

Performance Investigation and Tuning in the Interoperable Cloud4E Platform

Steffen Limmer¹, Maik Srba², Dietmar Fey¹
FedICI, Porto, August 26, 2014

¹Friedrich-Alexander University Erlangen-Nürnberg (FAU), ²GWDG Göttingen

Outline

Introduction

Overview Cloud4E Platform

Performance Investigations and Tuning

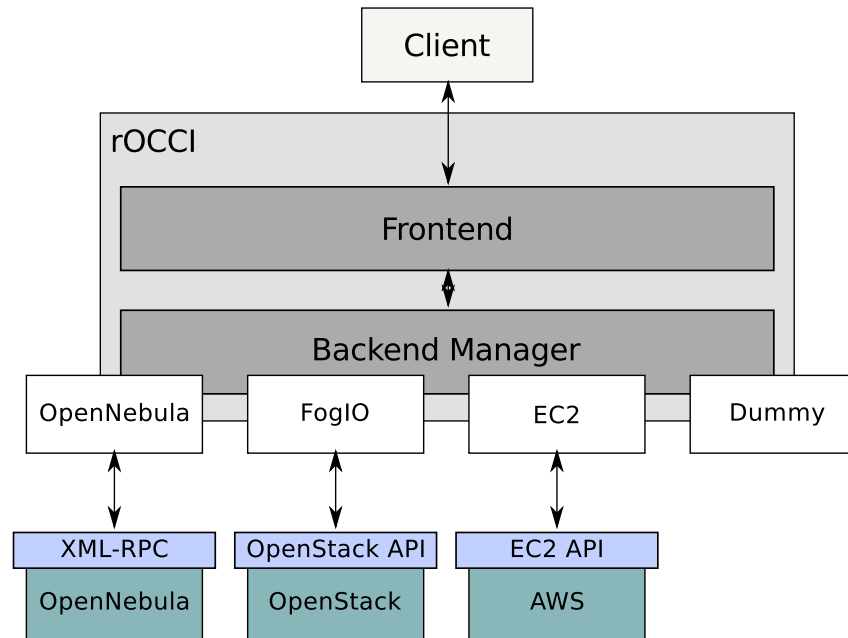
Summary



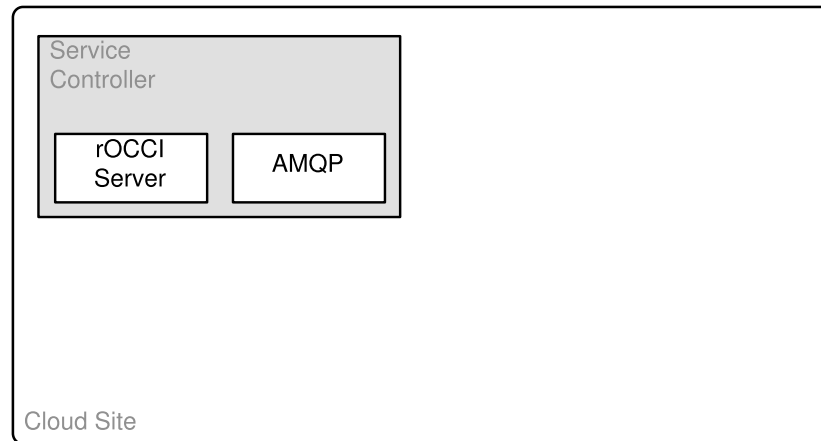
- Goal: Platform that allows provisioning of existing simulation software as cloud service
 - Regional compute centers as infrastructure and platform providers
- Portability important
- Limited resources of regional compute centers
- Enable federated clouds

- Usage of OCCI (Open Cloud Computing Interface)
 - Open cloud interface by Open Grid Forum
 - Intended for control of IaaS
 - Extended in Cloud4E to interface for SaaS
- AMQP (Advanced Message Queuing Protocol) as transport protocol
 - Good scalability
 - More features than HTTP
 - Central AMQP server

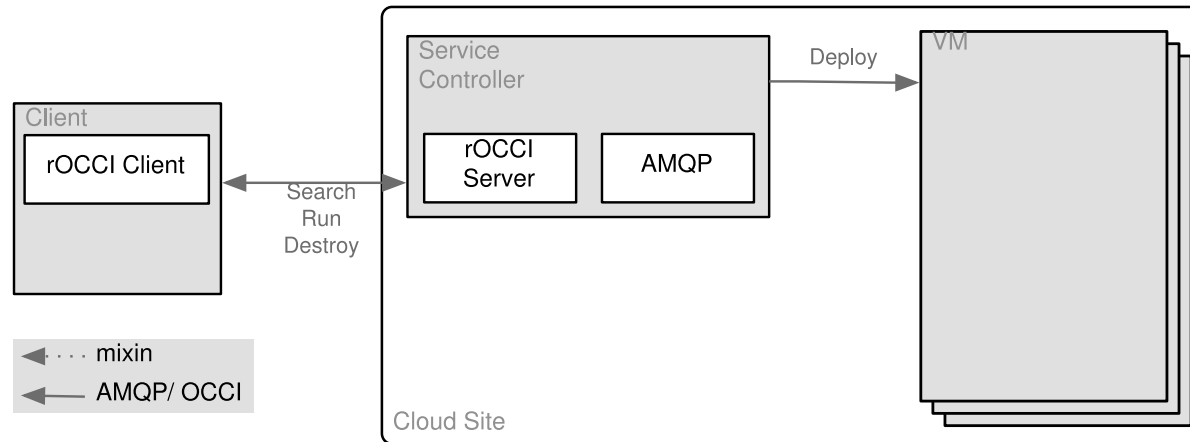
- Use rOCCI server as OCCI server



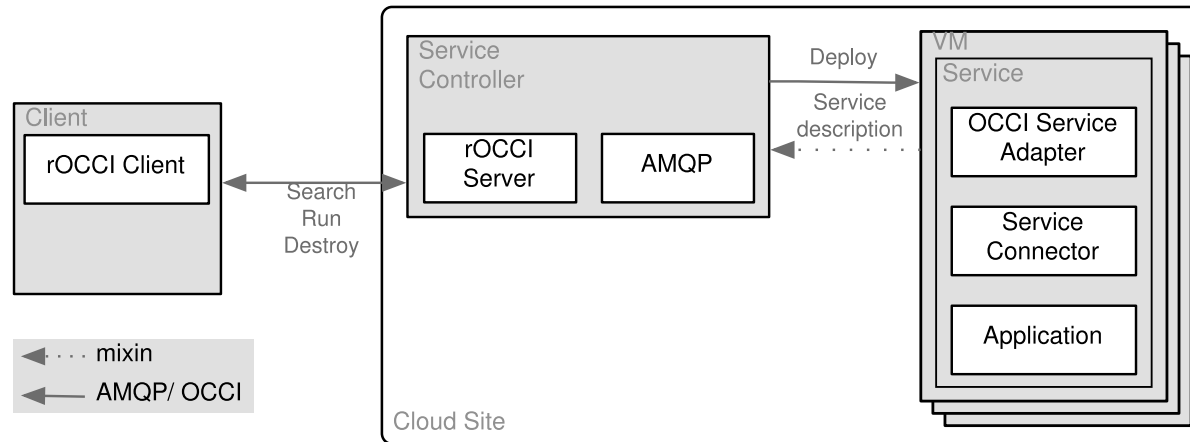
- Crucial impact on the performance
- Investigated in detail together with OpenNebula cloud middleware



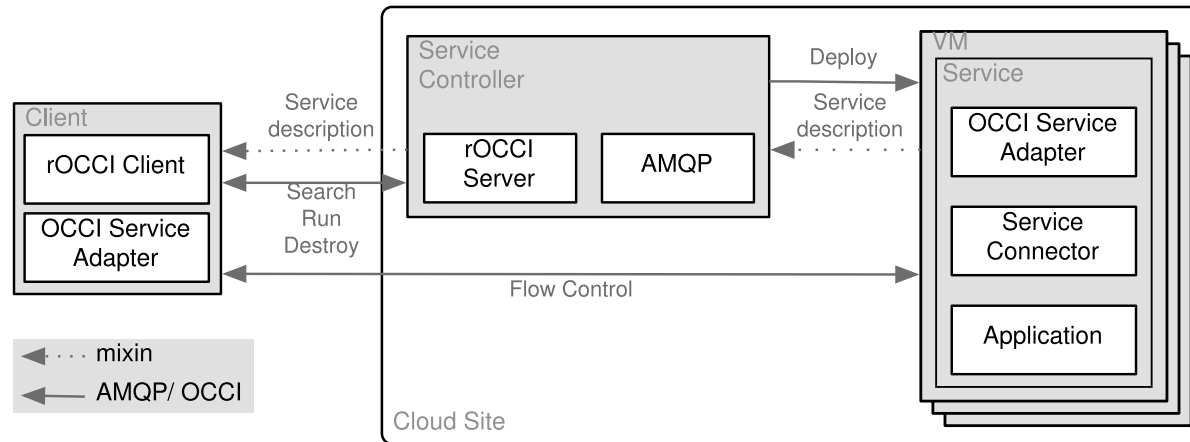
- Central part: Service Controller



- Start of virtual machines (VMs) (“compute resources”) over rOCCI
- Control of VMs over actions and attributes

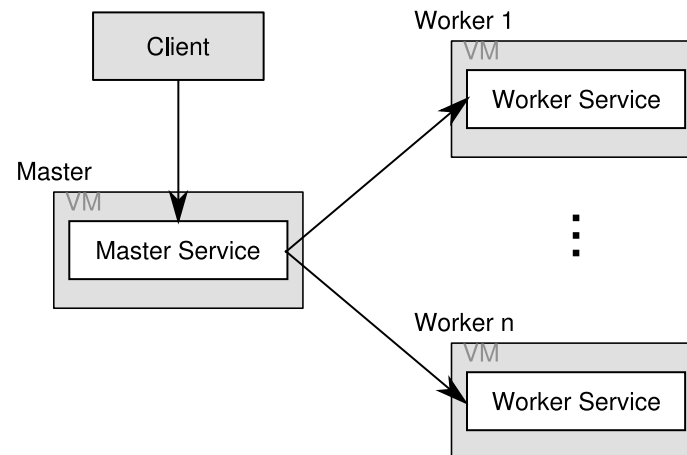


- Service = *OCCI Service Adapter* + *Service Controller* + simulation tool
- Control of service over actions and attributes
- Actions and attributes defined and implemented in *Service Controller*
- *OCCI Service Adapter* as generic interface



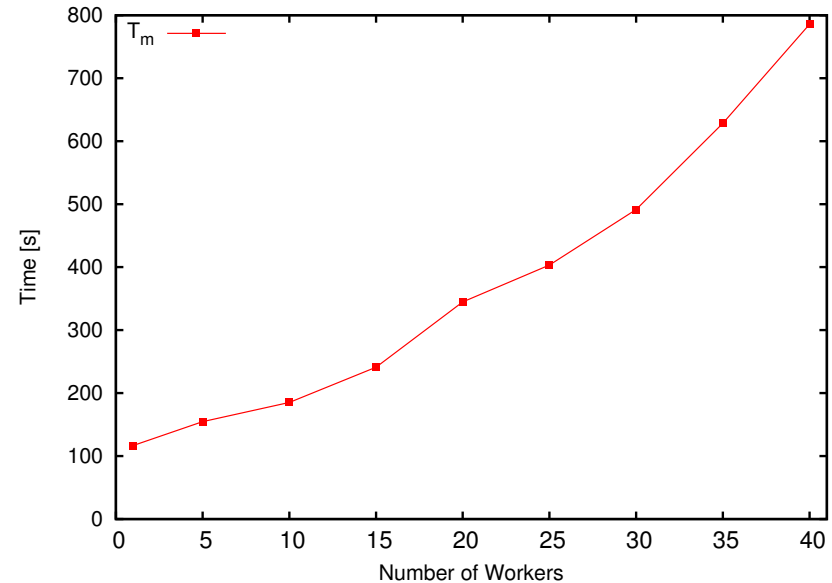
- OCCI Service Adapter used for client side control

- Initially, performance problems related to rOCCI server and cloud middleware
- Particularly for start of worker services over master service

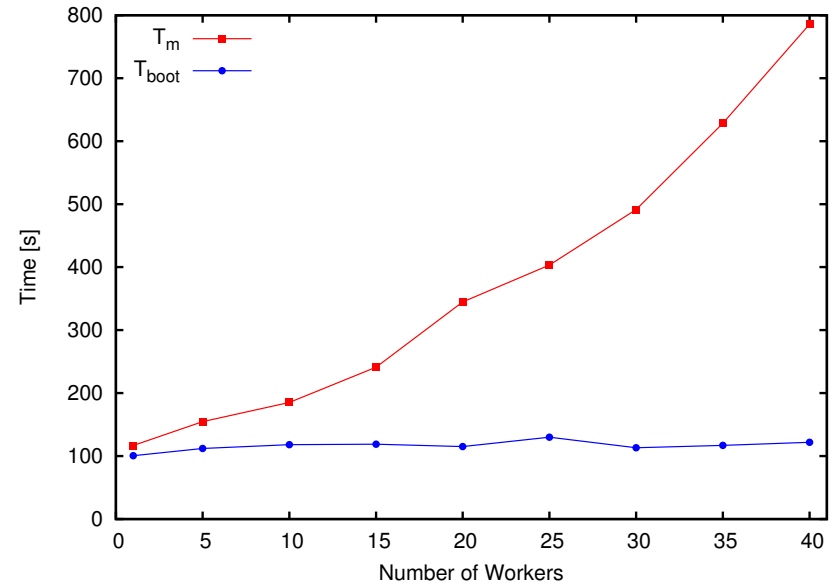


- Start of workers over `start_workers` action of master
- Master polls state of workers from rOCCI server until all worker services started

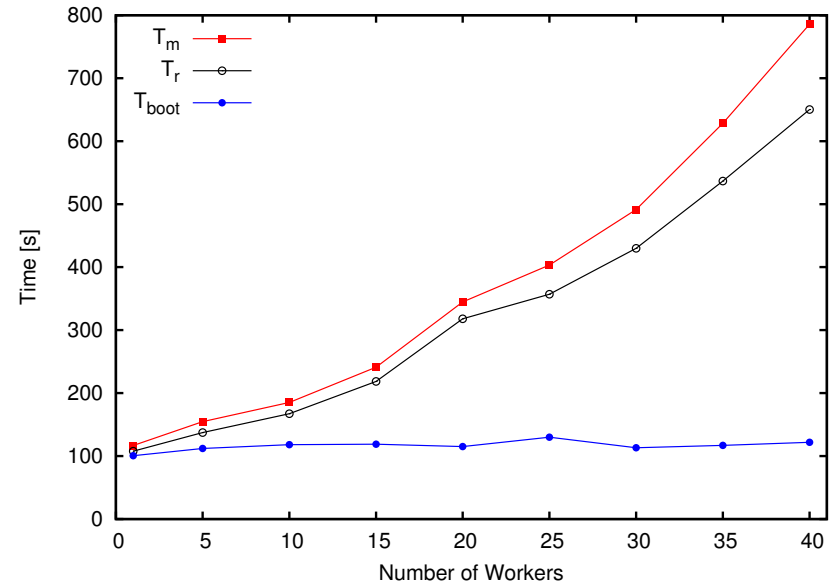
- T_m : timespan between triggering the `start_workers` action and the moment when the master recognizes that all workers are running (averaged over 5 measurements)
- OpenNebula 4.0.1, Ruby 2.0.0, Windows worker
- Master polls worker states in intervals of 15 s



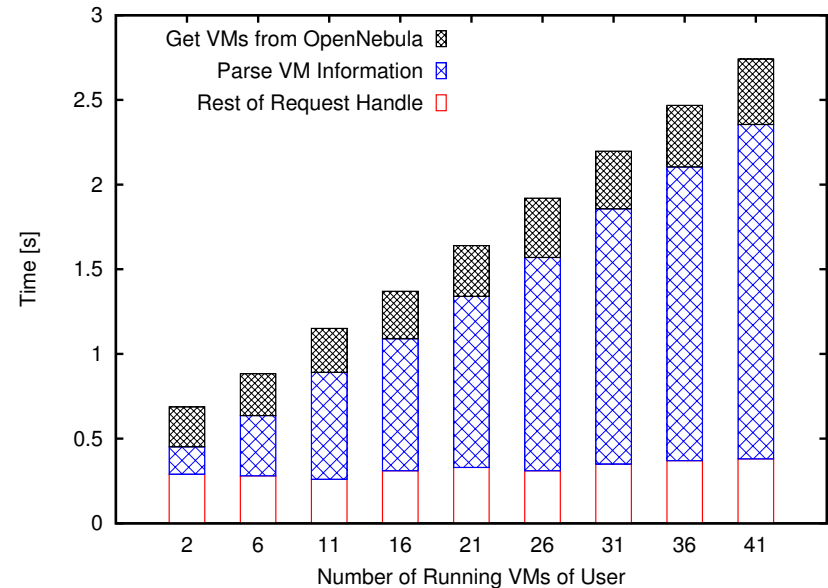
- T_{boot} : time to start and boot the worker VMs over OpenNebula (averaged over 10 measurements)



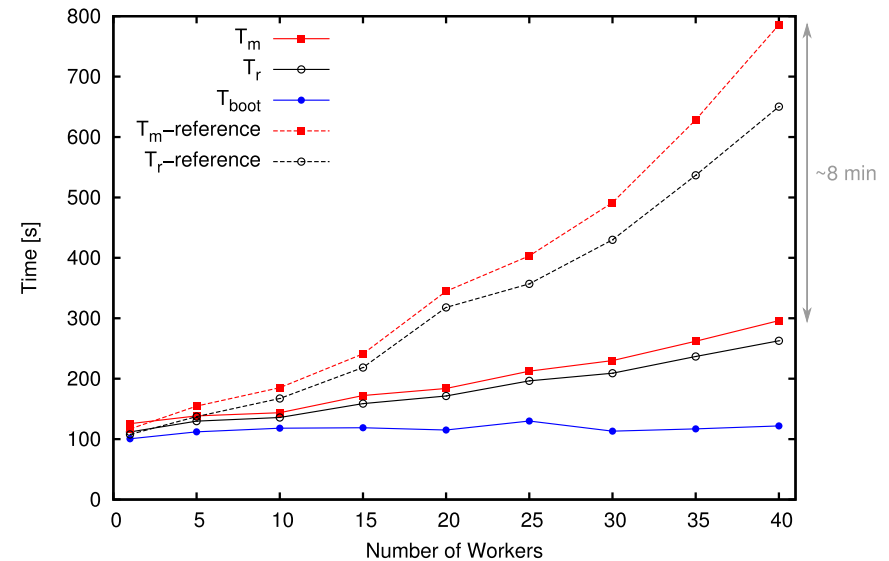
- Delay between moment when all worker services are started and the moment when this is recognized by master
 - T_r : start time worker measured over rOCCI server (averaged over 5 measurements)
- “real” start times



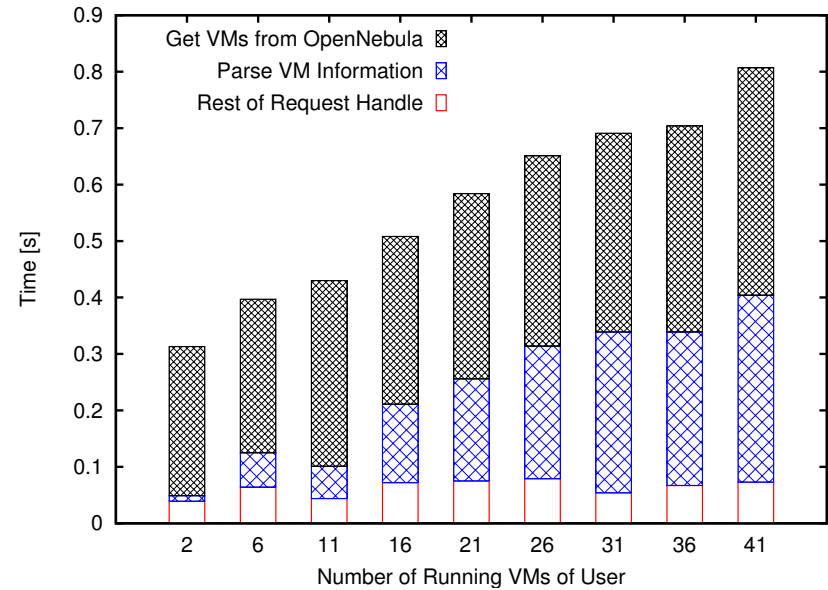
- Slow start of worker services after start of worker VMs
 - 5 requests to OCCl server per worker service
 - 2.7 s to handle one request when 41 VMs running
- Reason for performance issues



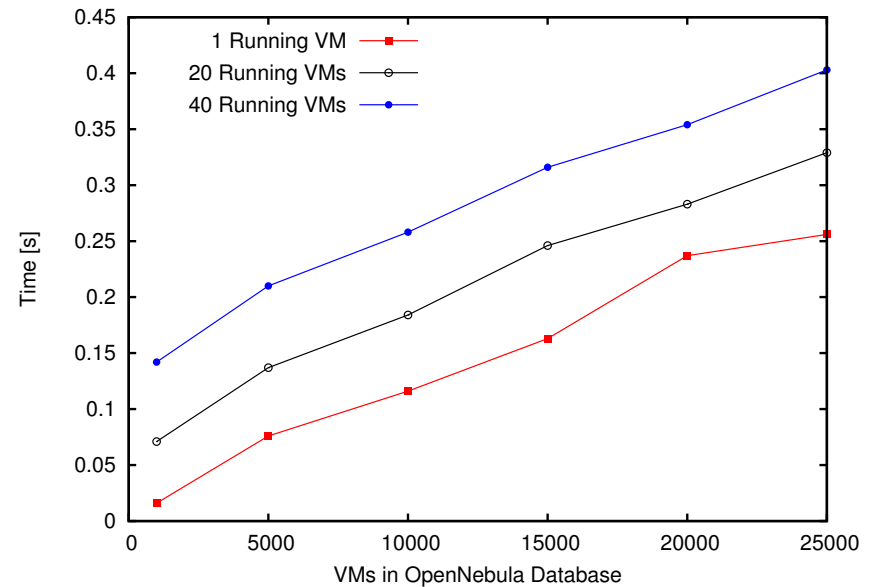
- Removed redundant and unnecessary tasks from rOCCI
- for 40 workers improvement of about 8 *min*
- Smaller gap between T_m and T_r



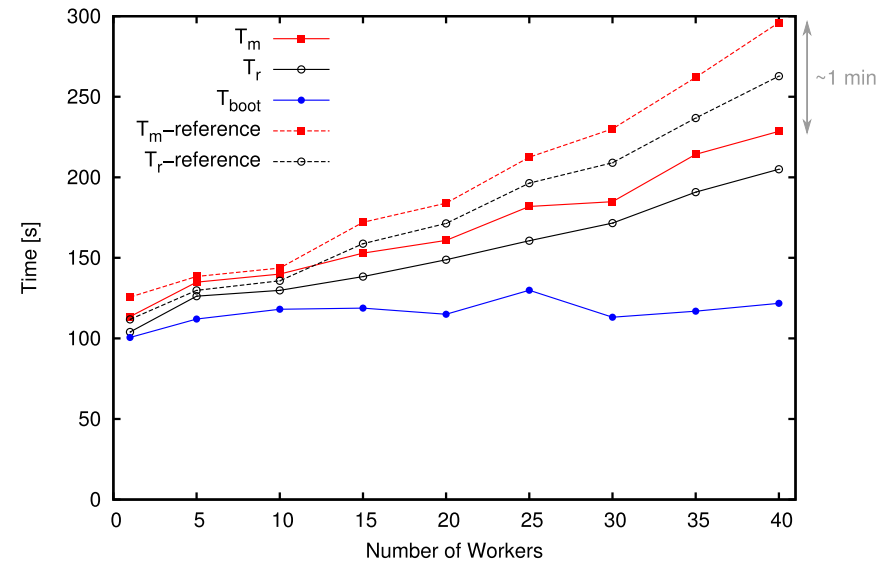
- After improvements the query of VMs accounts for a bigger portion of the times to handle a message



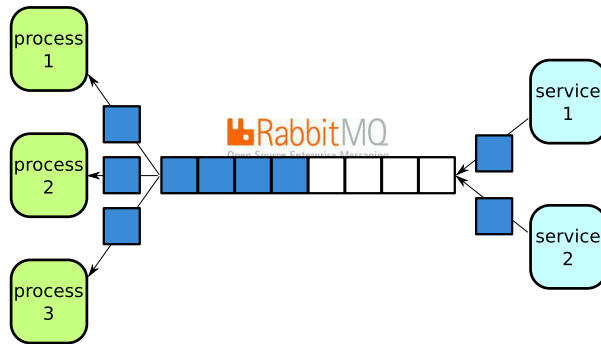
- Time querying running VMs depends on number of VMs in ON database
- For 40 VMs 3 times faster with 1,000 VMs in DB than with 25,000 VMs in DB
- For 20 VMs 16 times faster with 1,000 VMs in DB than with 25,000 VMs in DB



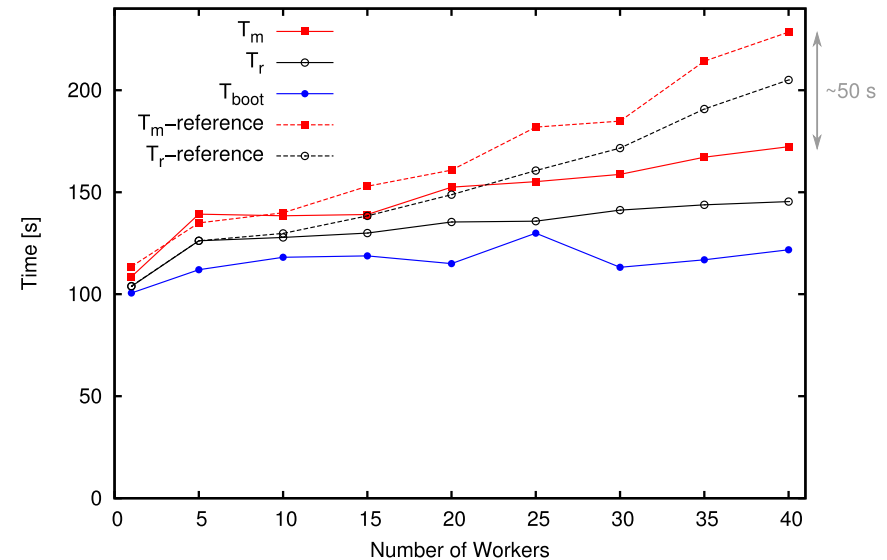
- Times after deleting old VMs from ON database
- For 40 Workers more than 1 *min* faster
- Still about 100 s more than T_{boot}



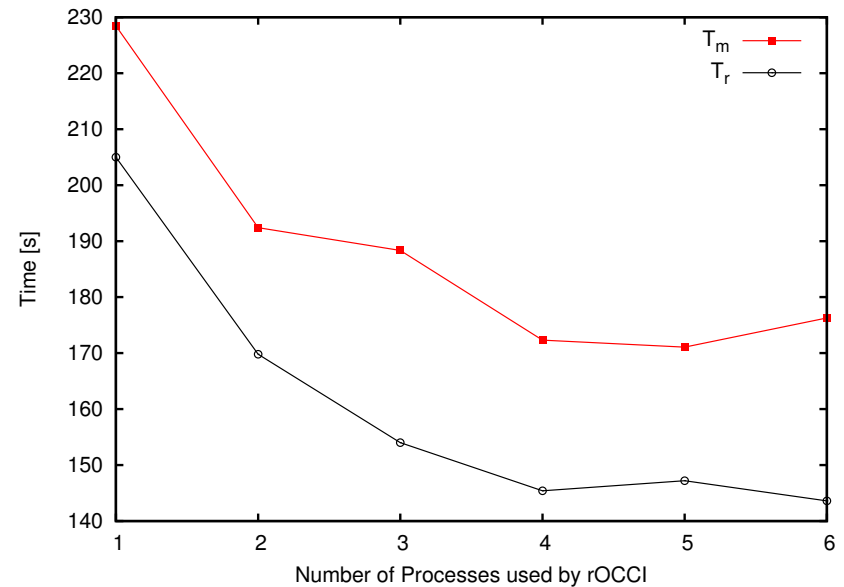
- Further improvement by parallelization of rOCCI server
- Multiple processes connected to same AMQP queue



- Times measured with 4 processes

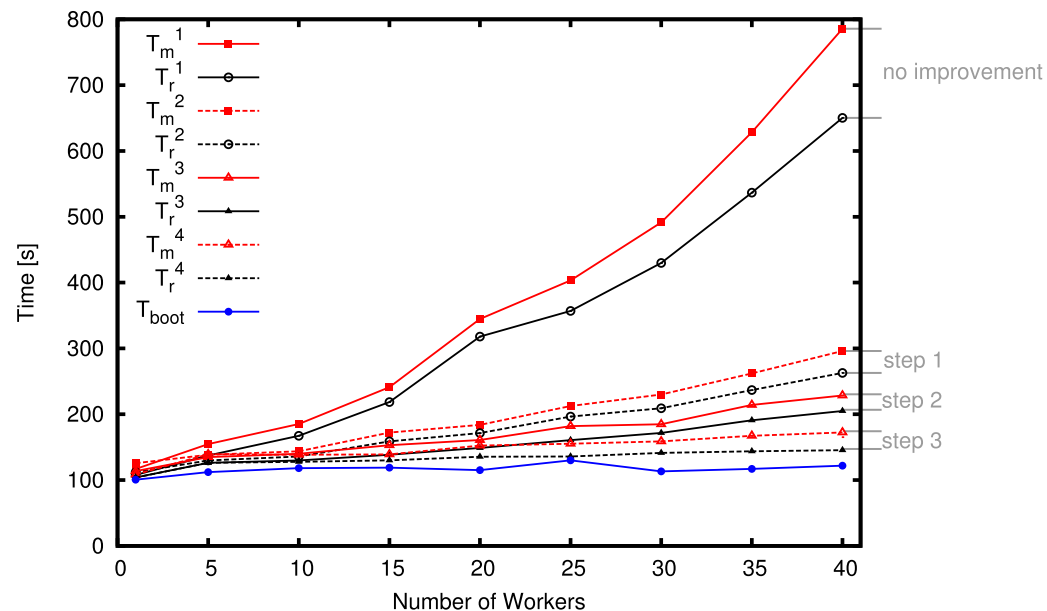


- No further speedup with more than 4 processes (start of 40 workers)



- Interoperability ensured and vendor lock-ins avoided by use of open standards OCCI and AMQP
- rOCCI server has a significant impact on the performance
- Possibilities for improving performance of rOCCI server
 - Remove unneeded features
 - Adaptions to the cloud middleware
 - Parallelization (easy with AMQP as transport protocol)

- Acceptable times after improvements
- Better scalability



Thank You!



Supported by:



Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag

www.cloud4e.de

- Simple Service Connector with an attribute state und an action
start_action:

```
module OcciServiceAdapter
  module Connector
    class Myconnector < OcciServiceAdapter::Connector::Service2

      attribute :state, {:type => 'String', :required => false, :default => 'ready'}

      action :start_action, {:parameter => {:type => 'String', :require => true}}

      def start_action(params={})
        @state='started'
        system("./my_application "+params[:parameter])
        return 0
      end
    end
  end
end
```


- Start service in directory of Service Connector:
occi-service-adapter connector start
- Remote control over generic graphical client:

