Interpreted Applications within BOINC Infrastructure

Daniel Lombraña González, Francisco Fernández de Vega, L. Trujillo, G. Olague, M. Cárdenas, L. Araujo, P. Castillo, K. Sharman and A. Silva

12 – 15 of May, 2008

Computing Resources in Institutions

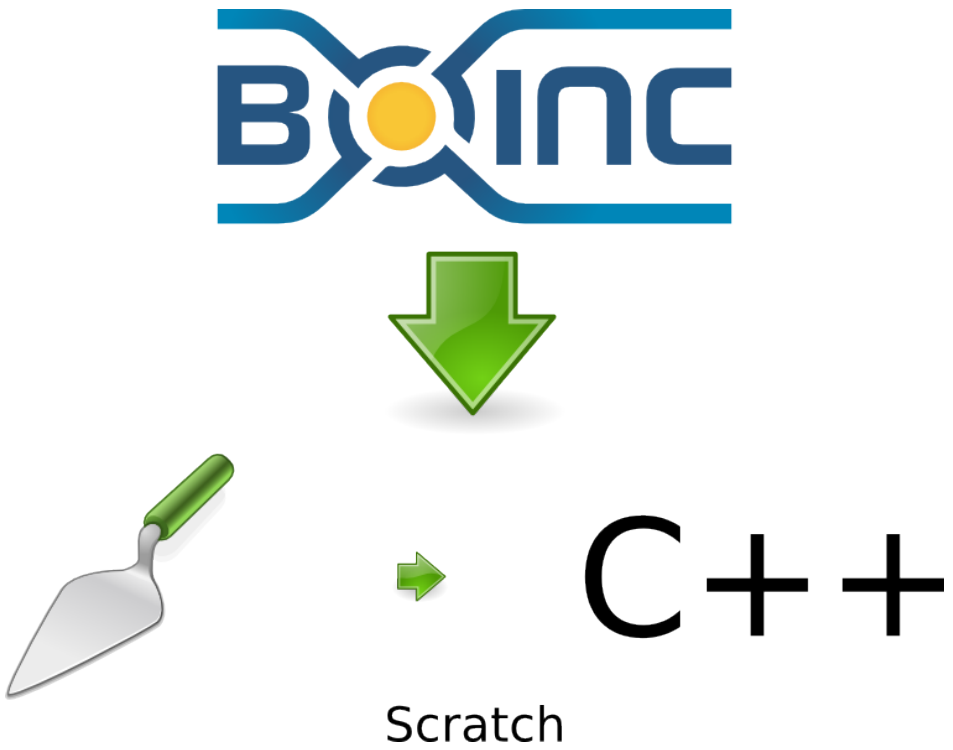
- Desktop PCs are widely used nowadays.
- Institutions like Universities have a large number of desktop PCs.
- However, these PCs (institutional and personal) are usually underemployed.
- Nevertheless, the computing power of these computers is really good (multi-cores, 1GB of RAM, etc.).

Exploiting PC Resources

- These resources can be harnessed by means of BOINC.
- BOINC is a middleware widely used by researchers:
  - 1,154,833 users
  - 2,364,170 PCs
  - 703.040 TeraFLOPS

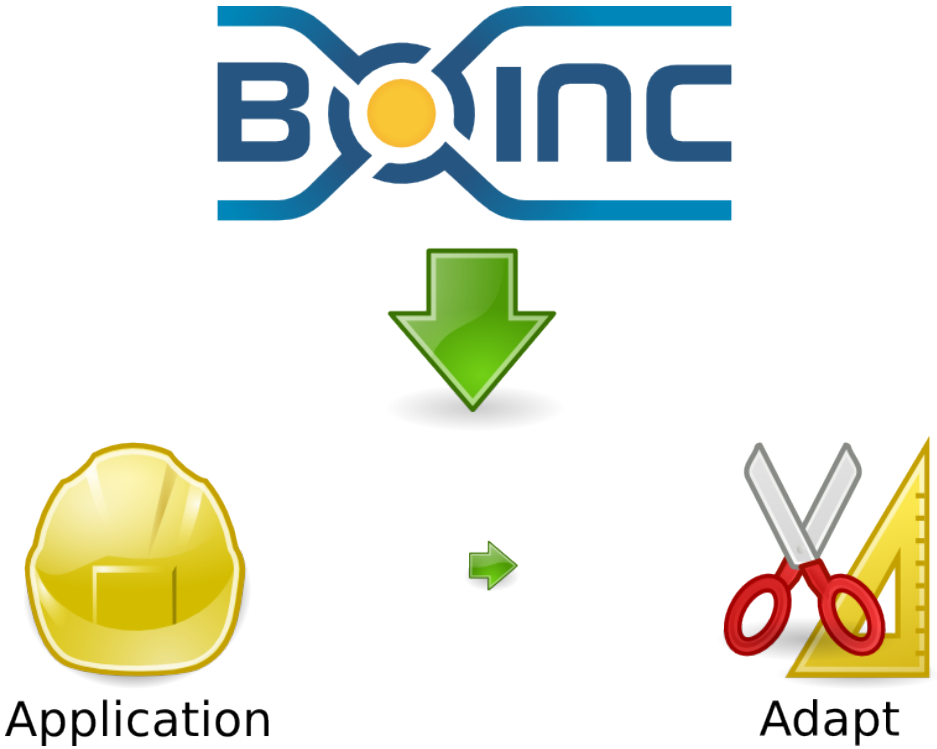
How to Support BOINC

Figure: A BOINC Project from Scratch



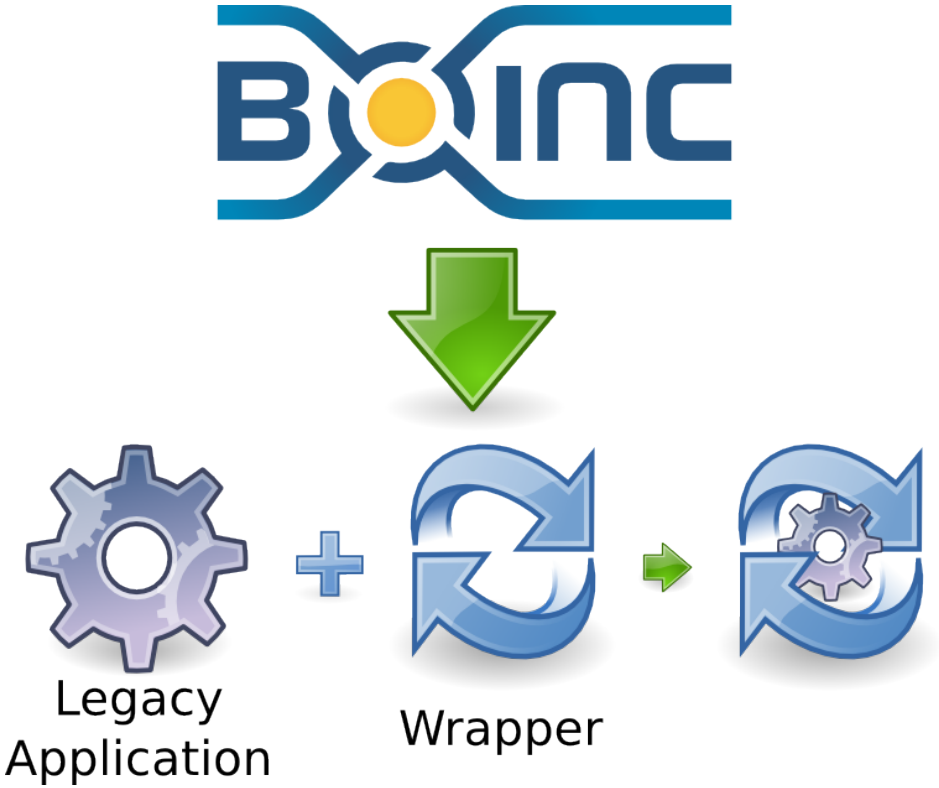
How to Support BOINC

Figure: Adapting a Scientific Application for BOINC



How to Support BOINC

Figure: Using the Wrapper



Using the Wrapper

Drawback

The *Wrapper* lacks from a checkpointing facility.

Solution

The *Legacy Application* should provide it.

Interpreted Applications (IAPs)

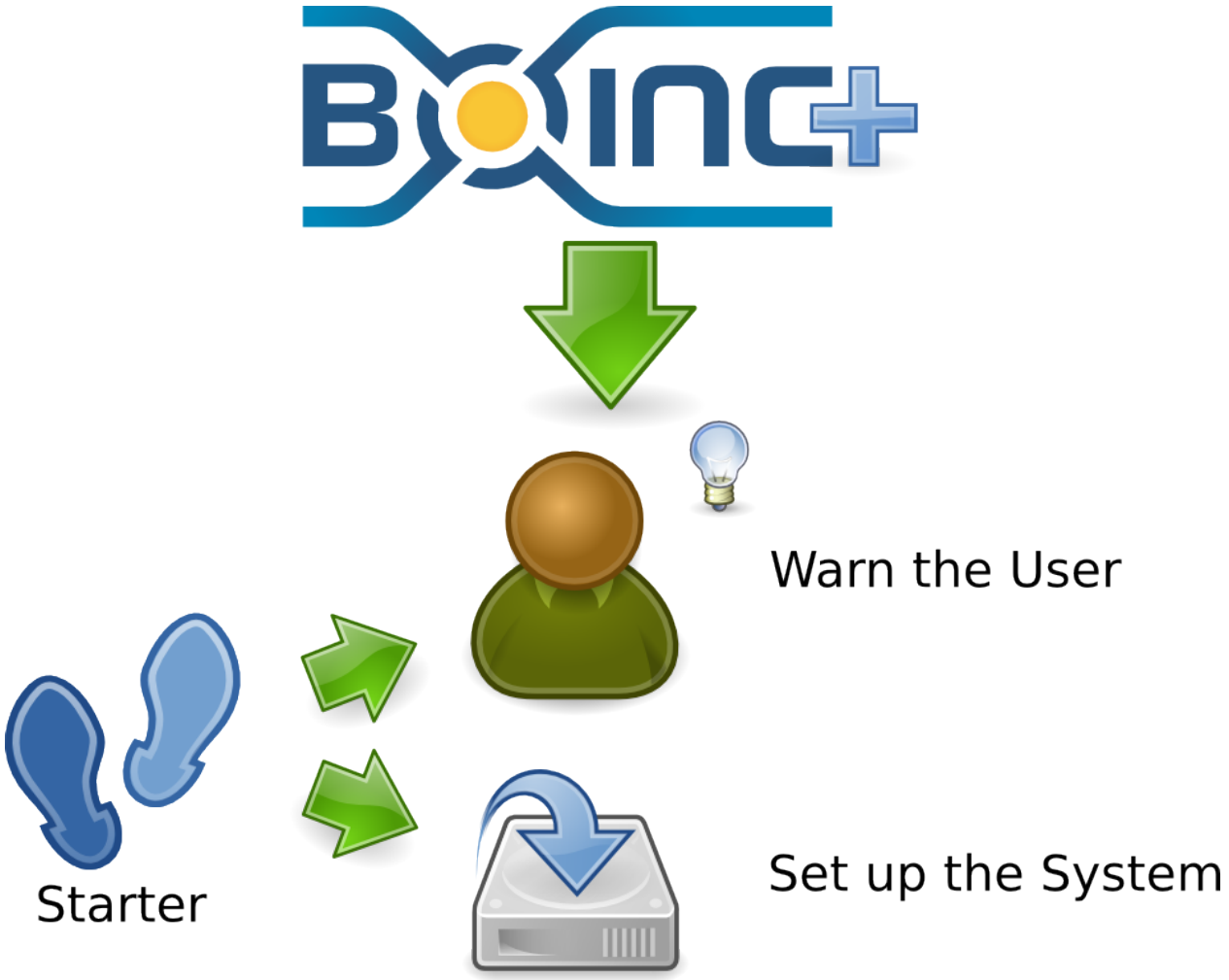
- There are applications widely used by researchers which are not statically linked. (Matlab, R, Java, etc.)
- Moreover, due to lack of a checkpointing facility some of them could never employ BOINC.

Therefore, it is not feasible to use IAP directly within BOINC.

Our Proposal

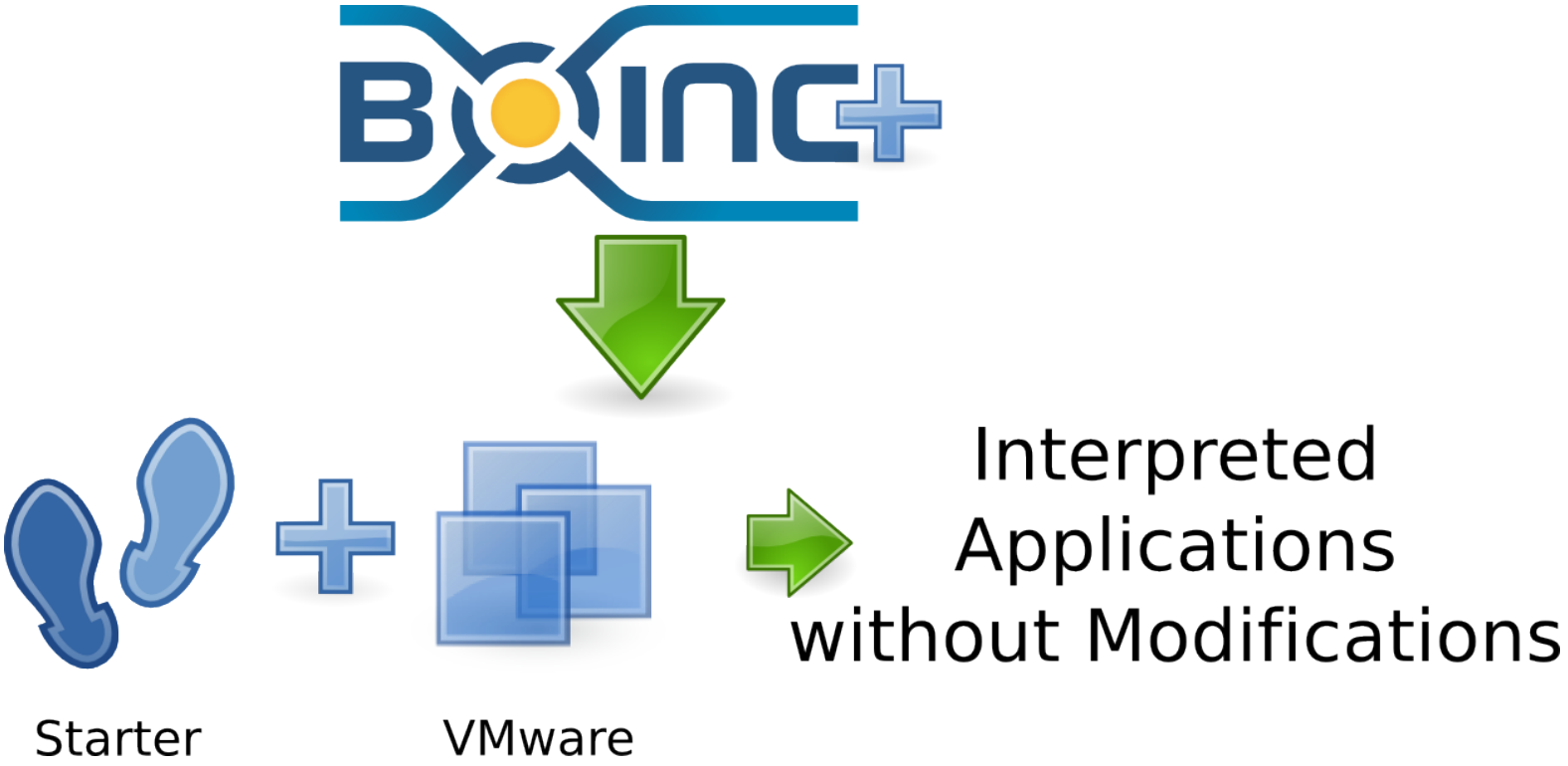
- Our proposal is to extend the BOINC framework by:
  1. Complementing it with a new program called the *Starter*, which lets to:
     - Aware the user about lacking the required software infrastructure, and/or
     - Set up the environment to run IAP jobs.
  2. Adding a new *virtualization* layer which simplifies the deployment of IAPs within BOINC.

BOINC+: the Starter



The Virtualization Technology

BOINC+: the Starter & Virtualization



Experiments

- Three experiments were set up:
  - ECJ. A Java based problem.
  - R. An R based problem.
  - Matlab. A Complex Scientific Application.
- All the experiments employ the *Wrapper + Starter* proposed solution.
- The goal is to check if it is possible to run them within BOINC.

Performance Improvement

- The *Performance Improvement* is measured by the following equations:

Speed Up

$$A = \frac{T_{seq}}{T_B}$$

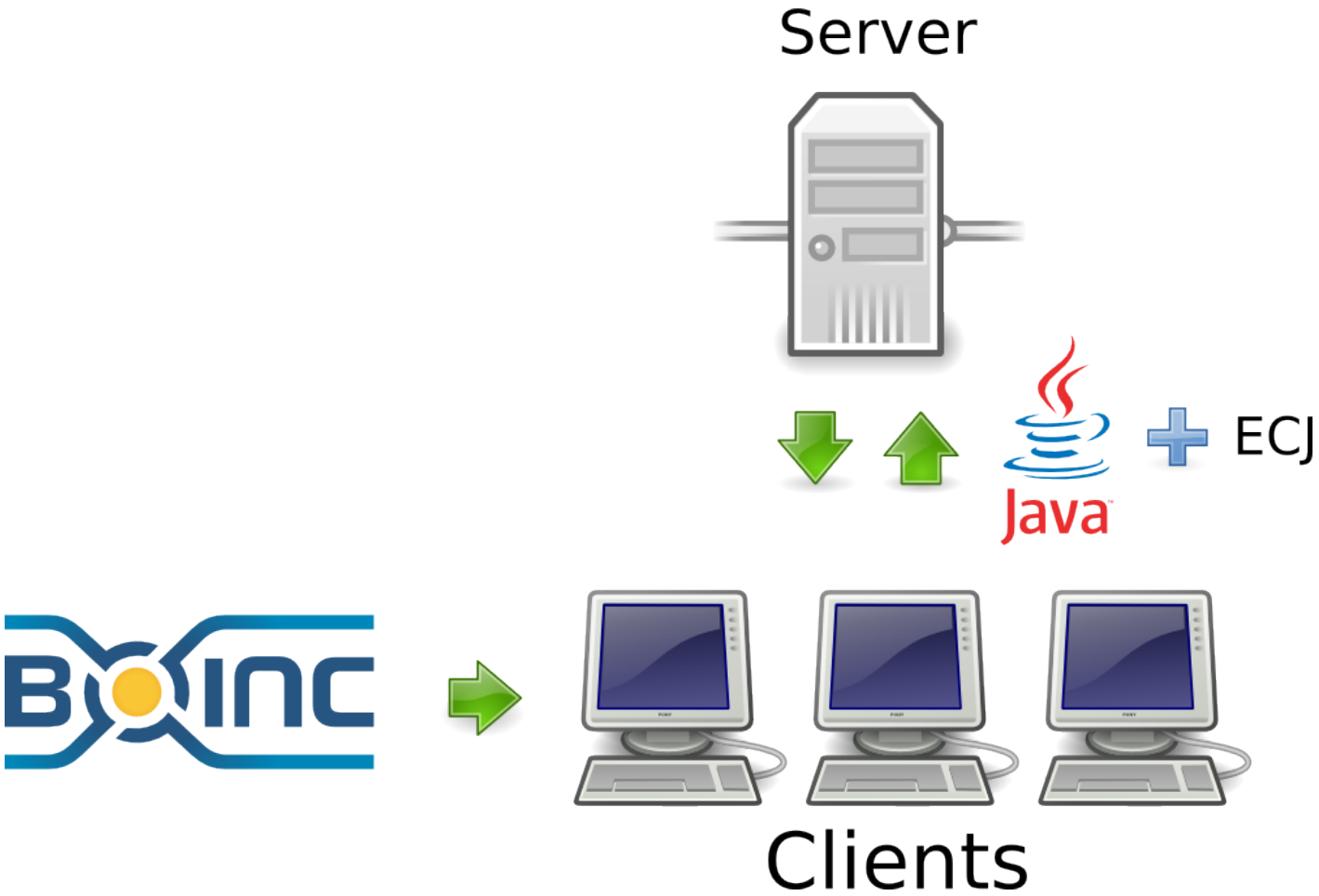
Computing Power

$$CP = X_{arr} * X_{life} * X_{ncpus} * X_{flops} * X_{eff} * X_{onfrac} * X_{active} * X_{red} * X_{sh}$$

ECJ a Evolutionary Computation Research System

- ECJ is a Java based evolutionary computation research system.
- We employ a standard benchmark problem for Genetic Programming: The Multiplexer of 20 bits.
- The goal is to check if the BOINC clients are able to run ECJ and return some results.

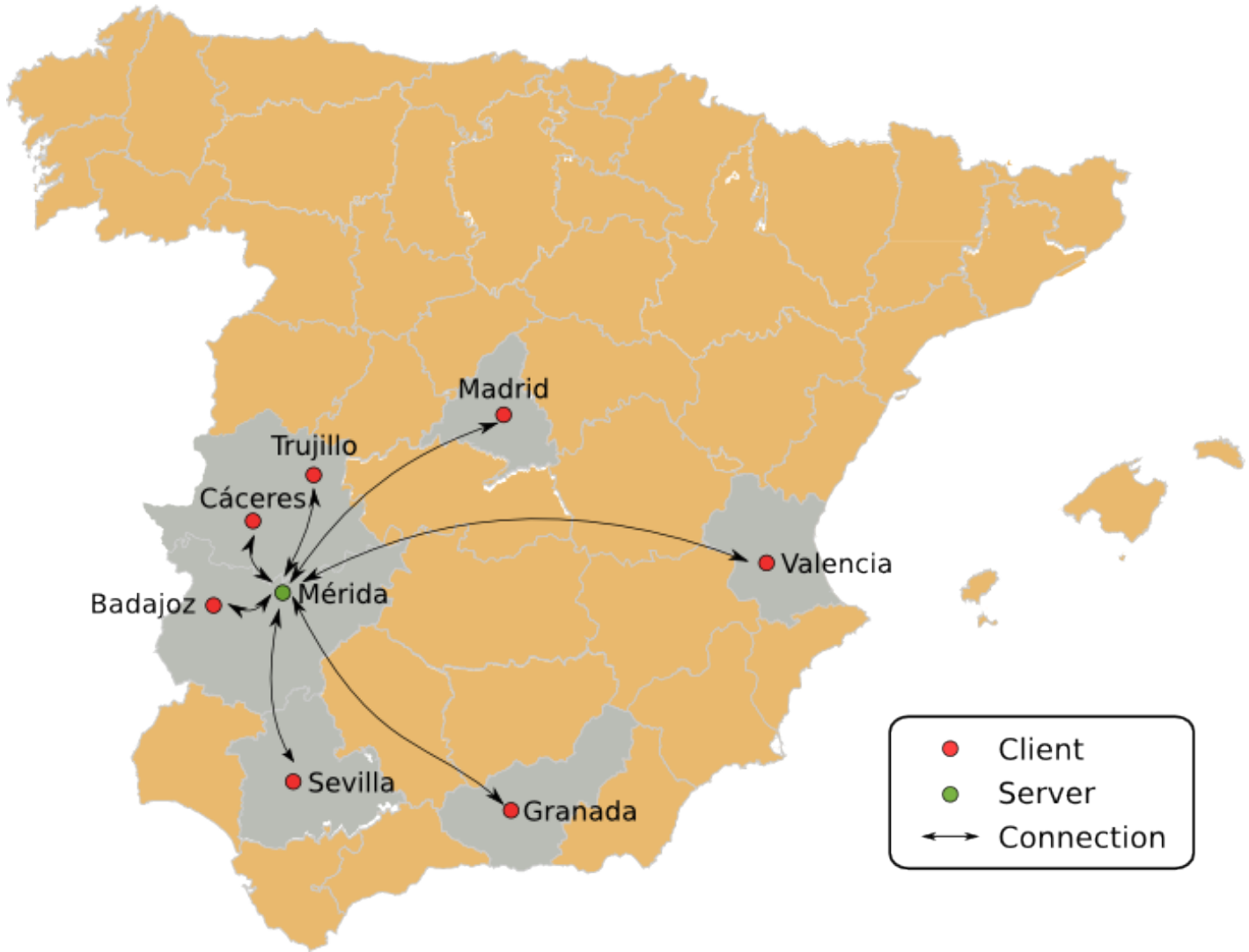
Java, a Statically Linked Version



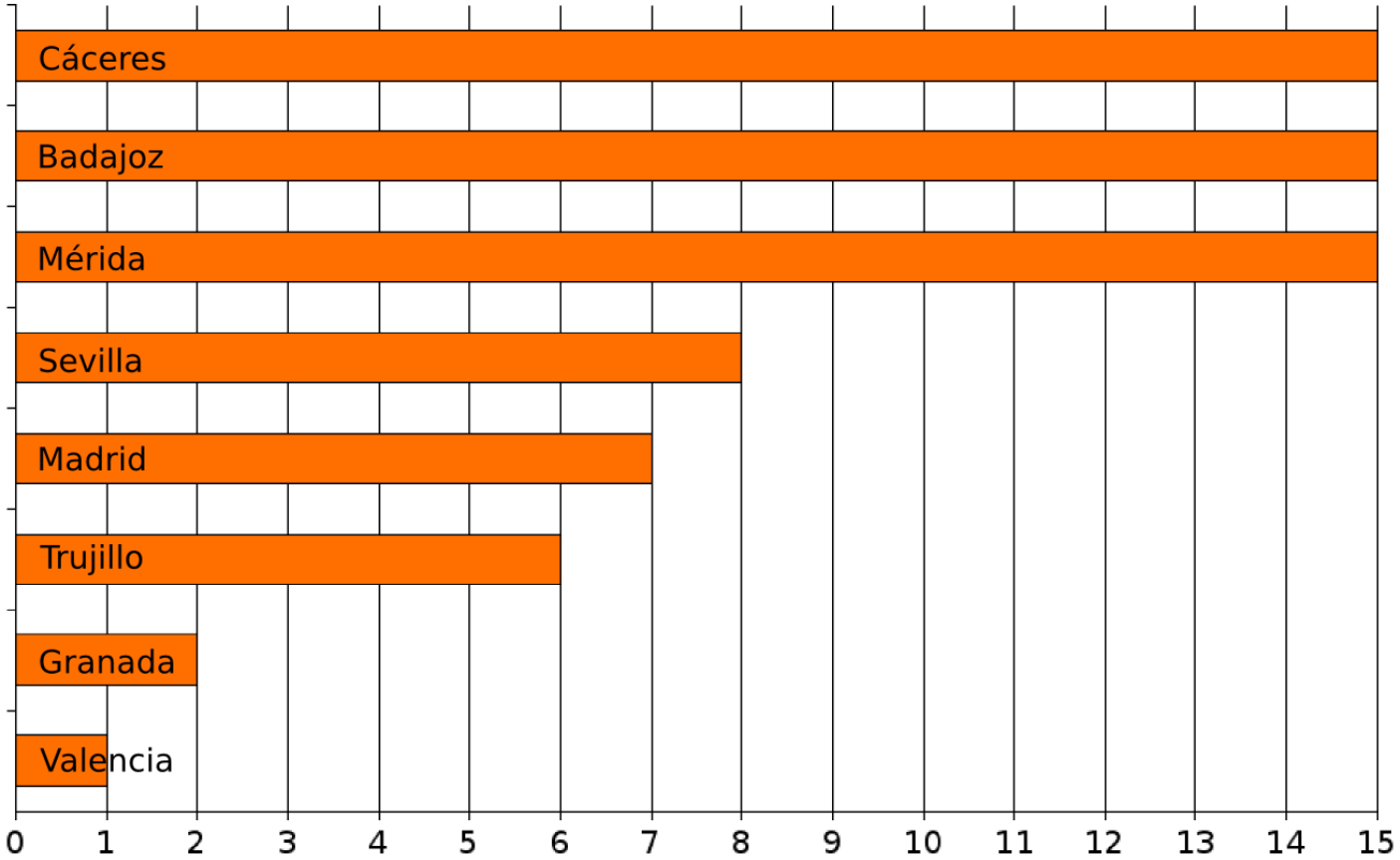
Checkpointing

- In this case, the ECJ tool provides the checkpointing feature.
- The *Starter* is in charge of handling the checkpointing issues.

ECJ BOINC Infrastructure



Clients per City



Obtained Results

Table: Execution time for ECJ and ECJ-BOINC

	T_{seq}	T_B	Acc.	CP
42 R., 50 G., 1000 I.	1305330s	669759s	1.95	23 GF

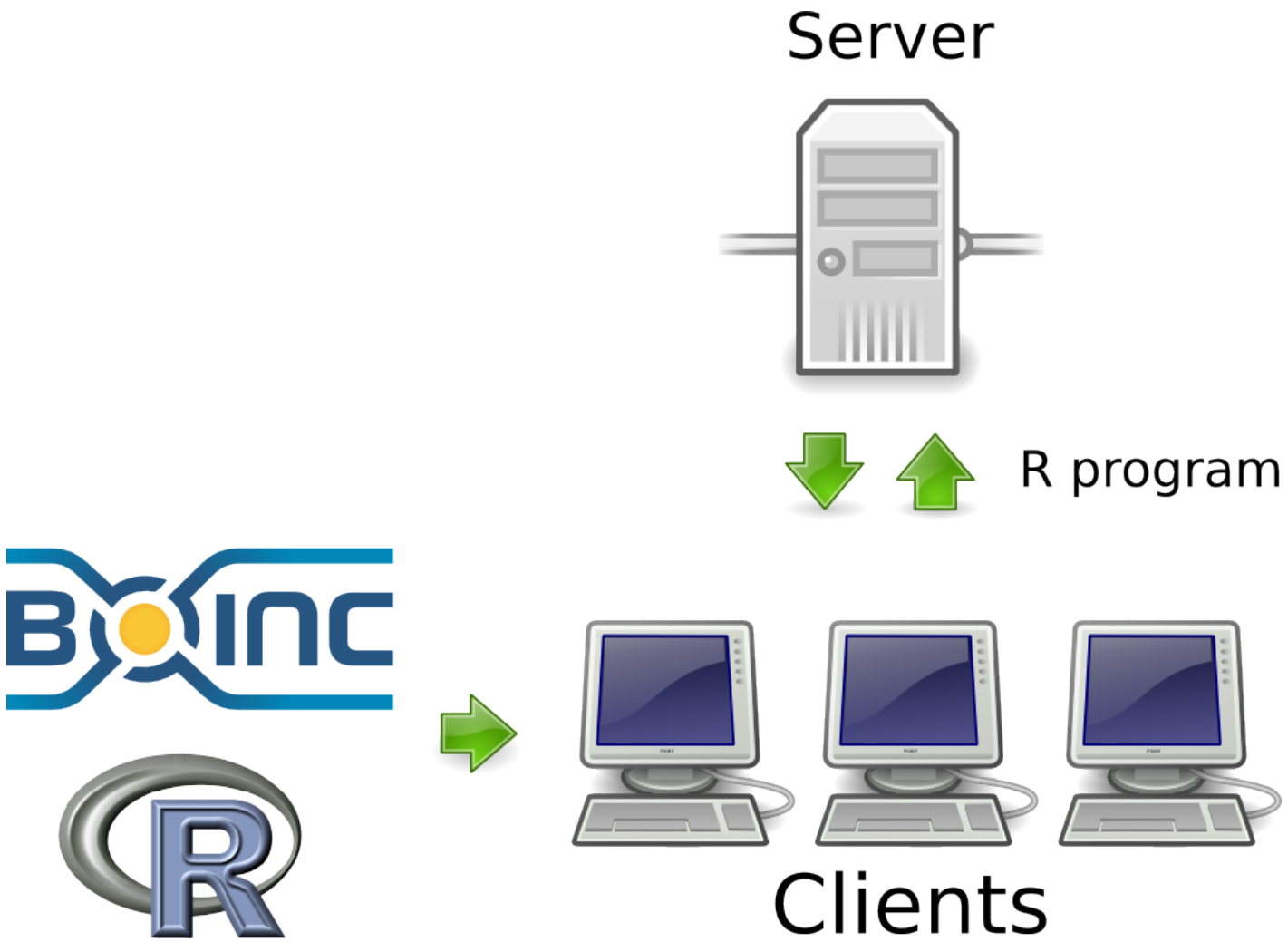
R a Statistical Tool

- R is an open source statistical software.
- R does not provide a statically linked version.
- Therefore, all the clients have to install R before BOINC.

The R problem

- A laboratory with 20 computers were set up for the experiment.
- The R problem is a proof of concept: some float and integer operations which produces a result.

R BOINC Infrastructure



Checkpointing

- R lacks from a checkpointing facility.
- For this reason, we propose to extend BOINC with a *virtual* layer.

The Virtualization Features

- Resource Isolation.
- Guest OS Instantiation.
- Snapshots.

The Virtualization Features

Thanks to the virtualization features:

Snapshot Feature

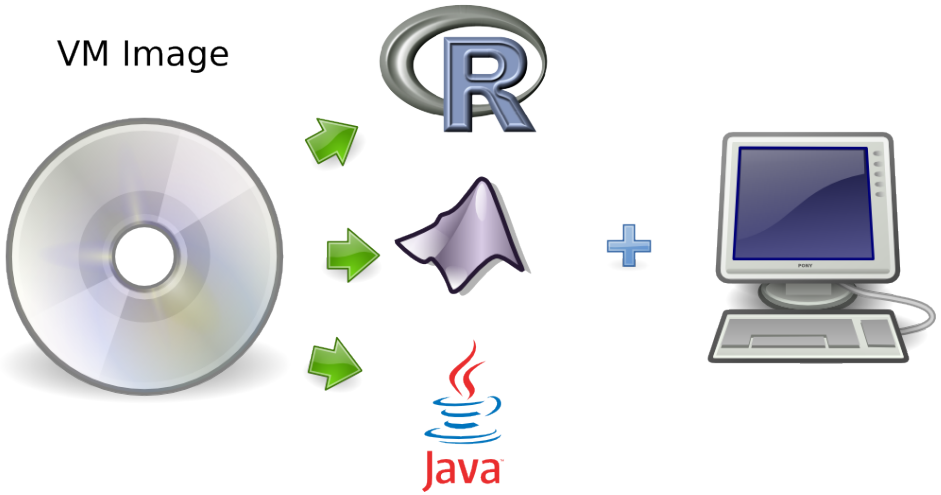
BOINC can have a checkpointing facility for any IAP.

Any IAP can be run within BOINC without modifying the source code.

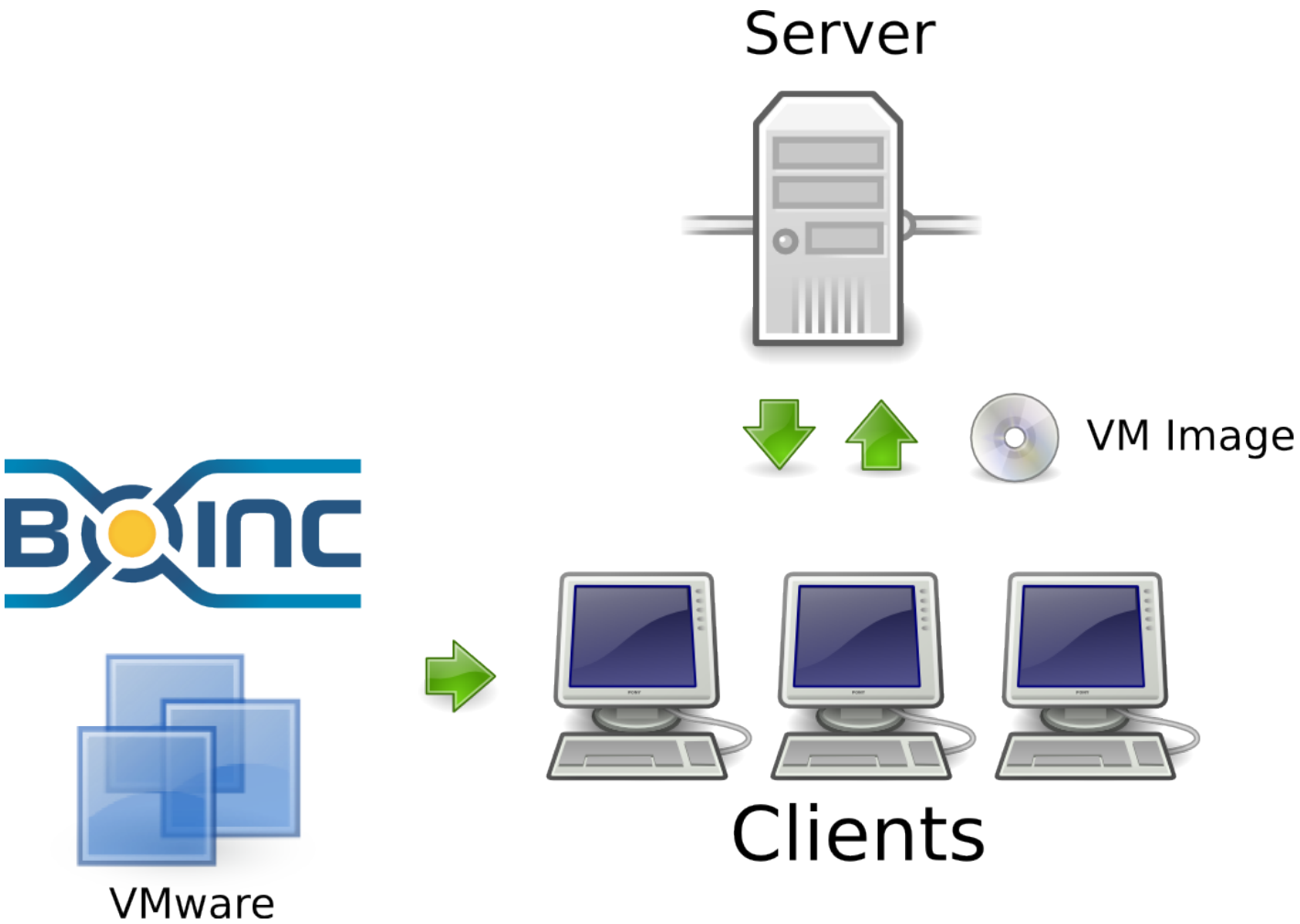
A Computer Vision Problem

- A Complex environment is employed to test the new BOINC + Virtualization infrastructure.
- The problem employs Matlab and several toolboxes.
- Additionally, the Computer Vision Problem lacks from a checkpointing facility.

Virtual Machine Image



VMware and BOINC Deployment



Obtained Results

Table: Execution time for Sequential and VM-BOINC Matlab

	T_{seq}	T_B	Acc.	CP
75 Gen, 75 Ind.	215h	48h	4.48	25.67 GFLOPS

Conclusions

- We have presented a new approach to run IAPs.
- A new program called *Starter* has been developed to compliment the *Wrapper* solution.
- We have extended BOINC via a virtualization layer which lets to:
  - Run any IAP directly within BOINC without modifying any single source code line.
  - Have a checkpointing facility for any IAP.

Acknowledgments

This work was supported by:

- Cátedra CETA-CIEMAT Universidad de Extremadura,
- Regional Gridex project PRI06A223 Junta de Extremadura, and
- National Nohnes project TIN2007-68083-C02-01 Spanish Ministry of Science and Education.

Questions

Contact

Daniel Lombraña González **daniellg@unex.es**
Francisco Fernández de Vega **fcofdez@unex.es**