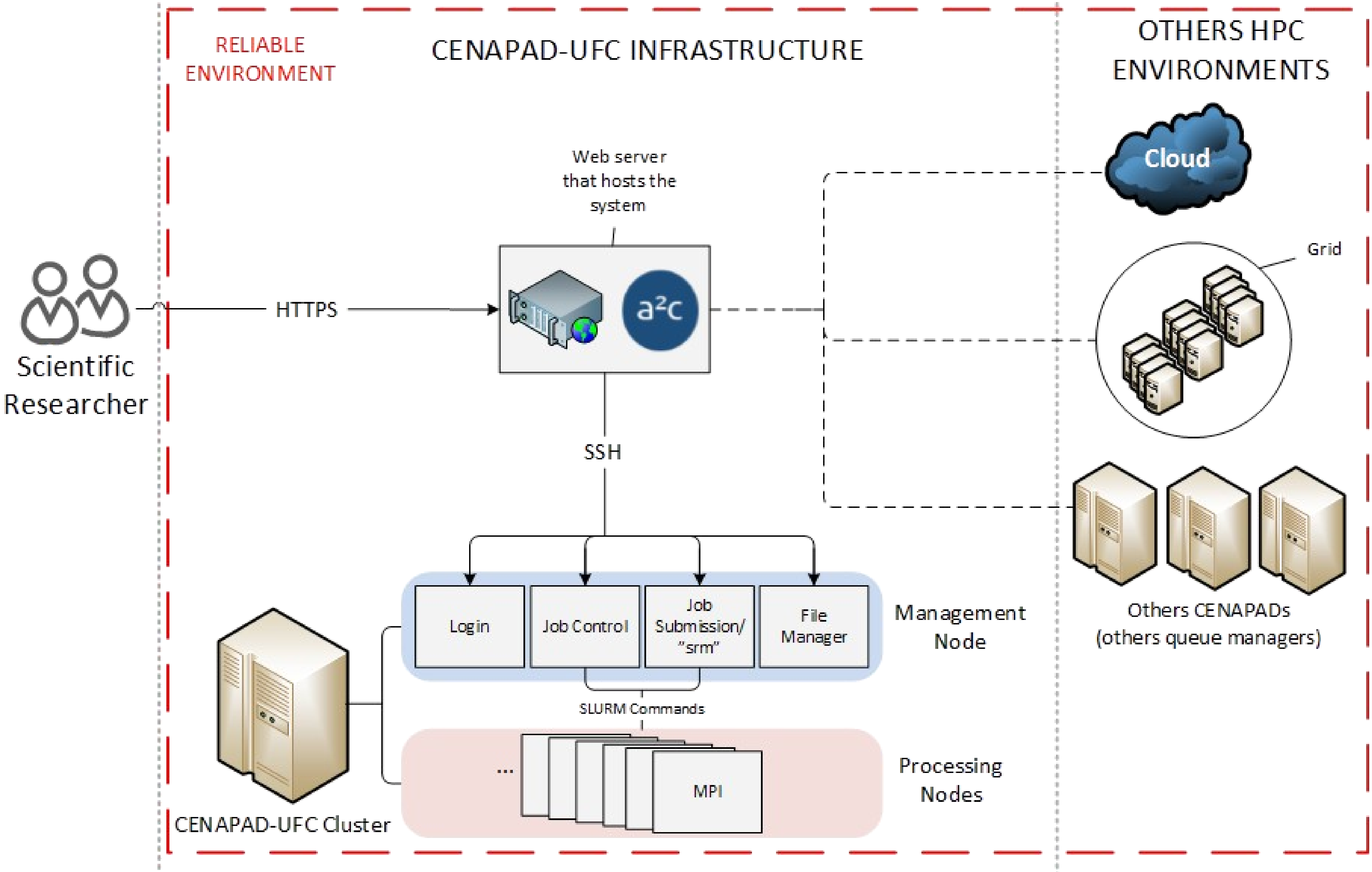
GREat

- Improving end user interaction with HPC resources
- Investigating future scenarios for Cloud Computing
  - HPC
  - Cloud Infrastructure Resource Allocation
  - Future Directions
    - Mobility
    - Security

GREat

a²c

- a²c – a web portal access to cluster
- Motivations
  - Enable easy access to a variety of users with different needs
  - Provide abstractions while retaining flexibility
  - Work as an entry point where scheduling policies may be applied without changing the underlying infrastructure
- INCT-MACC



a²c - Environment Overview

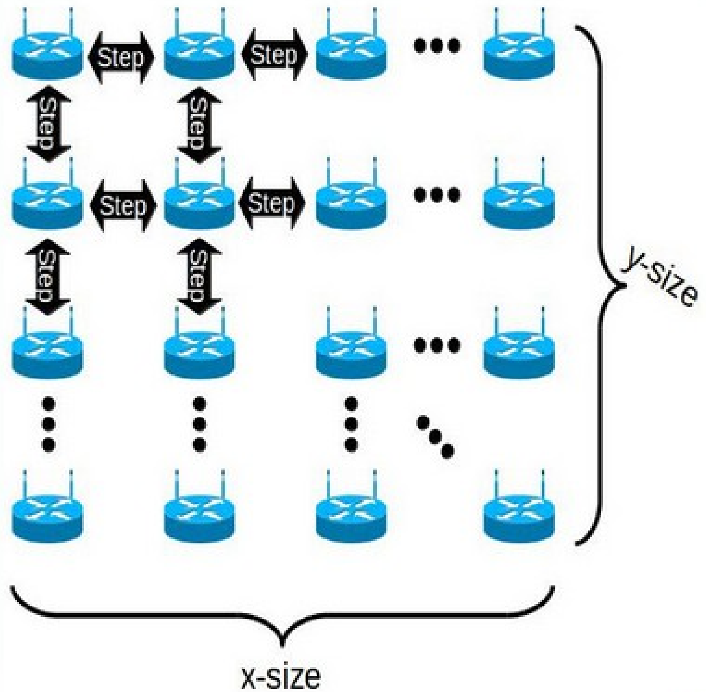
- Portals
  - **Generic:** graphical interface to SLURM (resource manager)
  - **NS3:** network simulation tool
  - **NAMD:** molecular simulation tool

English | Português | Logoff

NS3 | Namd

Redes Mesh | Script Generico

Topologia em grade | Topologia em disco aleatório



Parâmetros da simulação

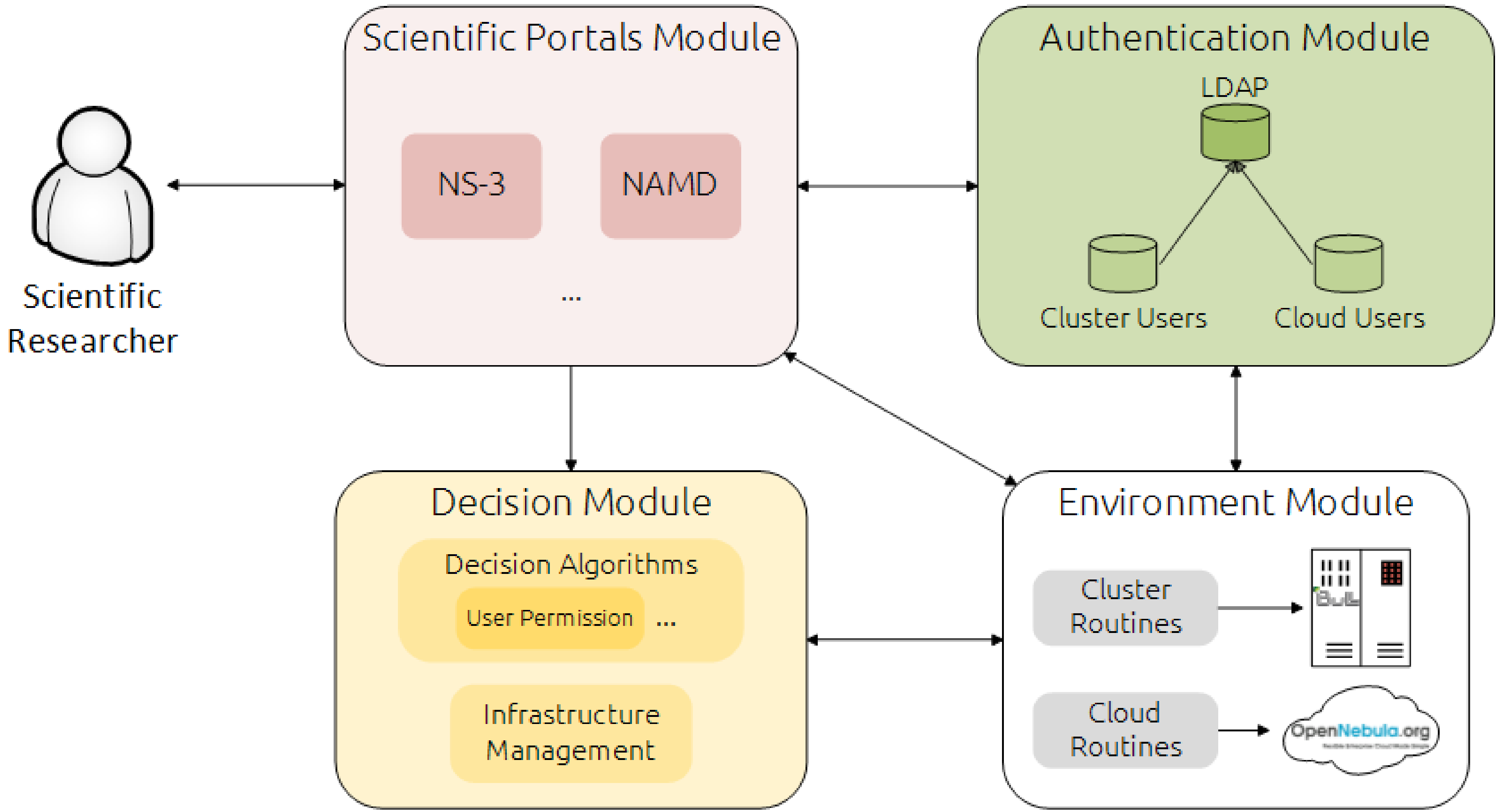
Tamanho Horizontal da grade (x-size)	5
Tamanho vertical da grade (y-size)	5
Espaço entre nós (step)	100
Tempo de simulação (segundos)	100
Número de interfaces de rádio por nó	1
Intervalo de tempo entre transmissão pacotes (segundos)	0,001
Tamanho dos pacotes (Bytes)	1024
Política de escolha de canais (channels)	complete spread
Tipos de trace desejado	XML, PCAP, Graphs

Simular | Cancelar

Simulações

Inicio da Simulação	status	Programa	Parametros da simulação	Baixar	Excluir
Não foi localizado nenhum arquivo					

Centro Nacional de Processamento de Alto Desempenho
Av. Mister Hull, s/n – Campus do Pici – Bloco 901 [Térreo do STI]
CEP: 60.440-554 | Fortaleza – CE – BRASIL | Contato: +55 085 3366.9472 | 3366.9225 | Email – suporte@cenapad.ufc.br



a²c – Architectural Overview

Cloud for HPC

- Cluster and Cloud Integration
  - Deploy cloud infrastructure at CENAPAD-UFC
  - Enhance user experience with cloud resources
  - **If you have a powerful cluster, why do need a private cloud?**

Cloud for HPC

- Cluster
  - Server class processors
  - Great memory per node
  - Fast interconnect
  - Low flexibility
    - Many nodes, changing the configuration is not trivial
    - There are devices with proprietary drivers that may not be updated
  - Expensive to expand
    - Hardware is available, but is expensive
    - Blade architecture may require new chassis
    - Infiniband expansion is not cheap
  - **Perfect tool for scientific computing**

Cloud for HPC

- Private Cloud
  - Server class and desktop class processors
  - Less memory per core
  - Ethernet
  - High flexibility
    - Virtualization allow different OS images to run in the same resource
    - Virtual machines may be updated easily without changing the physical host
  - Cheap expansion
    - Commodity hardware
    - Increasing the number of nodes is straightforward
  - **Not the best performance for scientific computing**

Cloud for HPC

- Perfect scenario for the Cluster
  - A researcher wants to run parallel distributed applications with MPI
    - Low latency demand
    - Using the cloud would offer performance decrease due to ethernet

Cloud for HPC

- Possible scenarios for the Cloud
  - A researcher wants to run legacy applications or code developed by himself/herself with specific requirements for number of processes, operating system, compilers, libraries, etc
    - Changing the cluster setup may not be simple. For example, **incompatible** libraries versions
    - With virtualization, it is possible to create a software environment **identical** to the researcher's setting
  - A researcher wants to execute serial code or multithread only, without MPI
    - He/she is using CENAPAD for performance, but also for **reliability** (no-break, redundant power, etc)
    - Using the cluster may take nodes that would be better used by MPI applications
    - Running serial or multithread only code on the cloud offers **acceptable performance**

Cloud for HPC

- a^2c Decision Module
  - If cluster usage is **high**, but still with available nodes
    - Send all new serial or multithread only jobs to the cloud
  - If cluster queue is **full**
    - Send all jobs (MPI or not) to the cloud. However, if available nodes appear on the cluster, migrate MPI applications from the cloud to the cluster. Migration must be supported by application (for example, GROMACS)
  - If cluster usage is **low**
    - Send all jobs to the cluster

GREat

Cloud for HPC

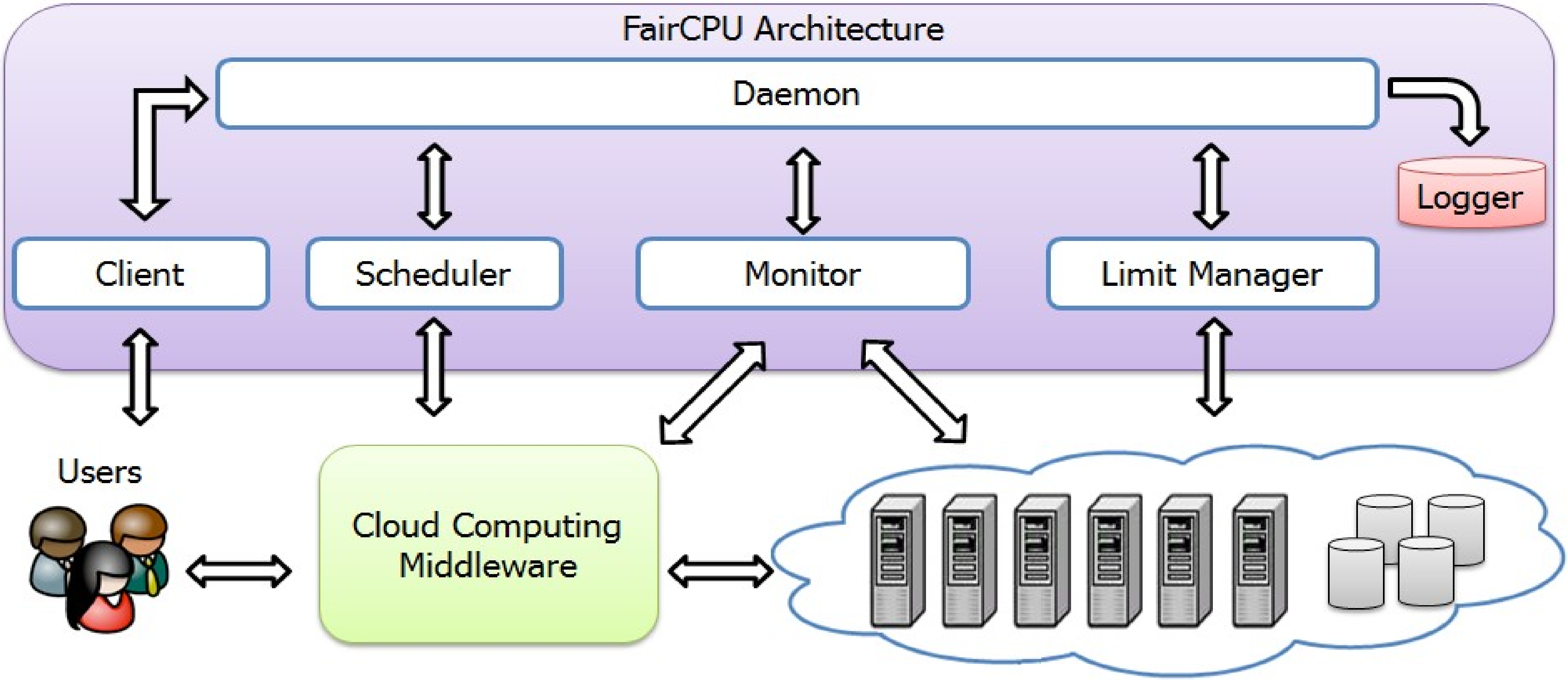
- a²c Cluster and Cloud Integration
  - It is still a work in progress, but from the preliminary data we can see that the queueing time is reduced
  - With more usage data, we expect to show that the overall execution time is lower
  - **Conclusion:** the cluster is faster, but the cloud is easier and cheaper to expand and use, may increase the job throughtput and decrease configuration time before running applications
- Future Work:
    - Futher study the execution of MPI applications on the cloud
    - Migration Strategies
    - Create more portals according CENAPAD-UFC's users needs

Cloud Infrastructure Resource Allocation

- Evaluation of software to set up a private/hybrid cloud
  - OpenNebula, OpenStack, Eucalyptus, CloudStack
- Creation of virtual appliances for easier the deployment of cloud applications
- Development of solutions to handle the heterogeneity of the data center’s physical machines to achieve an homogeneous performance (**FairCPU architecture**)
- Development of techniques to handle elasticity among different cloud solutions and cloud datacenters (hybrid cloud)
- Performance evaluation of parallel applications to Big Data
  - Hadoop, YARN

GREat

Cloud Infrastructure Resource Allocation



Mobile Cloud Computing

- Development of solutions to improve the performance of mobile applications and reduce battery consumption
  - Exploit cloud capabilities (storage and compute) through the use of offloading techniques
  - Orchestration of cloud services in private/local resources (cloudlet concept) and public/remote resources
  - Frameworks for Android and Windows Phone
- Study to improve the performance of private cloud infrastructure for different workload behavior of mobile applications
- Handle mobility issues of such kind of application
  - Handoff, loss of connectivity, mobile applications
- Handle QoS and SLA for mobile applications

Cloud Security

- Data stored on the public cloud should be kept private
- The security requirements might be different
  - The SLA negotiation should regard the customers needs
  - The provider might cash in accordance with the defined security level
- SLA Violation
  - The customers should identify if the SLA was violated
  - The provider should use mechanisms to avoid a violation or to repair after a violation
- How to assure the data privacy when they are stored or processed in the cloud?
  - The metrics related to the parameters should be measurable
  - The negotiation can be automated

GREat

Publications

- VIANA, N. P. ; Trinta, A. M. Fernando ; VIANA, J. R. M. ; ANDRADE, R. M. C. ; GARCIA, V. C. ; ASSAD, R. E . . aCCounts: **Um serviço de Tarifação de Recursos para Computação em Nuvem. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos** (SBRC), 2013, Brasília. XI Workshop de Computação em Clouds e Aplicações (WCGA) - SBRC 2013, 2013. p. 154-155.

- MACIEL, F. A. ; Cavalcante, M Tiago ; QUESADO NETO, J. ; de Alencar, J M. U. ; OLIVEIRA, C. T. ; ANDRADE, R. M. C. . **Uma Arquitetura Flexível para Submissão e Gerenciamento de jobs em Infraestruturas Computacionais de Alto Desempenho**. In: XI Workshop em Clouds e Aplicações (WCGA) - SBRC, 2013, Brasília. 31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2013.

- COUTINHO, E. F ; SOUSA, F. R. C. ; Gomes, Danielo G. ; de Souza, José Neuman . **Elasticidade em computação na nuvem: uma abordagem sistemática**. In: Joni da Silva Fraga; Jacir Luiz Bordim; Rafael Timóteo de Sousa Júnior; William Ferreira Giozza. (Org.). Livro de Minicursos do XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC). 1ed.Porto Alegre: Sociedade Brasileira de Computação (SBC), 2013, v. , p. 215-258.

- COUTINHO, E. F ; REGO, P. A. L. ; GOMES, D.G. ; SOUZA, J. N . **Métricas para Avaliação da Elasticidade em Computação em Nuvem Baseadas em Conceitos da Física**. In: Workshop de Computação em Clouds e Aplicações, 2014, Florianópolis-SC. Anais do XII Workshop de Computação em Clouds e Aplicações - WCGA 2014. Porto Alegre: Sociedade Brasileira de Computação (SBC), 2014. p. 55-66.

ParGO

- Hash Component Model
- Hash Programming Environment (HPE)
- HPC STORM

Hash Component Model

- “Separation of concerns (SoC) is a design principle for separating a computer program into distinct sections, such that each section addresses a separate concern.”
  - Philip Laplante - “What Every Engineer Should Know About Software Engineering”
- A concern is a set of information that affects the code of a computer program.
- A program build upon SoC is said to be modular.

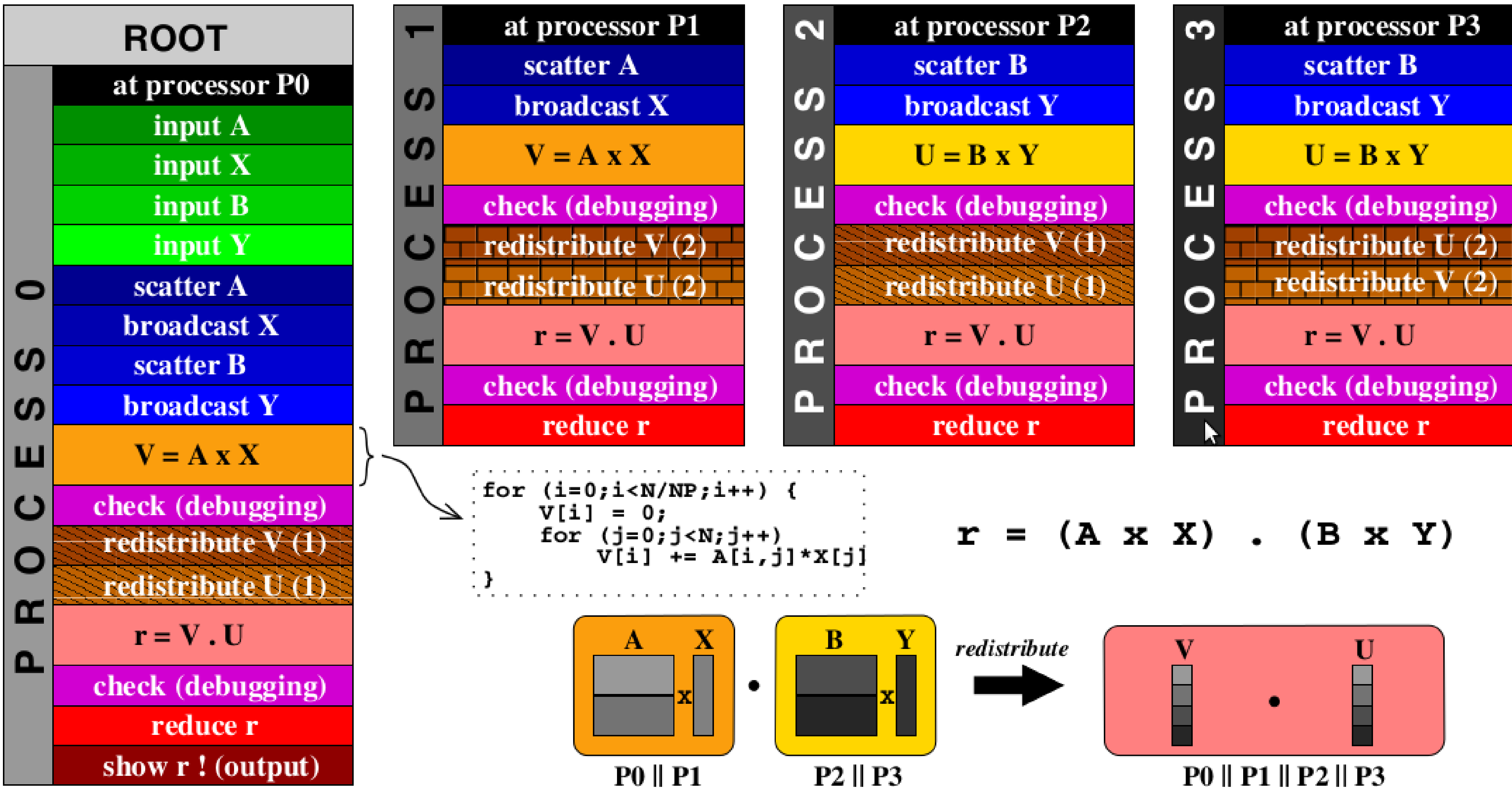
Hash Component Model

- For HPC codes, some examples of concerns
  - A piece of code that represents some meaningful calculation, for example, a local matrix–vector multiplication
  - A collective synchronization operation,which may be represented by a sequence of send/recv operations;
  - A set of non-contiguous pieces of code including debugging code of the process;
  - The identity of the processing unit where the process executes;
  - The location of a process in a given process topology

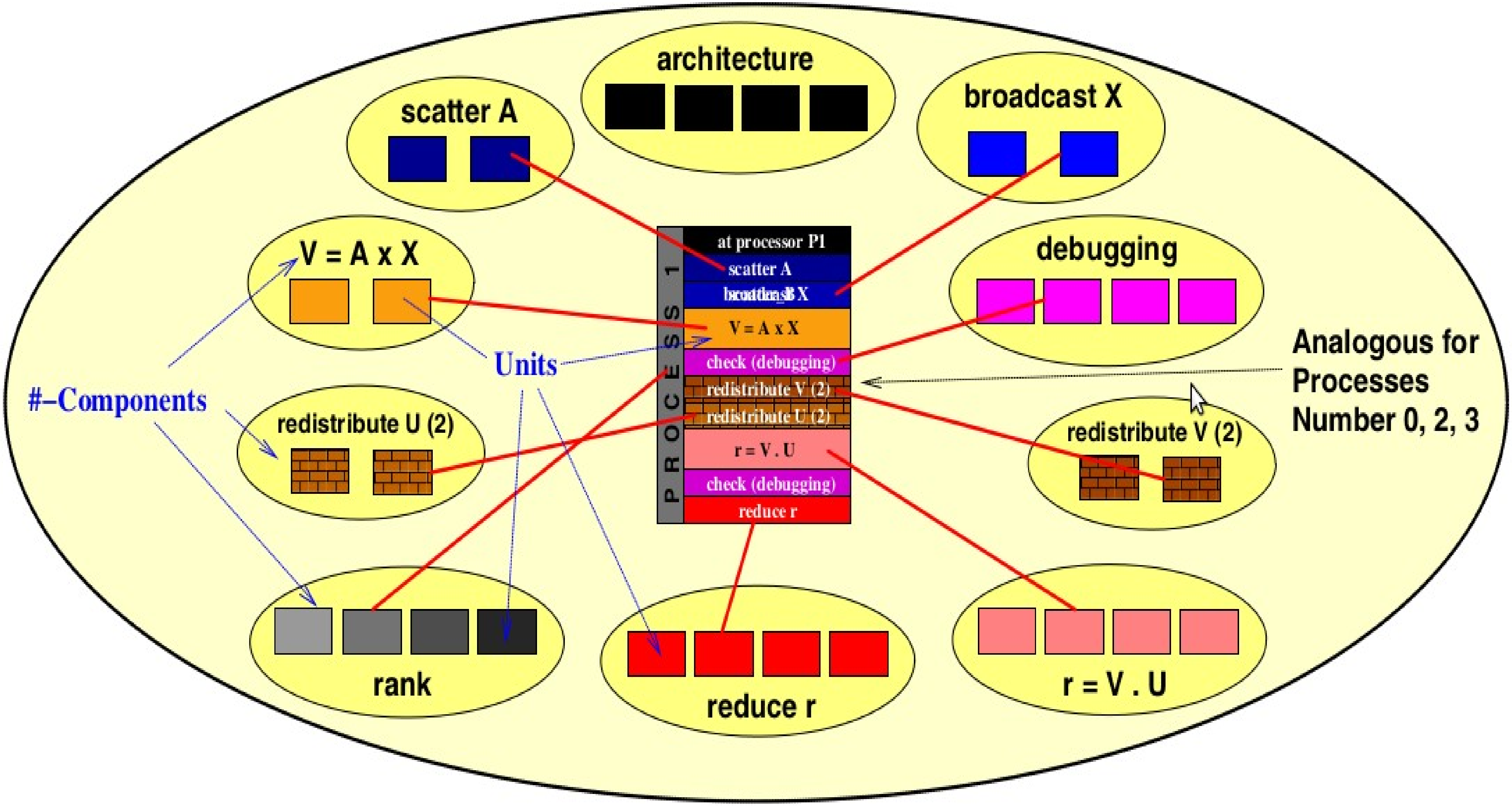
Hash Component Model

- Emerging large scale HPC applications from computational sciences and engineering
  - Software engineering requirements
  - Collaborative environments
  - Capability/capacity computing platforms
  - World-wide scale collaboration and computation
- The **Hash Component Model** enables the development of Component-Based High Performance Computing (CBHPC) applications
  - The separation of **concerns** through process slicing
  - **Orthogonality** between processes and concerns as units of software decomposition

Let A and B be n×n matrices and X and Y be vectors. It computes (A x X) · (B x Y)



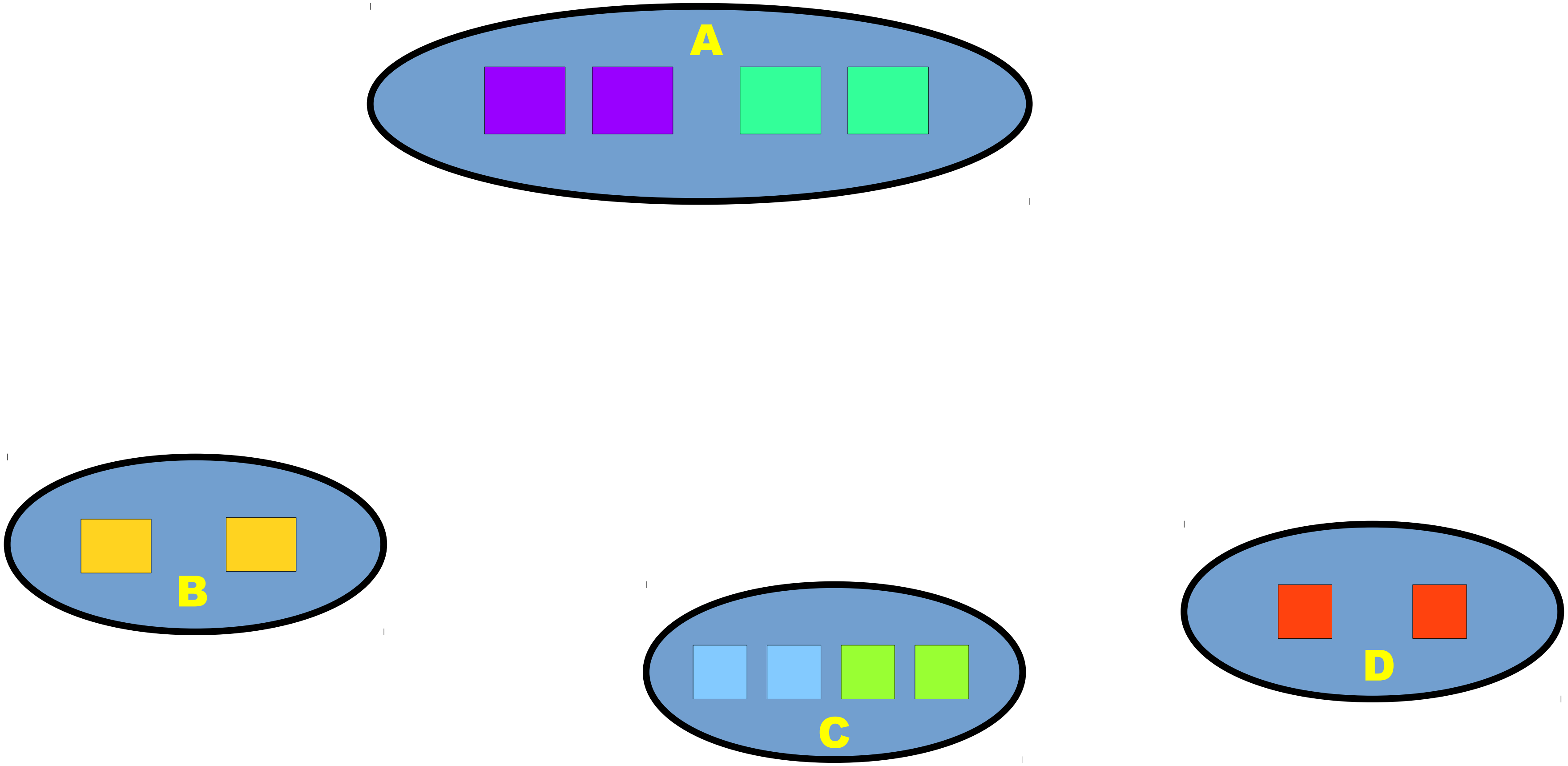
Hash Example: Slicing a simple parallel program by concerns



Hash Example: Slicing a simple parallel program by concerns

Hash Component Model

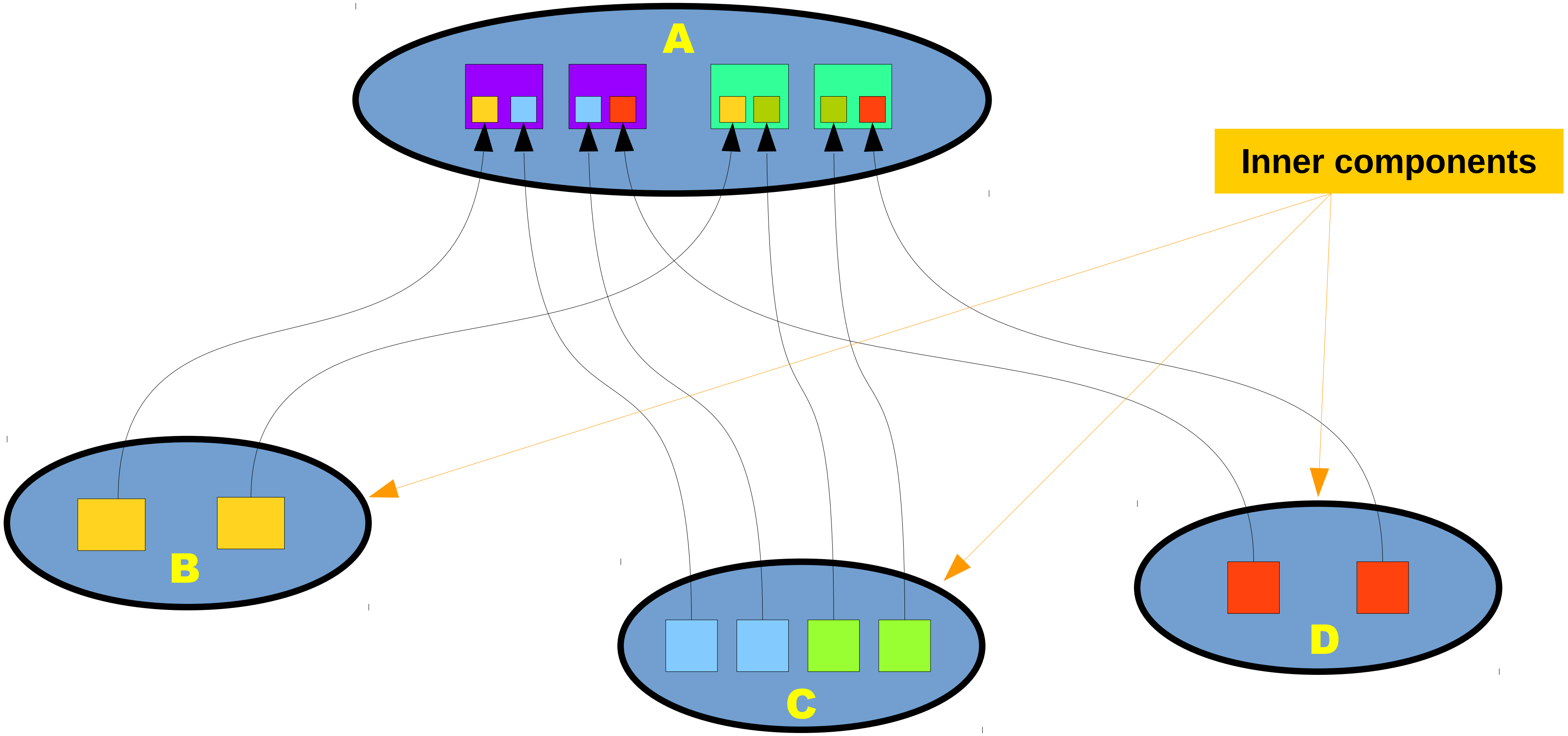
- A component model for distributed-memory parallel programs
- **Units** + overlapping composition + component kinds



ParGO

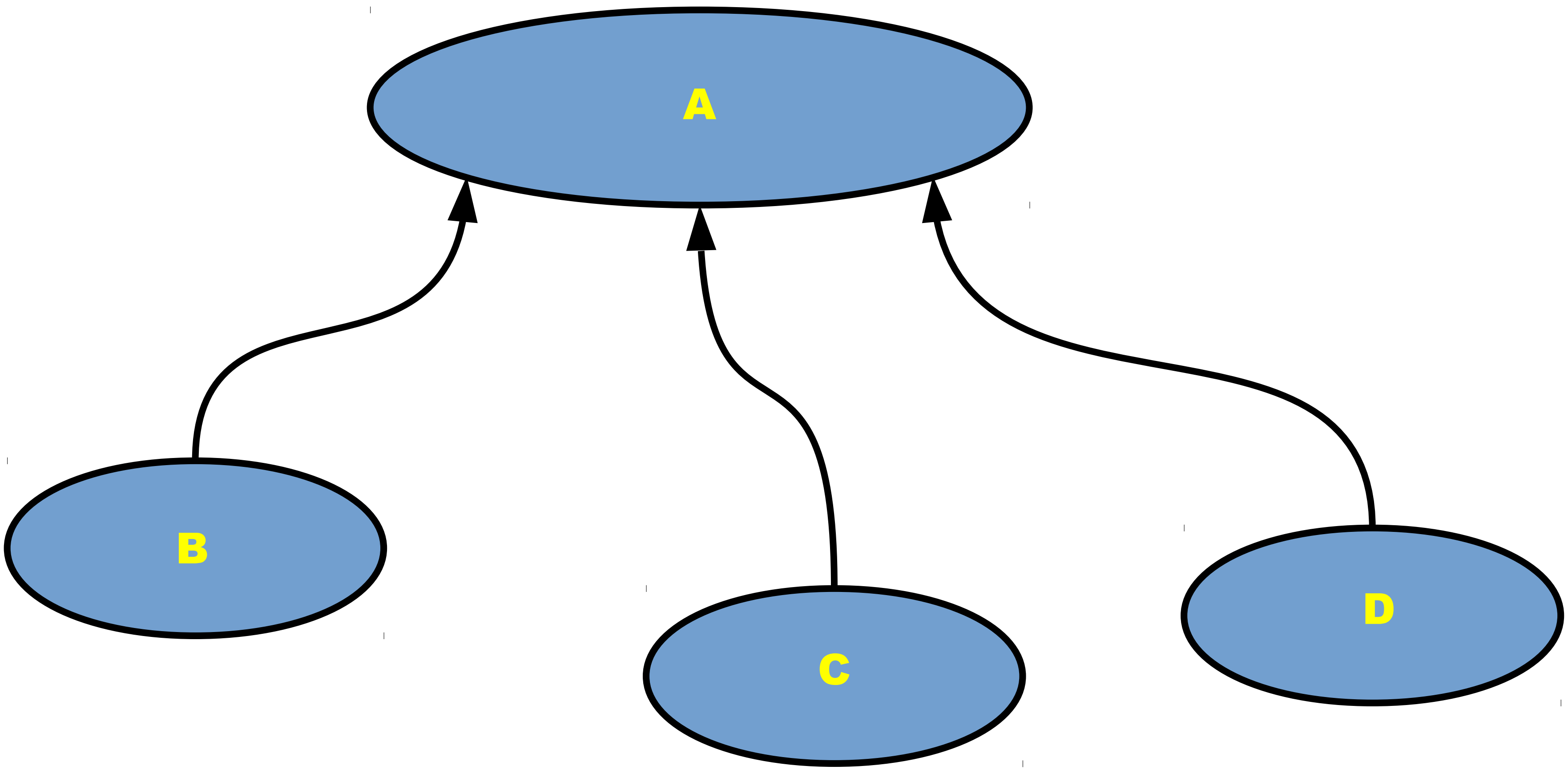
Hash Component Model

- A component model for distributed-memory parallel programs
- Units + **overlapping composition** + component kinds



Hash Component Model

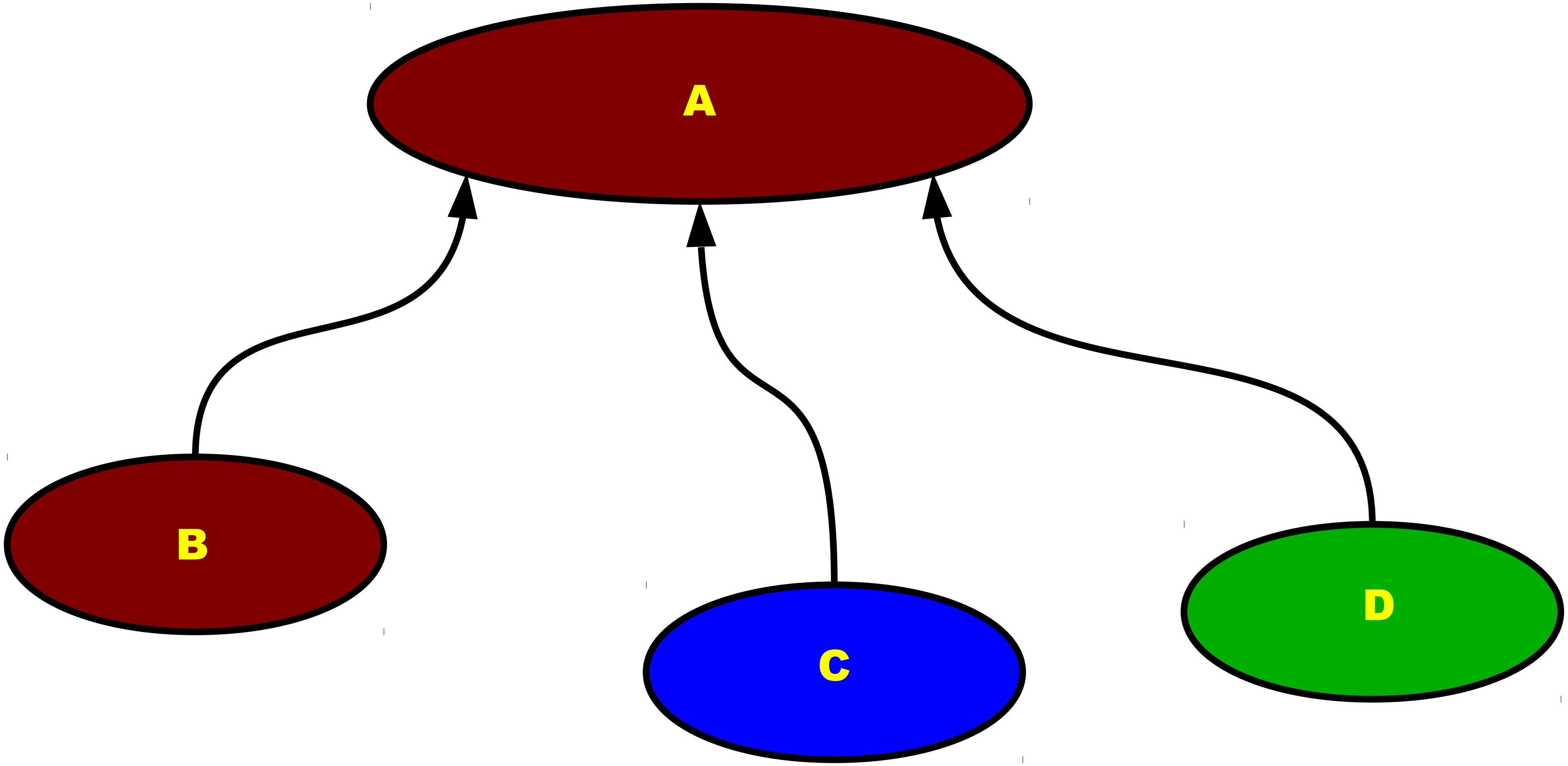
- A component model for distributed-memory parallel programs
- Units + overlapping composition + **component kinds**



ParGO

Hash Component Model

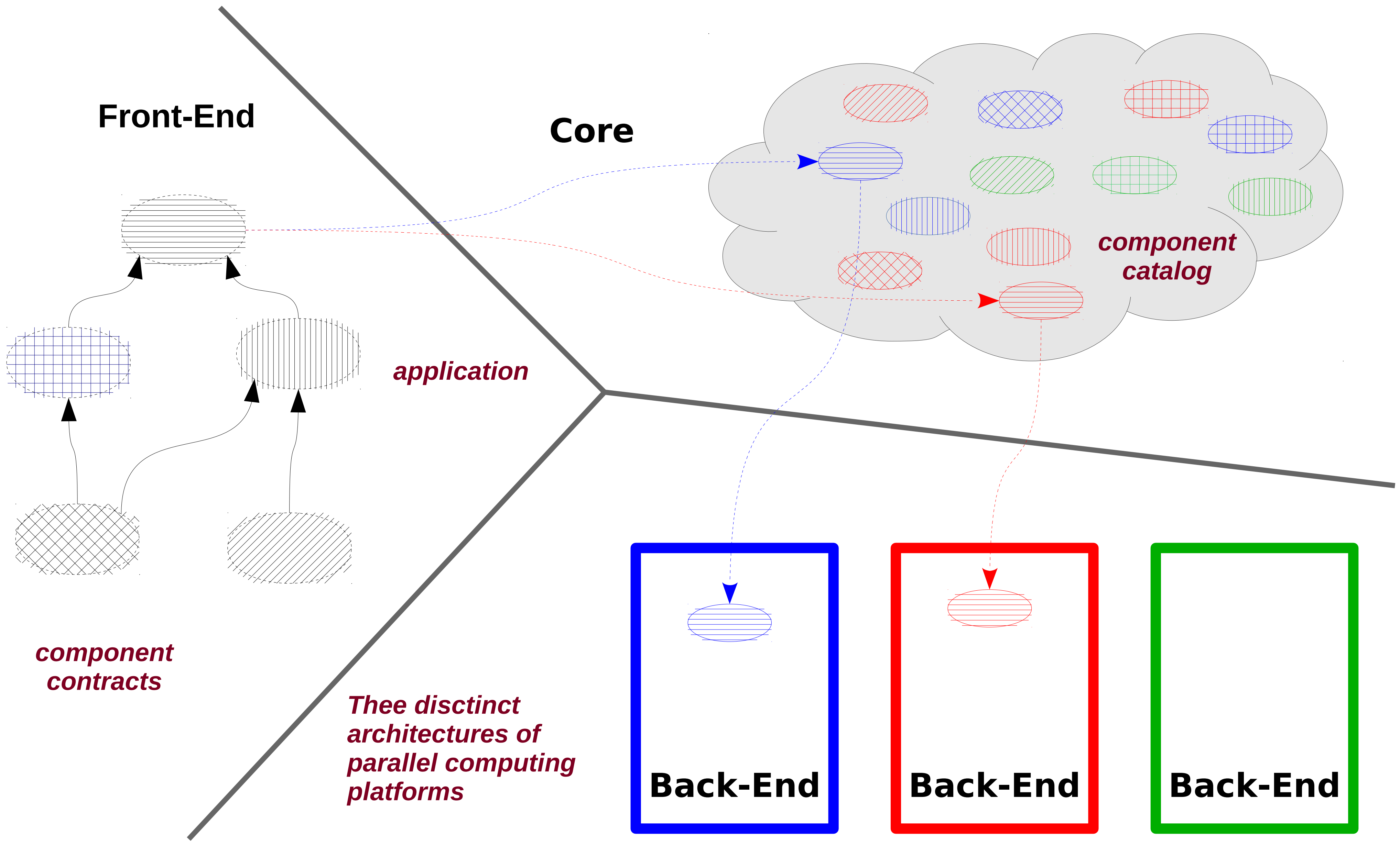
- A component model for distributed-memory parallel programs
- Units + overlapping composition + **component kinds**



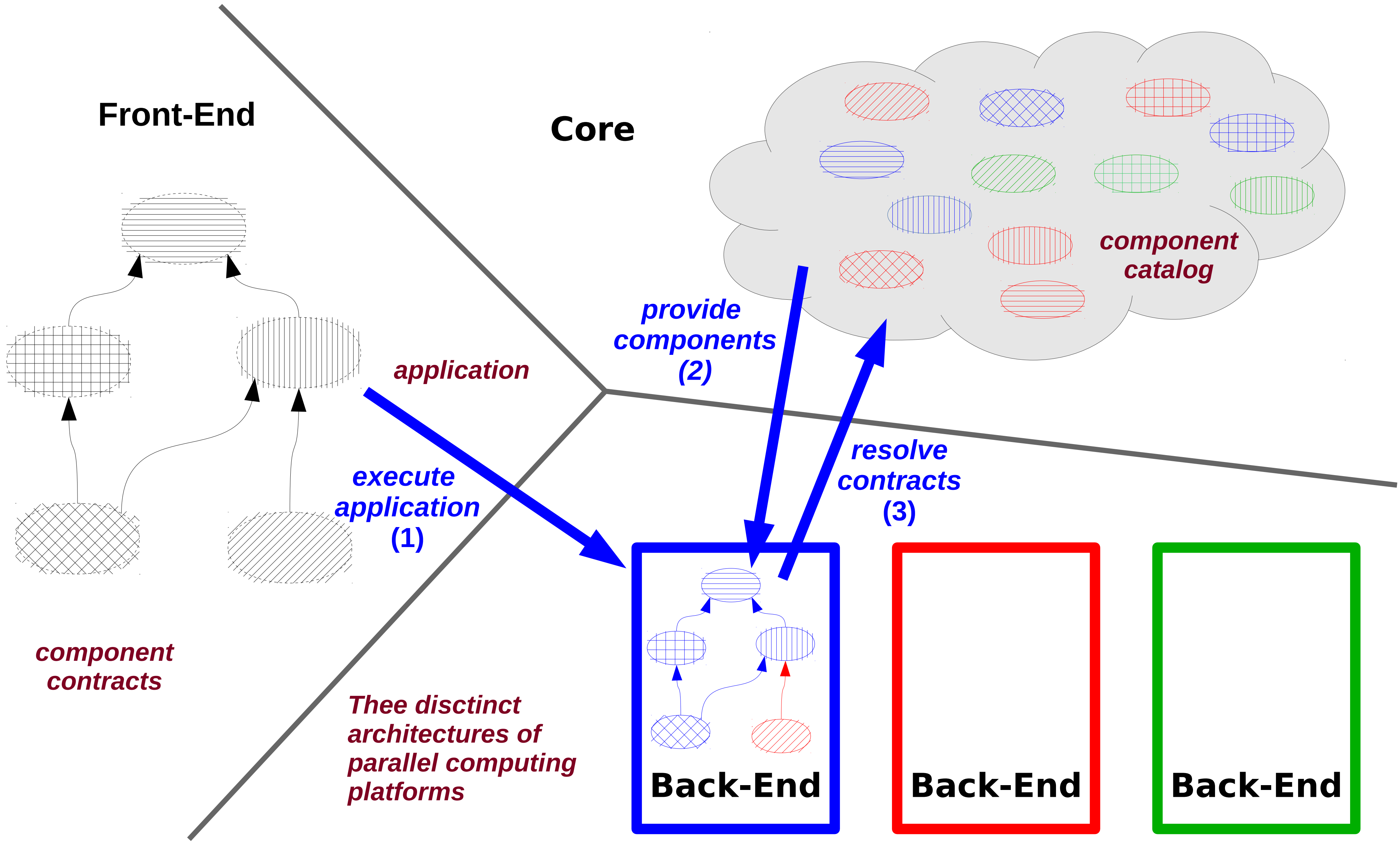
ParGO

Hash Programming Environment (HPE)

- A reference implementation of the Hash Component Model
  - https://code.google.com/p/hash-programming-environment/
  - Focus on **cluster** computing platforms
- Architecture: **Front-End**, **Core** and **Back-End**
- From the **Front-End**, programmers build new components by composition of <u>component contracts</u> retrieved form the **Core**, register them into the **Core**, and run applications in a parallel computing platform through the **Back-End** service;
- The **Core** is a <u>component catalog</u>, with tuned implementations for different application and execution platform contexts;
- When running an application, the **Back-End** looks at the **Core** for the best implementation of a parallel component for the architecture of the parallel computing platform it represents.



HPE Conceptual View



HPE Conceptual View

Hash Programming Environment (HPE)

- How to define components (contracts) that specify two things:
  - The concern to be addressed
  - The implementation assumptions about the execution context
    - execution context = parallel computing platform + application
    - goal: select the best component for each context
- For component reuse, the programmer details the concern and the contextual parameters (**Abstract Component**)
  - HPE finds the closest **concrete component** available (actual code)

ParGO

Hash Programming Environment (HPE)

$\textsc{LinearSystemSolver}$

$[\textit{\textbf{accelerator\_type}} = A\text{: } \textsc{AcceleratorType}, \textit{\textbf{multicore\_support}} = M\text{: } \textsc{MultiCoreSupport}$

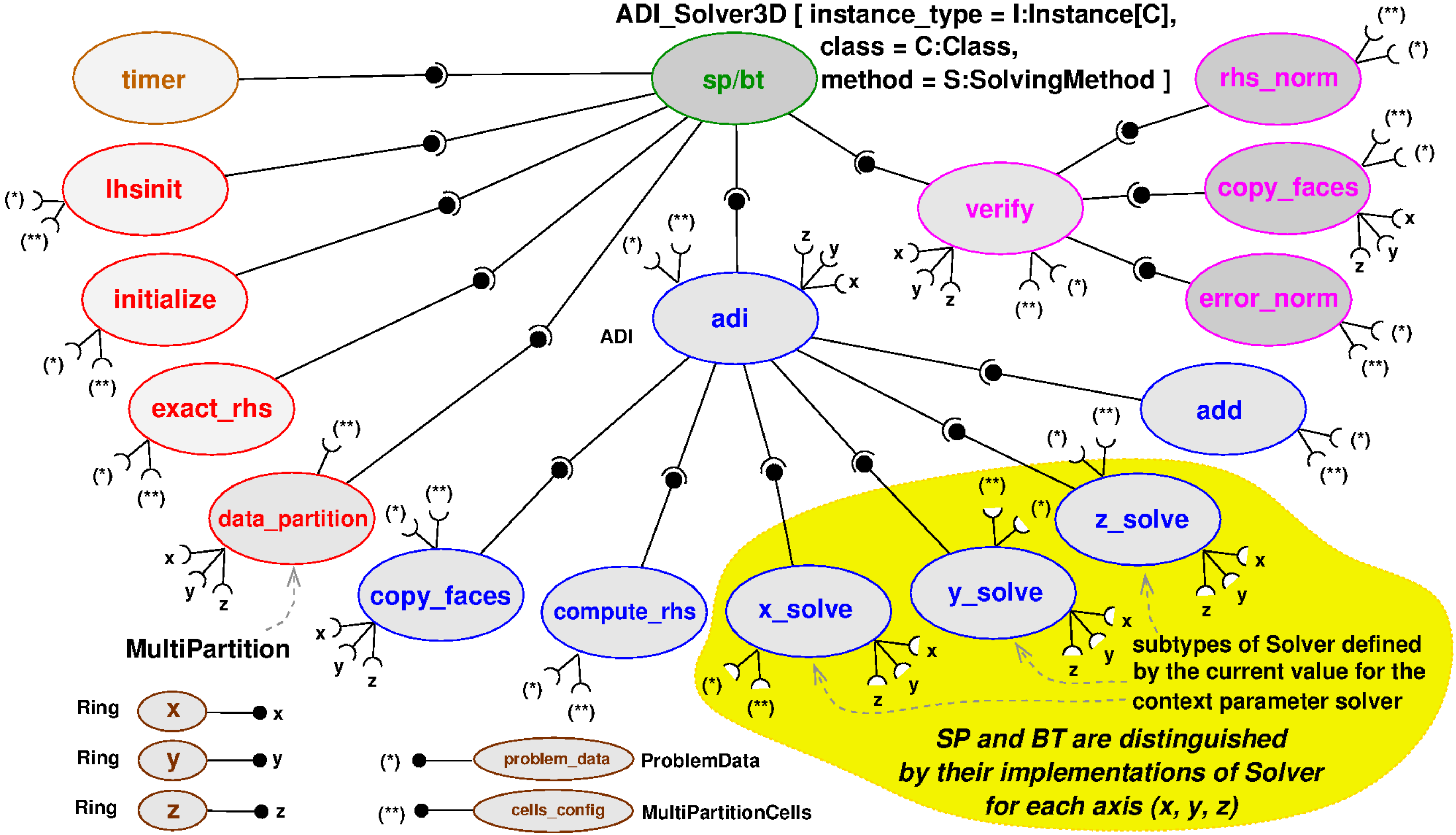
$\textit{\textbf{matrix\_pattern}} = P\text{: } \textsc{MatrixPattern}, \textit{\textbf{matrix\_partition}} = R\text{: } \textsc{MatrixPartition}\ [\textit{\textbf{multicore}} = M]$

$\textit{\textbf{matrix\_type}} = T\text{: } \textsc{MatrixType}\ [\textit{\textbf{property}} = P, \textit{\textbf{partition}} = R]]$

An abstract component signature with context parameters.

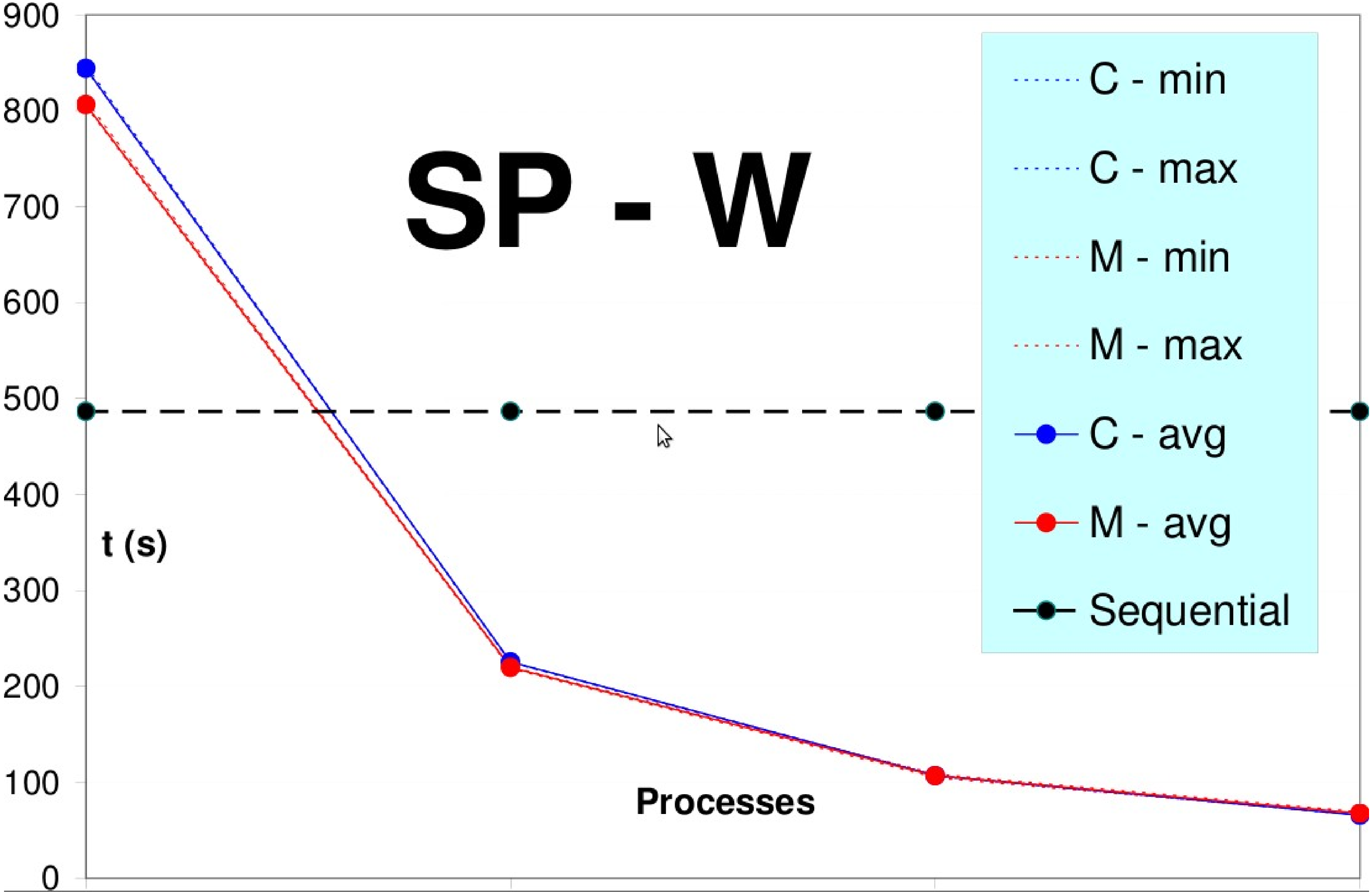
Hash Programming Environment (HPE)

- Evaluation
  - Implementing the NAS Parallel Benchmark (NPB) on HPE
  - Programs implemented:
    - FT, LU, SP and BT
  - Problem Classes
    - W and A
  - Comparison between the Fortran code version translated to C# and a Component-Based version
  - Castanhão Cluster
    - 16 nodes with 2 Intel Xeon 1.8 Processor
    - 32 GB RAM Total
    - Gigabit Ethernet
    - GCC Compiler
    - Mono 2.4



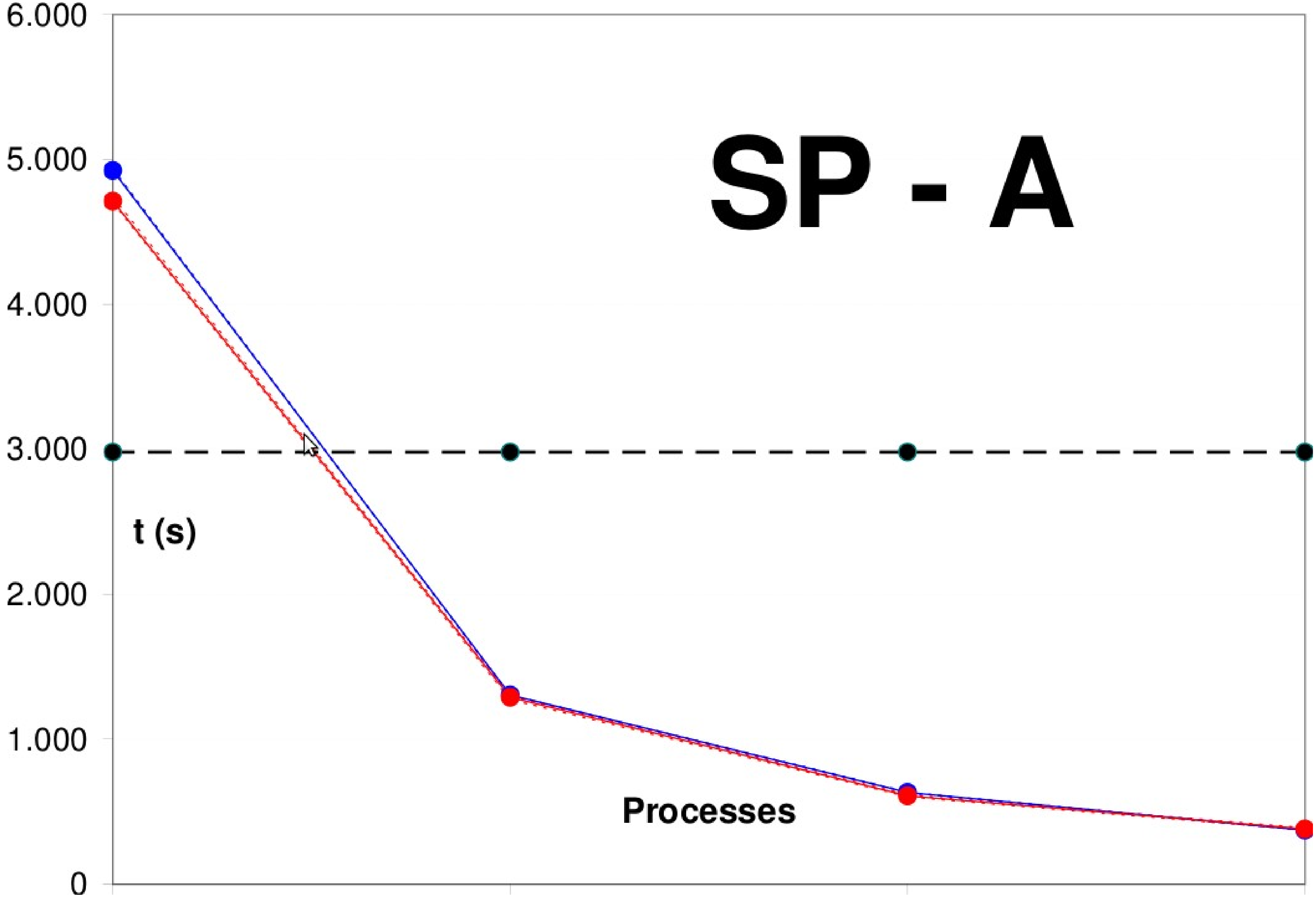
Decomposing SP and BT in Components

Hash Programming Environment (HPE)



ParGO

Hash Programming Environment (HPE)



Hash Programming Environment (HPE)

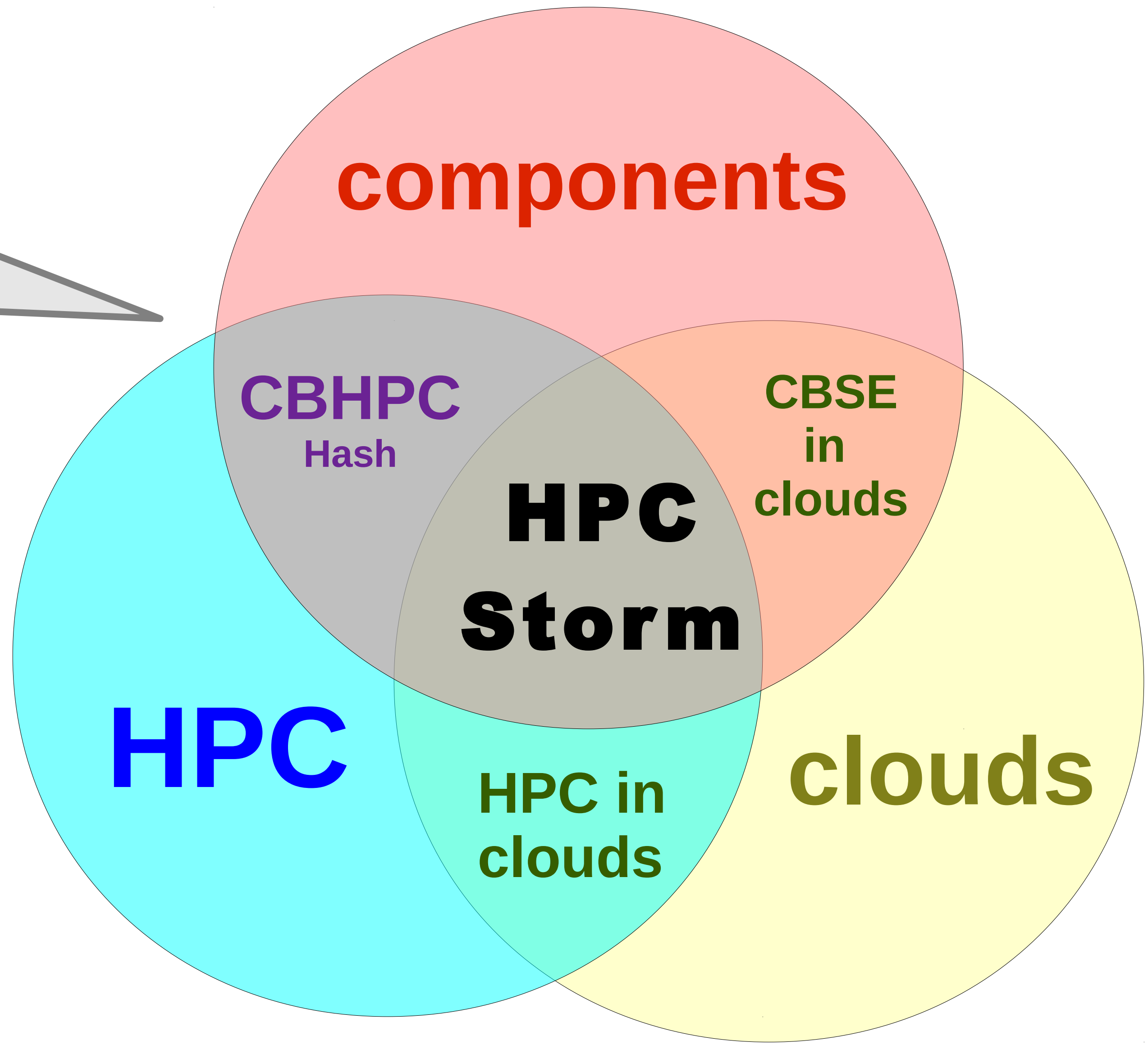
- Conclusions
  - The Hash Component Model provides a better way for code organization
  - Overhead due to a component-based architecture may be negligible
- Future Work
  - Cloud architectures (see next...)
  - Modeling other HPC applications:
    - Map-Reduce Algorithms
    - Graph Algorithms

ParGO

HPC Storm

HPC Storm *is*

HPC *in* **clouds** *through* **components**



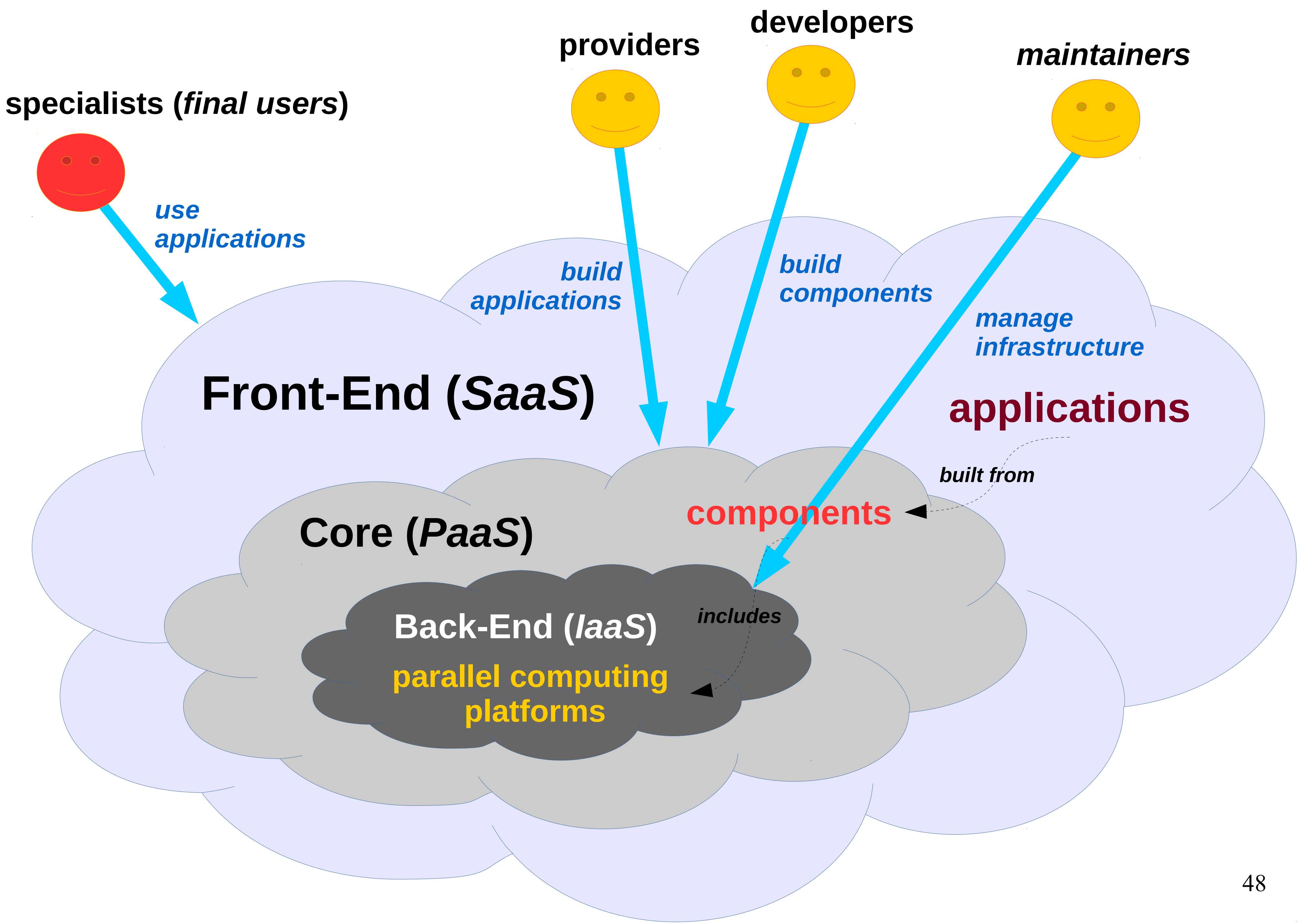
HPC applications

- *computational sciences*
- *engineering*
- …

ParGO

HPC Storm

- Services
  - **IaaS (infrastructure)**: comprising parallel computing platforms
  - **PaaS (platform)**: for developing components and applications that may exploit the potential performance of these parallel computing platforms
  - **SaaS (software)**: built from components, for attending HPC users
- Stakeholders
  - Domain **Specialists**
  - Application **Providers**
  - Component **Developers**
  - Platform **Maintainers**
- Architecture
  - Front-End, Core, Back-End



specialists (final users)
providers
developers
maintainers
use applications
build applications
build components
manage infrastructure
Front-End (SaaS)
applications
built from
Core (PaaS)
components
includes
Back-End (IaaS)
parallel computing platforms

HPC Storm

- Current Status
  - A new **Core** enhanced with ontological resource description for clouds components
    - Phd Student: Wagner Al-Alam
- Ongoing Work
  - A redesigned **Front-End** for the cloud, with support for Workflows, Domain-Specific Languages, etc
    - Phd Student: Jefferson Carvalho
  - **Back-End** with support for component adaptation with Elasticity, scale-out/scale-in virtual nodes
    - Phd Student: João Marcelo

Publications

- de Carvalho-Junior, Francisco Heron ; REZENDE, C. A. . **A case study on expressiveness and performance of component-oriented parallel programming. Journal of Parallel and Distributed Computing** (Print), v. 73, p. 557-569, 2013.
- de Carvalho Junior, Francisco Heron ; REZENDE, C. A. ; SILVA, J. C. ; MAGALHAES, F. J. L. ; JUACABA NETO, R. C. . **On the Performance of Multidimensional Array Representations in Programming Languages Based on Virtual Execution Machines**. In: XVII Simpósio Brasileiro de Linguagens de Programação, 2013, Brasília. Lecture Notes in Computer Science - Proceedings of the XVII Brazilian Symposium on Programming Languages. Berlim: Springer, 2013. v. 8129. p. 31-45
- de Carvalho Junior, Francisco Heron ; Rezende, Cenez Araujo ; SILVA, J. C. ; Al-Alam, Wagner . **Contextual Abstraction in a Type System for Component-Based High Performance Computing Platforms.** In: XVII Simpósio Brasileiro de Linguagens de Programação, 2013, Brasília. Lecture Notes in Computer Science - Proceedings of the XVII Brazilian Symposium on Programming Languages. Berlim: Springer Berlin Heidelberg, 2013. v. 8129. p. 90-104.
- de Carvalho Junior, Francisco Heron ; Marcilon, T. B. . **Derivation and Verification of Parallel Components for the Needs of an HPC Cloud.** In: XVI Simpósio Brasileiro de Métodos Formais (SBMF'2013), 2013, Brasília. Lecture Notes in Computer Science - Proceedings of the XVI Simpósio Brasileiro de Métodos Formais (SBMF'2013). Berlim: Springer, 2013. v. 8195. p. 51-66.

Thank you!