



Ibis Deploy and Coupled Codes

Niels Drost

Computer Systems Group
Department of Computer Science
VU University, Amsterdam, The Netherlands



Deployment

- How to get your application running in the Jungle
- For each resource used:
 - Find resource
 - Reserve resource
 - Copy input files
 - Run application
 - Copy back output files



Problems

- Large number of steps required to deploy an application (Jobs, Hubs, Server, etc)
- Middleware level interface too low level for users
- Deploying an application requires the user to write **another application!**
- Users want to simply “press a button” to deploy

Deployment is not very user friendly



We can do better

- All these steps is not what the user wants
- Deploy in a “single step”
- Idea: Make some assumptions
 - Application uses the IPL, and SmartSockets
 - All files initially on local machine
- Application can still use any executable to do the actual work

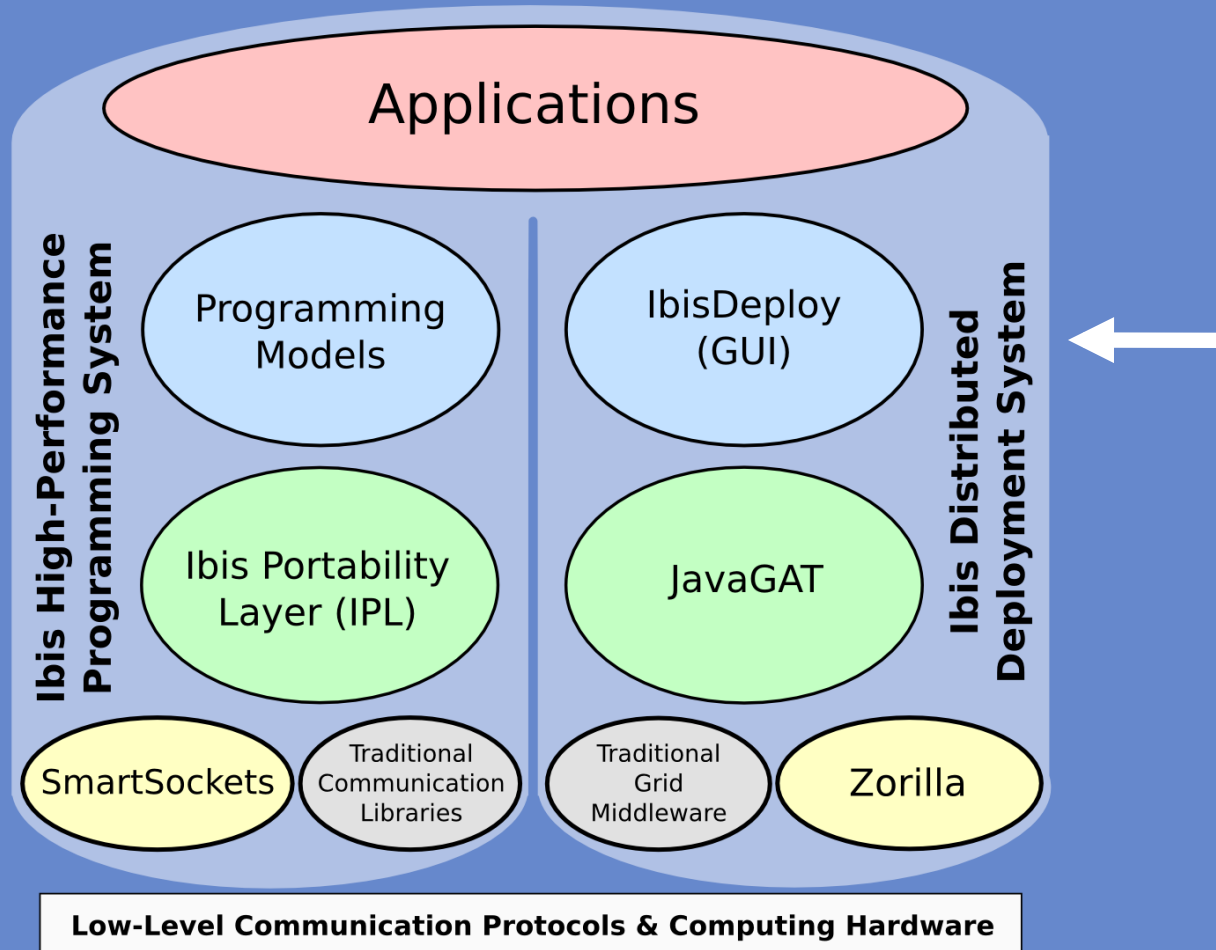


Ibis-Deploy

- Library for deploying IPL applications
- Simplify, Simplify, Simplify!
- Also deploys SmartSockets hubs and IPL registry
- Uses Java property files for configuration
 - Simplest configuration files yet, I promise!
- Comes with an (optional) GUI



Where are we?





Experiment Editor



Pool Name: demo

Select Application

Server

x

1

Select Cluster

VU

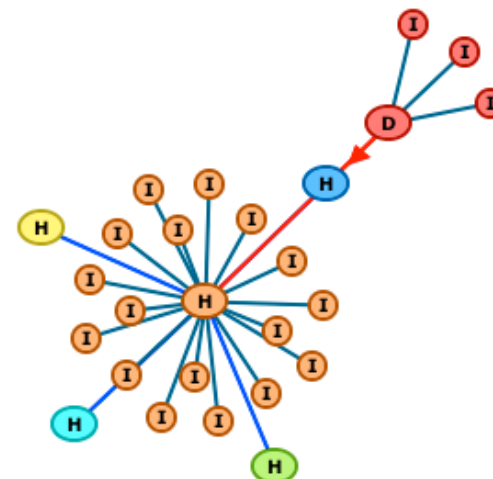
x

1

+ Create

➔ Create & Start

Experiment Monitor



	pool	name	job status	hub status	cluster	middleware	application	process count	resource count	output
<input type="checkbox"/>	demo	Client	DEPLOYED	DEPLOYED	local	local	Client	1	1	<button>output</button>
<input type="checkbox"/>	demo	Delft-Server	INITIALIZING	DEPLOYED	Delft	ssh	Server	16	16	<button>output</button>
<input type="checkbox"/>	demo	Leiden-Server	INITIALIZING	DEPLOYED	Leiden	ssh	Server	16	16	<button>output</button>
<input type="checkbox"/>	demo	Local-Server	DEPLOYED	DEPLOYED	local	local	Server	1	1	<button>output</button>
<input type="checkbox"/>	demo	Multimedia...	INITIALIZING	DEPLOYED	MultimediaN	ssh	Server	16	16	<button>output</button>
<input type="checkbox"/>	demo	UvA-Server	INITIALIZING	DEPLOYED	UvA	ssh	Server	16	16	<button>output</button>
<input type="checkbox"/>	demo	VU-Server	DEPLOYED	DEPLOYED	VU	ssh	Server	16	16	<button>output</button>

➔ Start All

✖ Stop All

— Remove All

➔ Start Selected

✖ Stop Selected

— Remove Selected

Property files

- Grid file: description of all resources (clusters)
 - Hostname of frontend, JavaGAT adaptor used, Geolocation, etc
- Application file: description of applications
 - Main class, jars needed, arguments, etc
- Experiment file: description of experiment
 - “run application A on 32 nodes of cluster B”
 - Alternative: specify in GUI
- Workspace: directory containing all 3 files



Pilot Job Example

- Identical to Pilot Job Framework
- **Minus** all the JavaGAT code
- Application file:

```
JobServer.main.class = tutorial20.glue.JobServer
JobServer.arguments = --executable convert --arguments
    -adaptive-blur 100%
    --input /Users/niels/workspace/tutorial/images
    --output /Users/niels/workspace/tutorial/output-images
JobServer.libs = lib-server, lib

Pilot.main.class = tutorial20.glue.PilotJob
Pilot.libs = lib-server, lib
```

Grid File

```
VU.server.adaptor = SshTrilead
VU.server.uri = ssh://fs0.das4.cs.vu.nl
VU.job.adaptor = SshTrilead
VU.job.uri = ssh://fs0.das4.cs.vu.nl
VU.file.adaptors = SshTrilead,Local
VU.java.path = java
VU.job.wrapper.script = das4.script
VU.latitude = 52.3328
VU.longitude = 4.8669
```

- Note: Grid file does not change very often



Ibis Deploy Demo



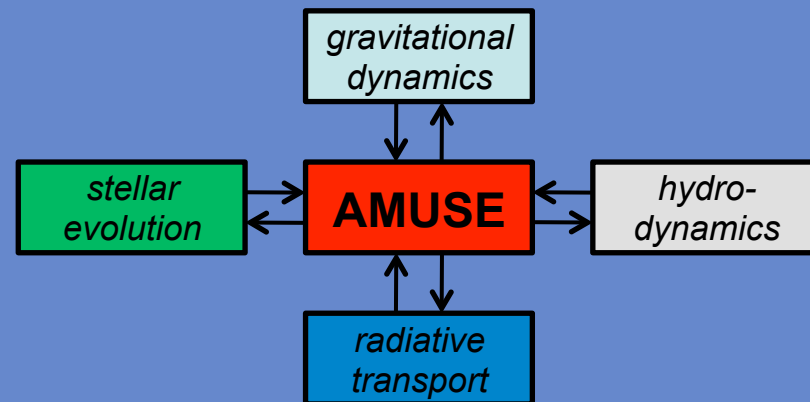
Coupled Codes

- JavaGAT + IPL + SmartSockets + Ibis Deploy allows us to run more complicated parallel codes as well
- Parallel codes on multiple resources
 - One big IPL application on an entire Jungle
- Coupled codes using multiple resources
 - Lots of different smaller applications tied together with the IPL.

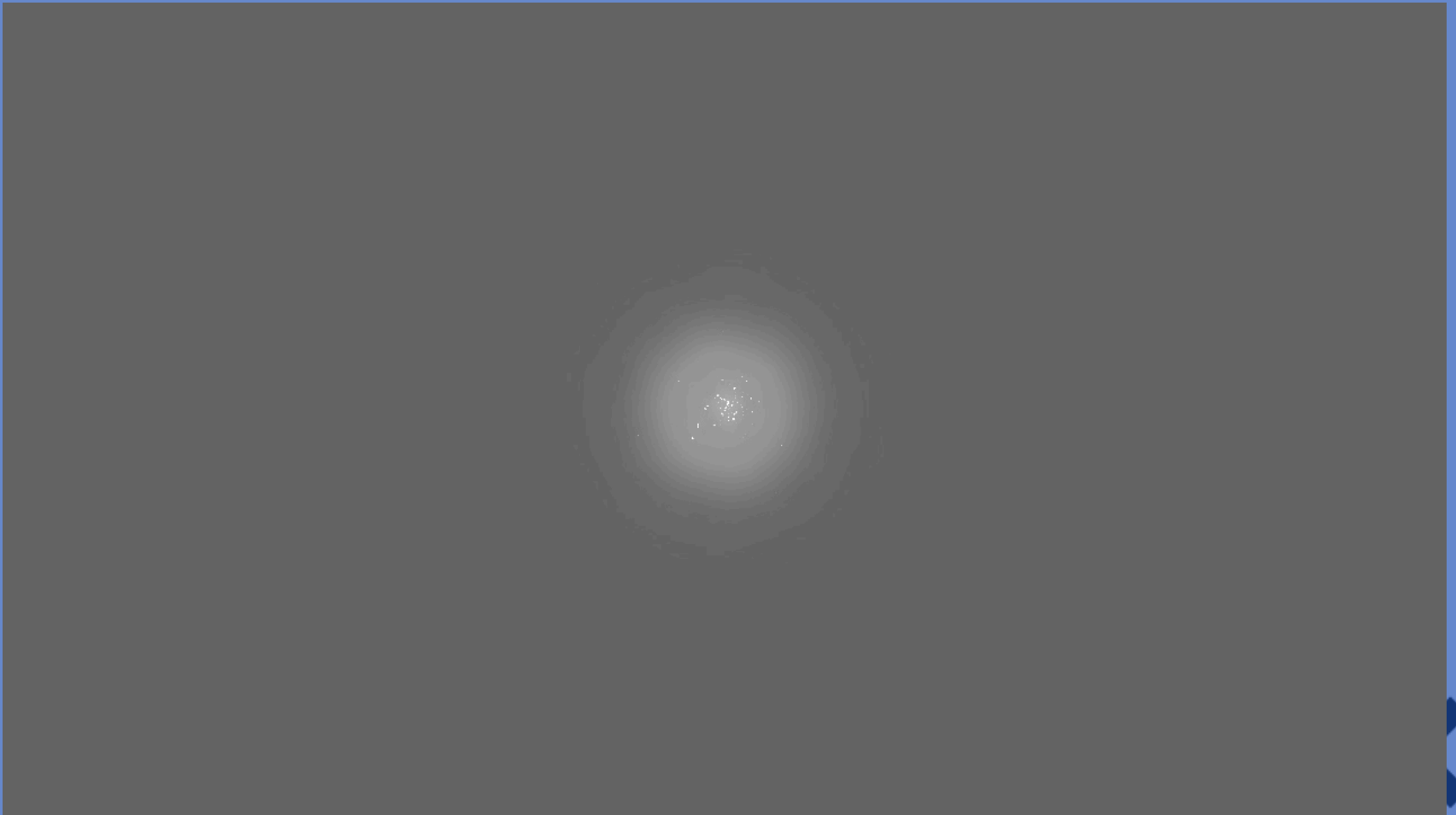


AMUSE

- Computational Astrophysics
- Simon Portegies-Zwart, Sterrewacht Leiden
- Framework in Python,
- Codes In Fortran 77, Fortran 2003, C, C++, CUDA, etc.



Example: Early Star Cluster



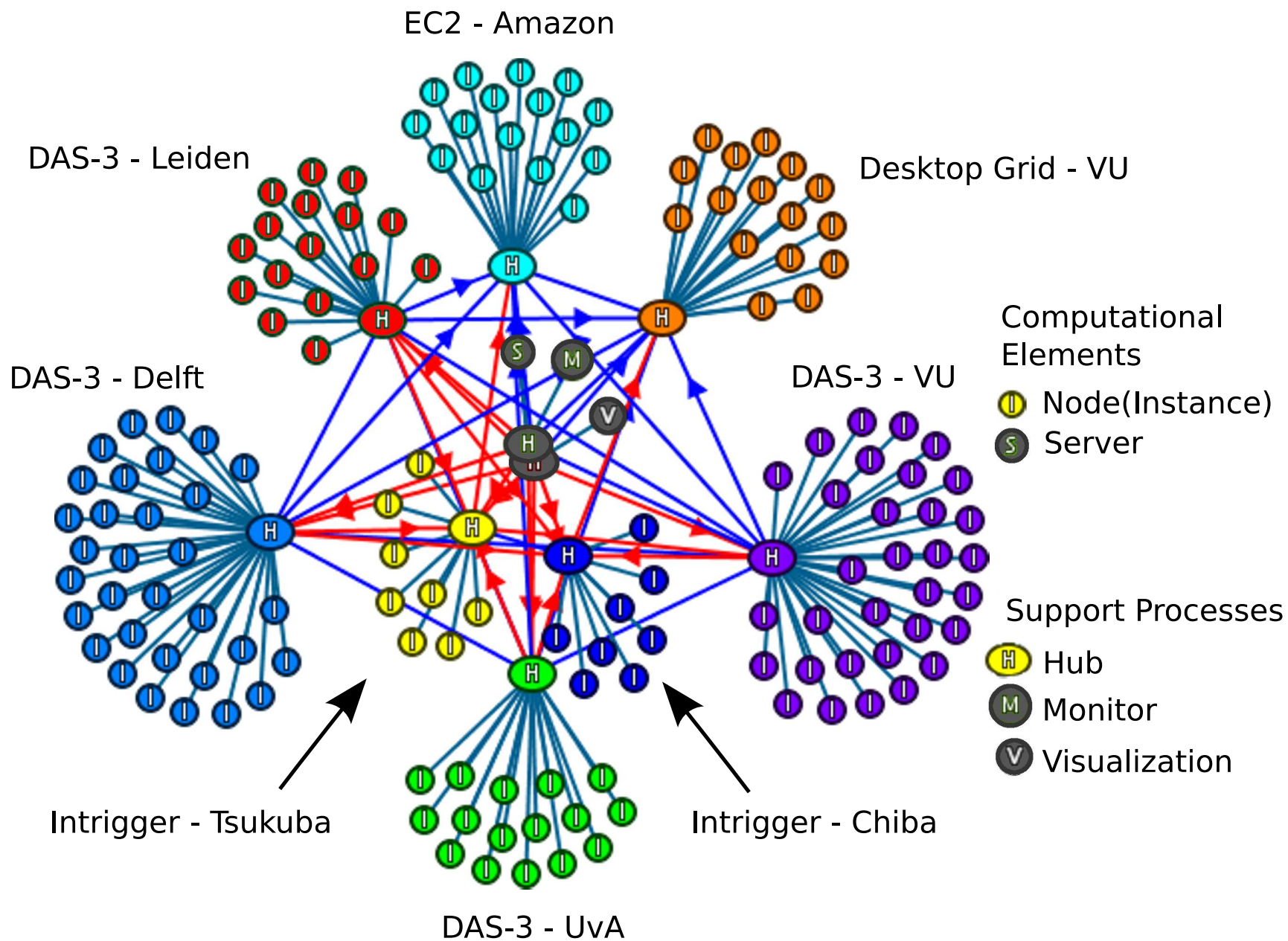
AMUSE Demo



Large scale experiment

- Gene sequence application, implemented using the Satin Programming model
- Satin in turn uses **IPL**
- Run on as many resources as we could find
 - Grids (DAS-3, Japan)
 - Cloud (Amazon EC2)
 - Desktop Grid (VU)
- Slow links and Firewalls all over the place
- Bad Idea, right?





401 Cores, 94.4% Efficiency

Conclusions

- Deployment is hard
- Ibis-Deploy enables users to deploy applications easily, and focus on their research.

