

Interfacing Different Application Types to the KOALA Grid Scheduler



Hashim Mohamed, Alexandru Iosup, Ozan Sonmez, Jeremy Buisson and Dick Epema
DAS-3 Symposium, June 1, 2007



Outline

- KOALA grid scheduler
- Support for different application types
- Some experimental results

Grid scheduling environment (1)

- Grids offer access to large collections of resources (processors and data sets) for applications
- Ideally, submitting jobs to grids should be “submit and take a break”
- A grid scheduler should:
 - Find suitable execution site(s) possibly at multiple locations, i.e., co-allocation
 - Transfer the application and if required input files to the sites and run it
 - Return results

Grid scheduling environment (2)

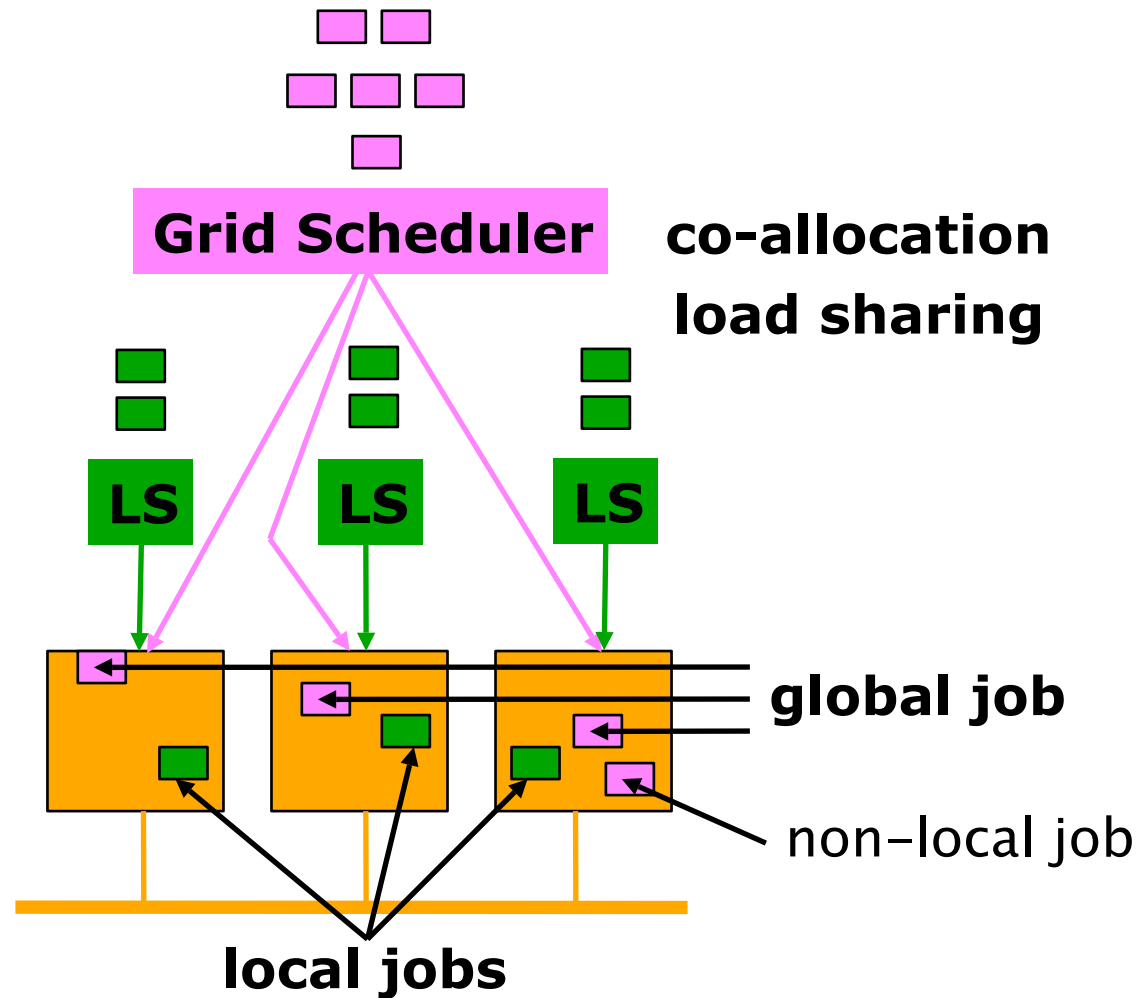
- Grid schedulers usually do not own resources themselves
- Grid schedulers have to interface to different local schedulers, e.g.:
 - Sun Grid Engine (SGE 6.0) on DAS-2/DAS-3
 - OAR on Grid'5000
- Workload
 - Various kind of applications
 - Various requirements

Scheduling in Grids

global queue
with grid
scheduler

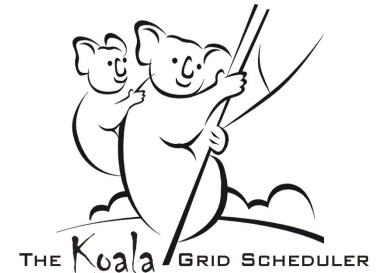
local queues
with local
schedulers

clusters

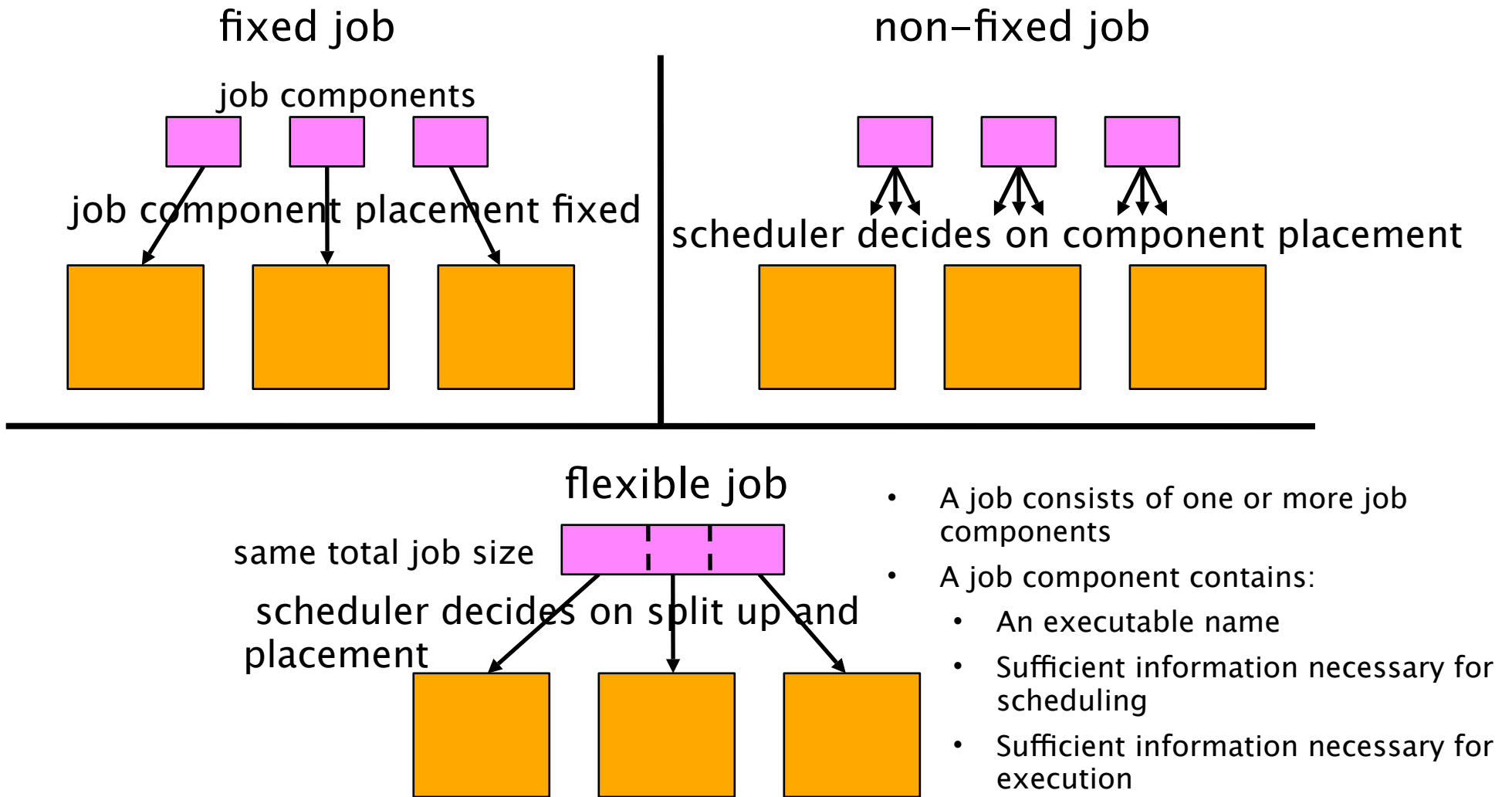


KOALA Grid Scheduler

- Developed in the DAS system
- Has been deployed on the DAS-2 in September 2005
- Ported to DAS-3 in April 07
- Its scheduler is independent from grid middlewares such as Globus
- Runs alongside local schedulers
- Main research goals:
 - Processor co-allocation in grids
 - Load sharing: in the absence of co-allocation
 - Automating the execution of different application types
 - Fault tolerance

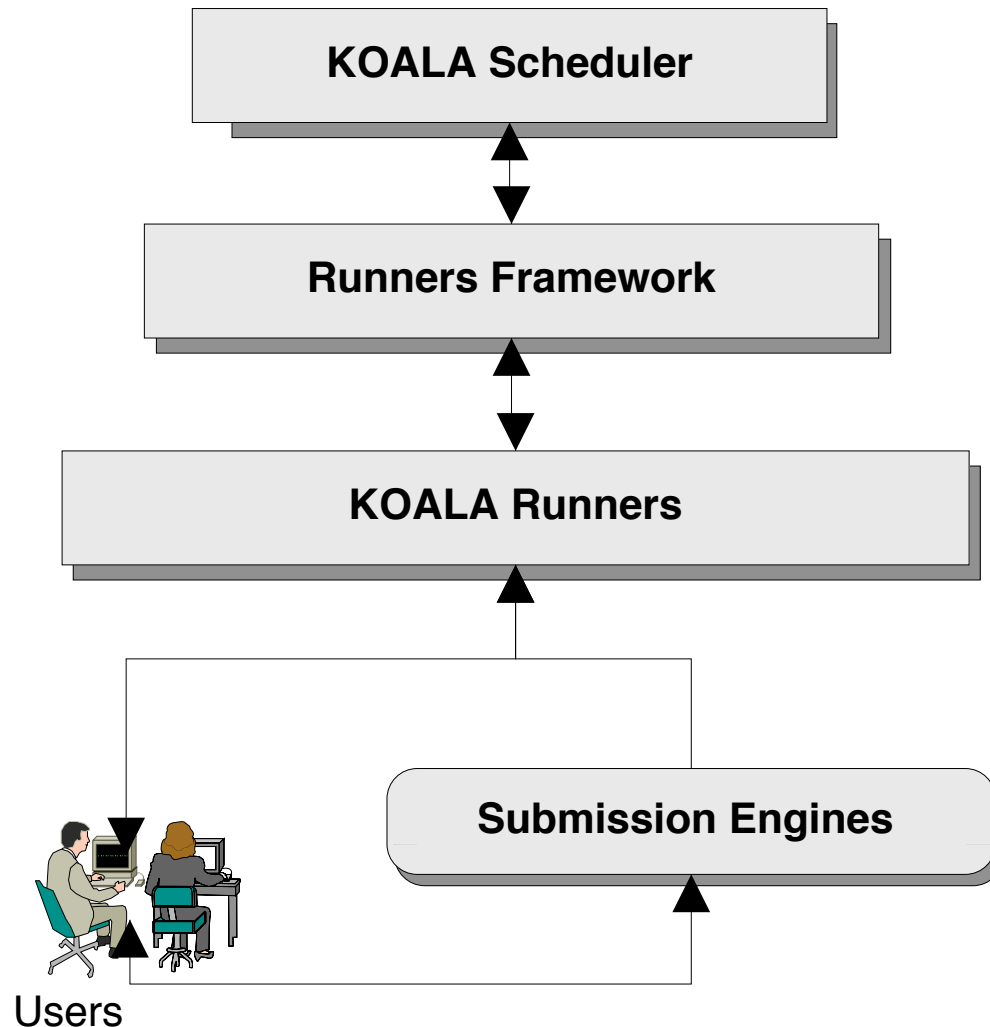


Background (1): KOALA job requests



- A job consists of one or more job components
- A job component contains:
 - An executable name
 - Sufficient information necessary for scheduling
 - Sufficient information necessary for execution

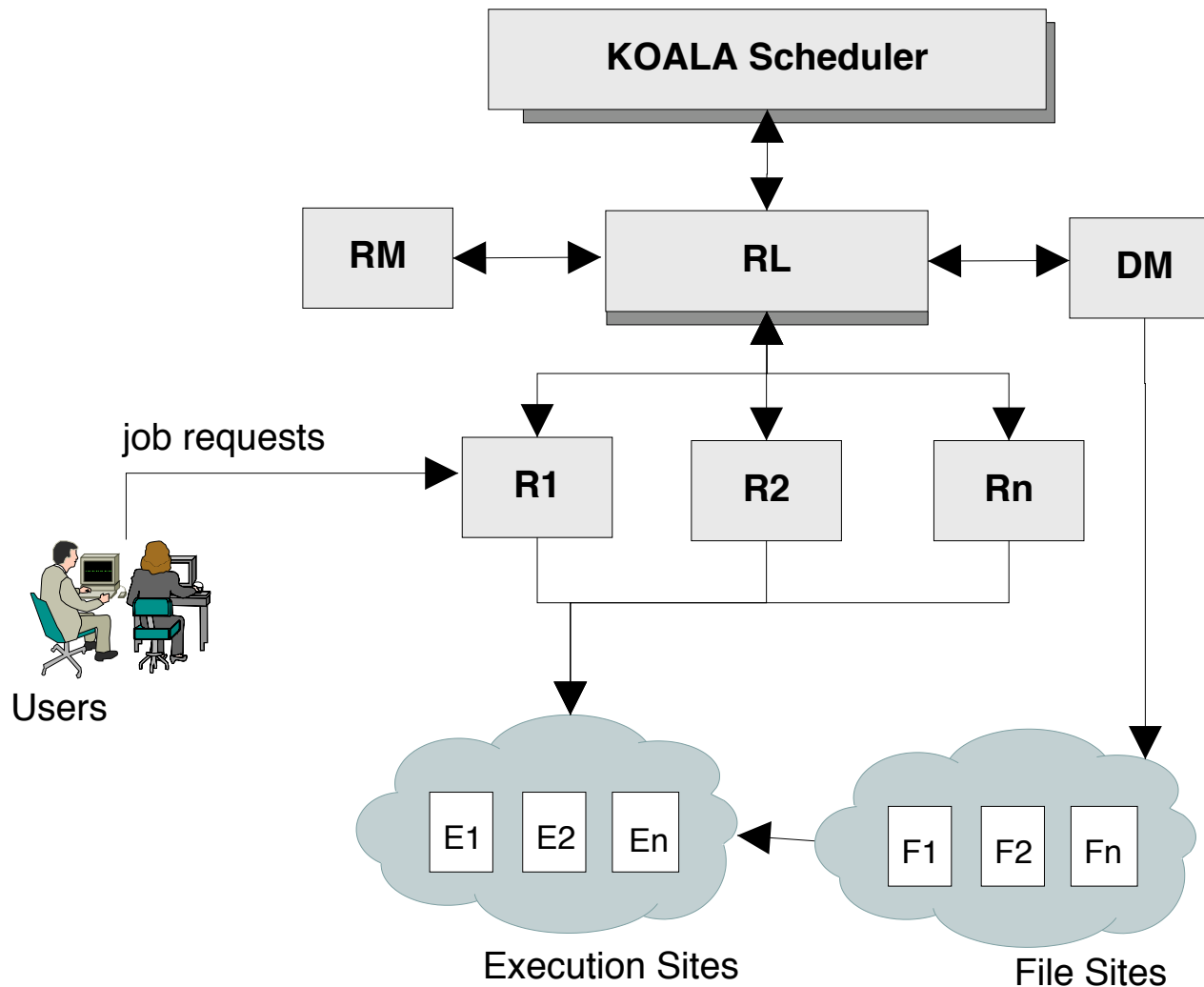
Background (2): KOALA Layered Architecture



Grid applications

- Different application types with different characteristics exist
- Important application types include:
 - Parallel applications
 - Parameter sweep applications
 - Workflows
 - Data intensive applications
- Challenges:
 - Applications have special characteristics and needs
 - Grid infrastructure is highly heterogeneous
 - Grid infrastructure configurations issues
 - Grid resources are highly dynamic

The KOALA Runners Framework



- RM: run monitor
- RL: runners listener
- DM: data manager
- Ri: runners

Fault Tolerance

- RM monitors three groups of errors:
 - **Hard errors**: caused by OS and network failures of the submission site, and grid middleware errors
 - **Default operation**: Abort the job and the runner
 - **Soft errors**: caused by execution-site specific errors
 - **Default operation**: Abort the failed job components and ask the scheduler to place them somewhere else
 - **Application-specific errors**: Bugs in the application
 - **Default operation**: Abort the job and the runner
- The runners can override the default operation
- Note: The runners are informed before the default operation is carried out

KOALA Runners

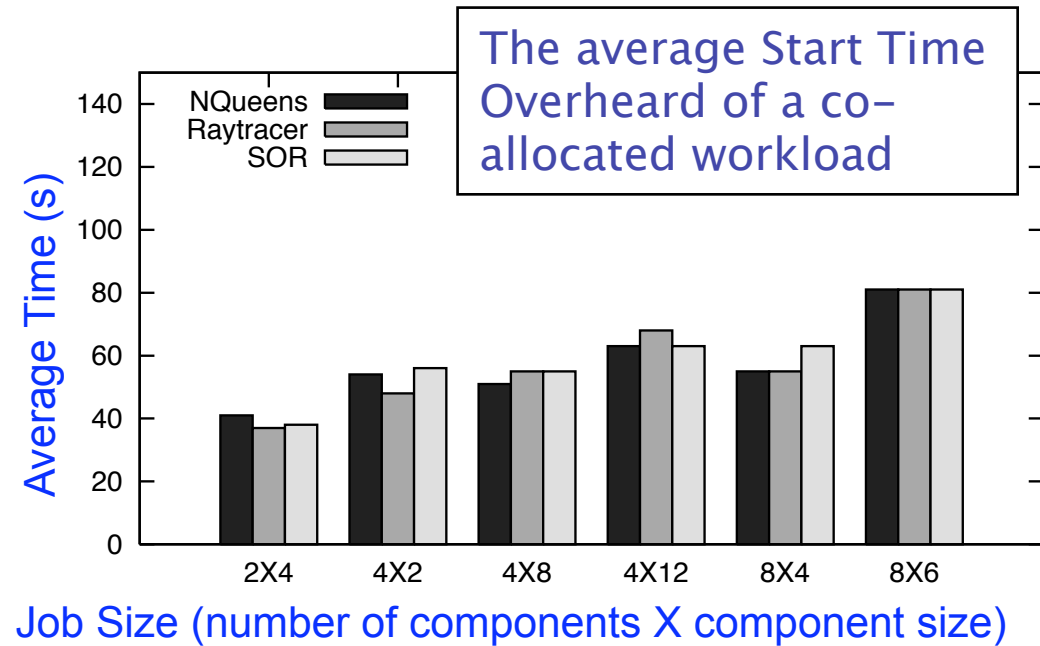
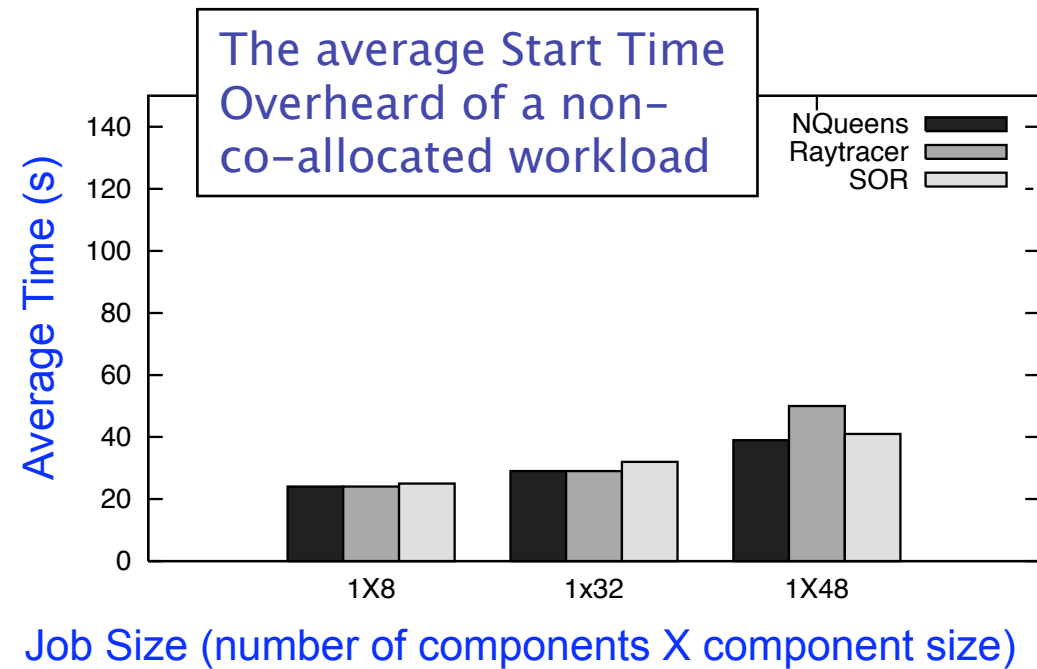
- Extends support for different job types
- Runners currently operational in the DAS-2/DAS-3:
 - DRunner: Globus co-allocation tool (only DAS-2)
 - KRunner: KOALA default co-allocation tool
 - IRunner: KRunner based Ibis job submission tool
 - DYCORunner: For malleable jobs based on the DYNACO framework
- Runners allow the job submission from the DAS as well as non-DAS machines, e.g., your desktop machines

The IRunner

- Designed to run Ibis applications
- IRunner is based on the Krunner
- IRunner can start a nameserver or use the one that has been started already

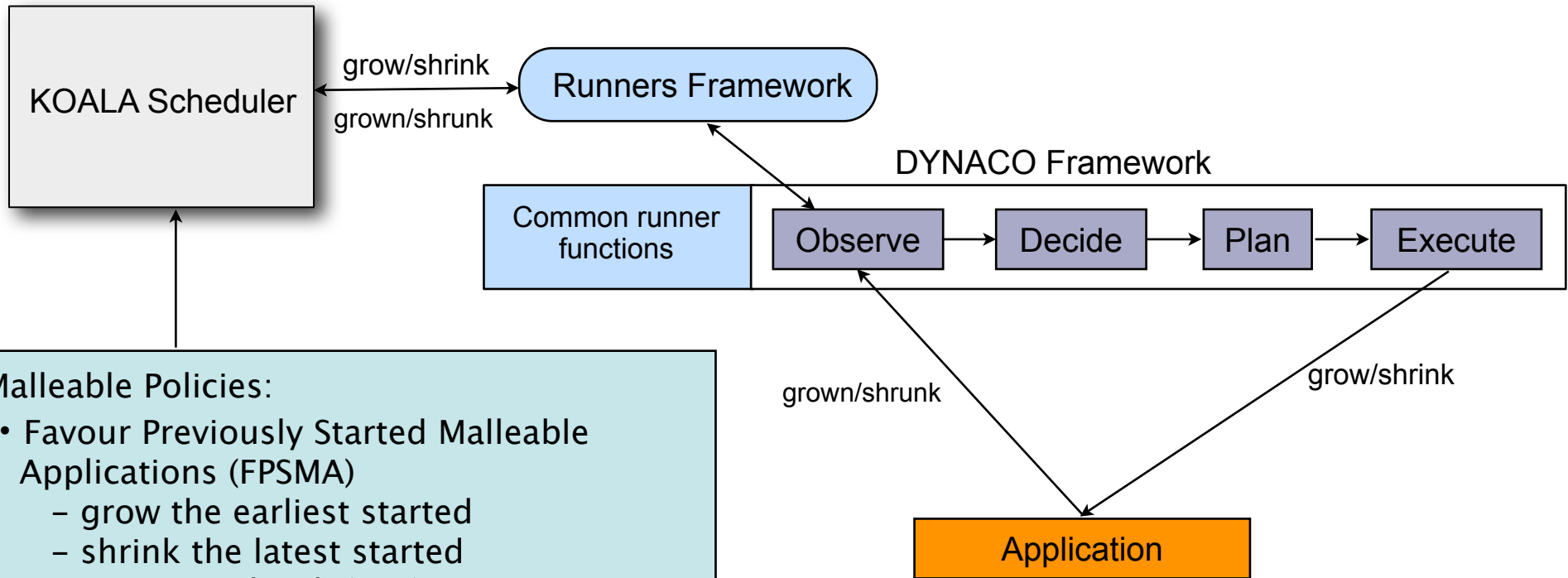
IRunner: some experimental results

- Experiments performed in DAS-2
- Metric:
 - Start Time Overhead: time incurred from when the runner starts deploying a job for execution until the time the job is actually running



DYCORunner

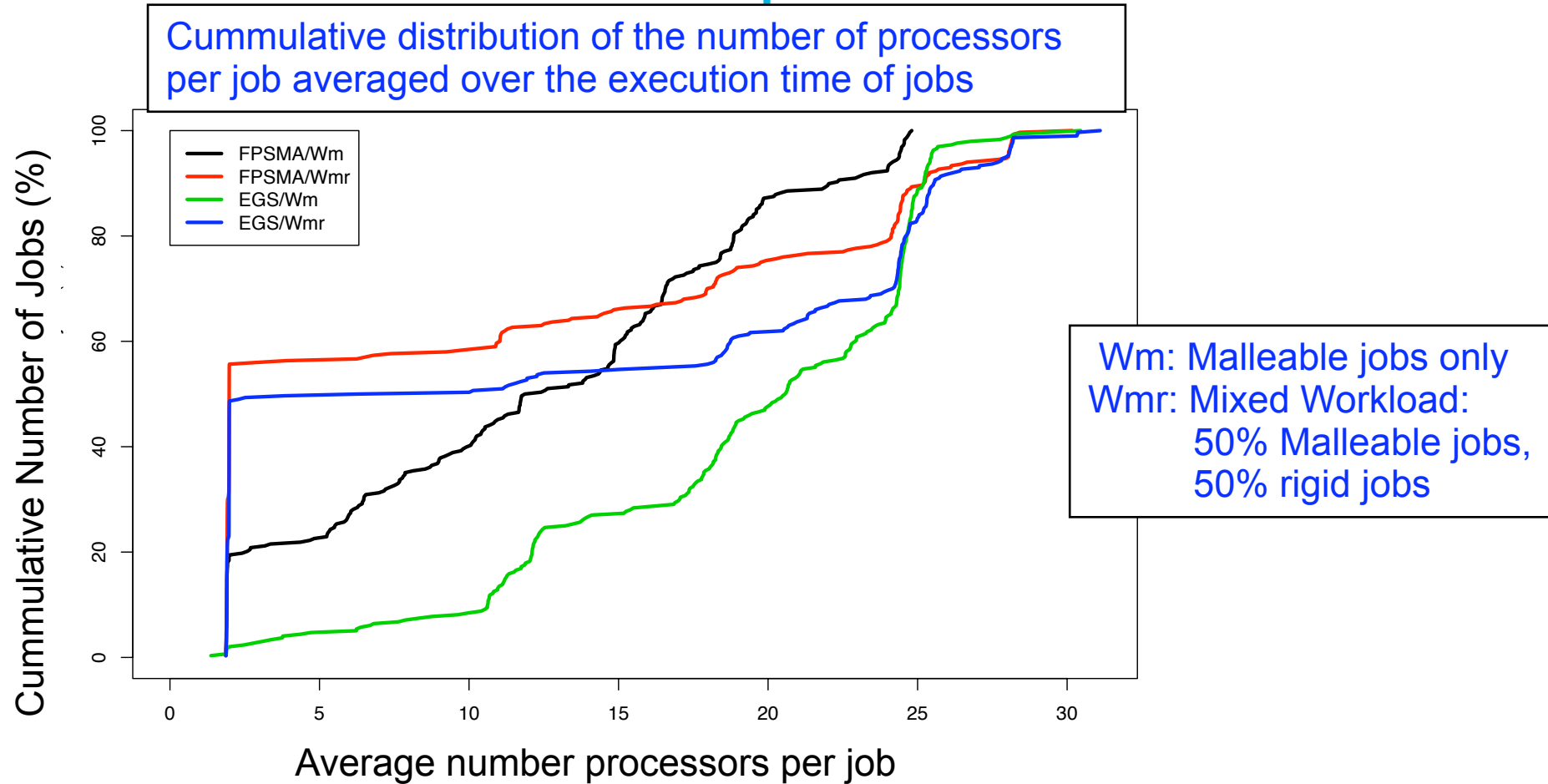
- DYNACO is a framework for building dynamically adaptable applications; more on <http://dynaco.gforge.inria.fr>



• Malleable Policies:

- Favour Previously Started Malleable Applications (FPSMA)
 - grow the earliest started
 - shrink the latest started
- Equi-Grow & Shrink (EGS)
 - Distribute total amount of growing/shrinking equally among running jobs

DYCORunner: some experimental results



- Experiments done in DAS-3
- Initial job size is set to 2
- Allow jobs only to grow
- More Malleable jobs grow with EGS than FPSMA

Conclusions and Future Work

- KOALA runners framework and some runners have been presented
- Our experiences show the correct and reliable operation of KOALA in the DASes
- More tests with the DYCORunner
- Add a runner for Parameter Sweep Applications
- Experiments on heterogeneous testbed (DAS-2, DAS-3, Grid'5000)

<http://www.st.ewi.tudelft.nl/koala>