Schedule Distributed Virtual Machines in a Service Oriented Environment

Lizhe Wang
Gregor von Laszewski
Suresh Marru (speaker)

Pervasive Technology Institute
Indiana University, US

Jie Tao
Marcel Kunze

Steinbuch Center for Computing
Karlsruhe Inst. of Tech., Germany

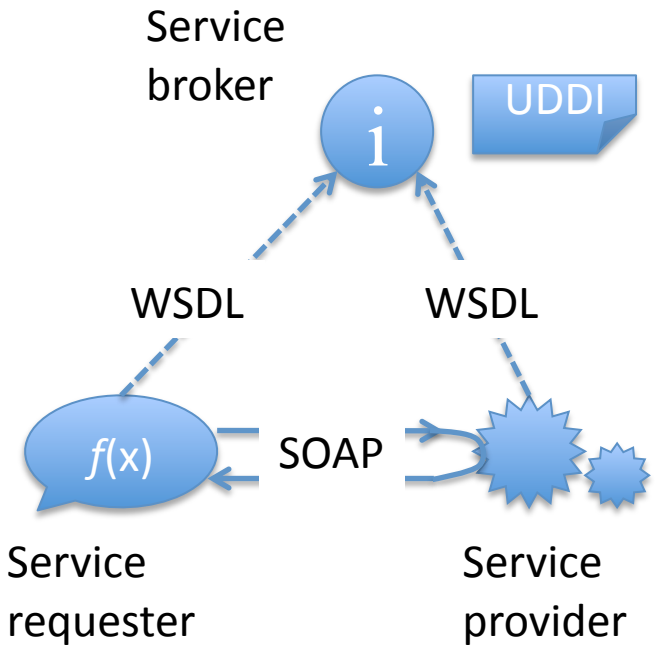
Outline

- Introduction
- Problem definition
- Schedule distributed VMs in a SOA environment
- Performance study
- Conclusion

Introduction

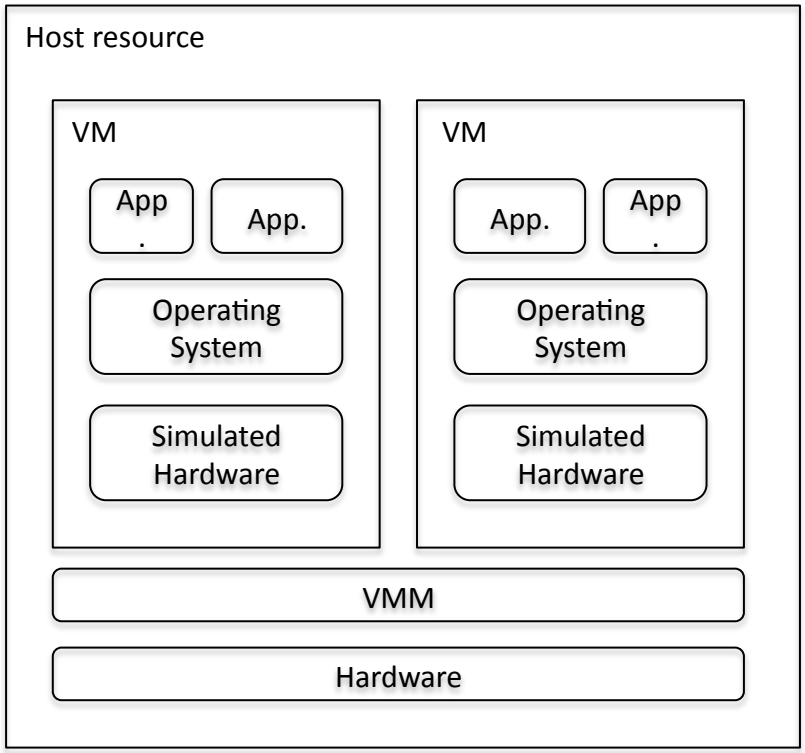
- Service oriented architecture
- Virtual machine
- Task scheduling

Service Oriented Architecture



Virtual machine

- A build block for modern IT infrastructures and Clouds
- Advantages:
  - On-demand creation and customization
  - QoS guarantee and performance isolation
  - Legacy software support
  - Easy management
- Popular VMMs
  - Xen
  - VMware



Task scheduling

- Static scheduling
  - the assignment of tasks to resources is performed before applications are executed
  - all the overhead of the scheduling process is incurred at compilation time
  - Sometime, Hard to get resource/task information before execution
  - Cannot adaptive to dynamic changed environment
- Dynamic scheduling
  - redistribution of tasks among PEs during execution time.
- VM is suitable for static scheduling
  - VM as a task, information can be achieved before exeuction
  - Dynamic migration of VM across SOA maybe expensive

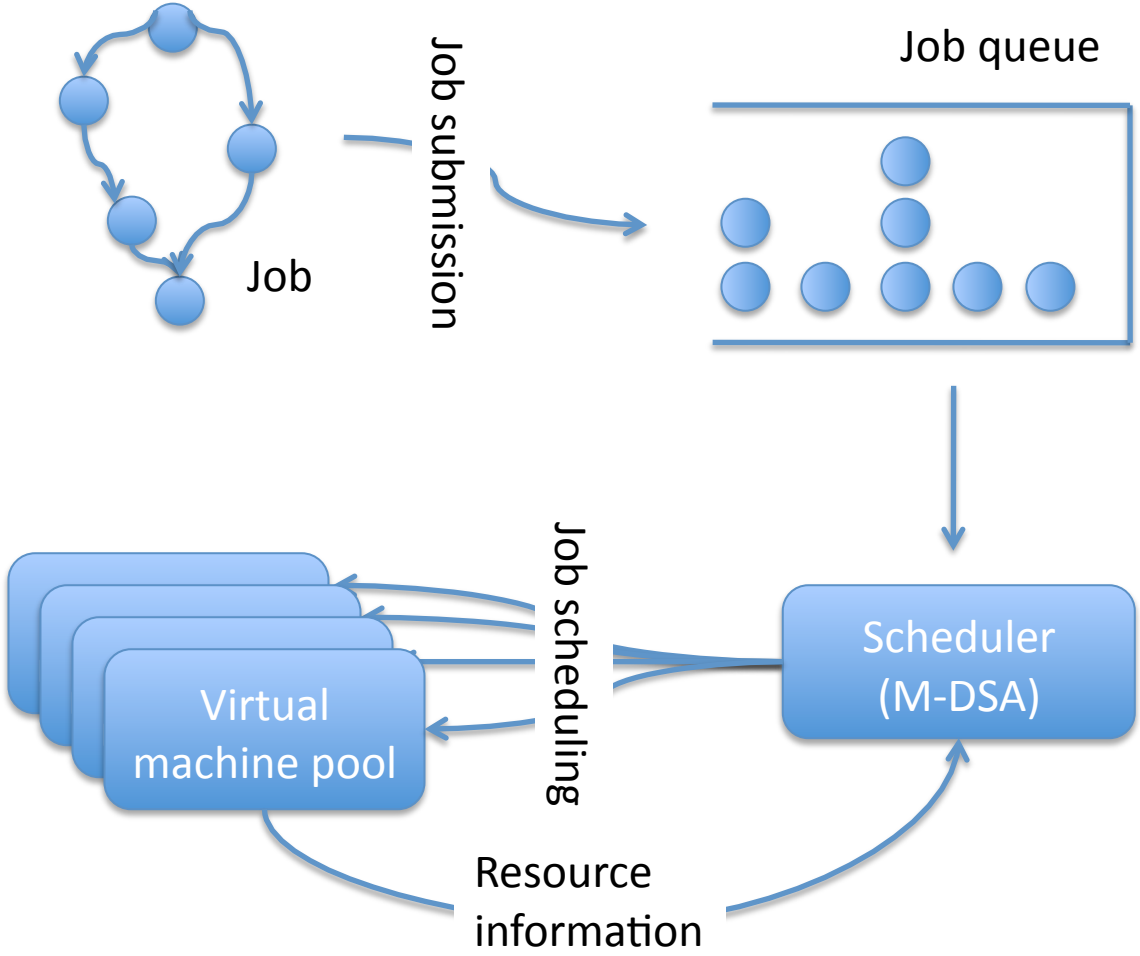
Schedule VM in a SOA

- Computing resources required to be scheduled are multiple dimension, for example, CPU bandwidth, memory, and software licenses. In traditional scheduling environment, only processor resources are considered for resource allocation.
- Resource allocation should be considered with more restrictions, for example, some applications can only be scheduled to certain virtual machines that provide the required application execution environment. This scenario does not exist in the traditional distributed system.

Formal definition

- SOA = {Host, VM, VM affiliation}
- Task= {Job, job dependency}
  - A job can have multiple resource requirements:
    - CPU number
    - Memory,
    - OS
    - Software licenses
    - ……
- Problem definition:
  - Find a map from Task to SOA
  - Minimize Task Execution Time

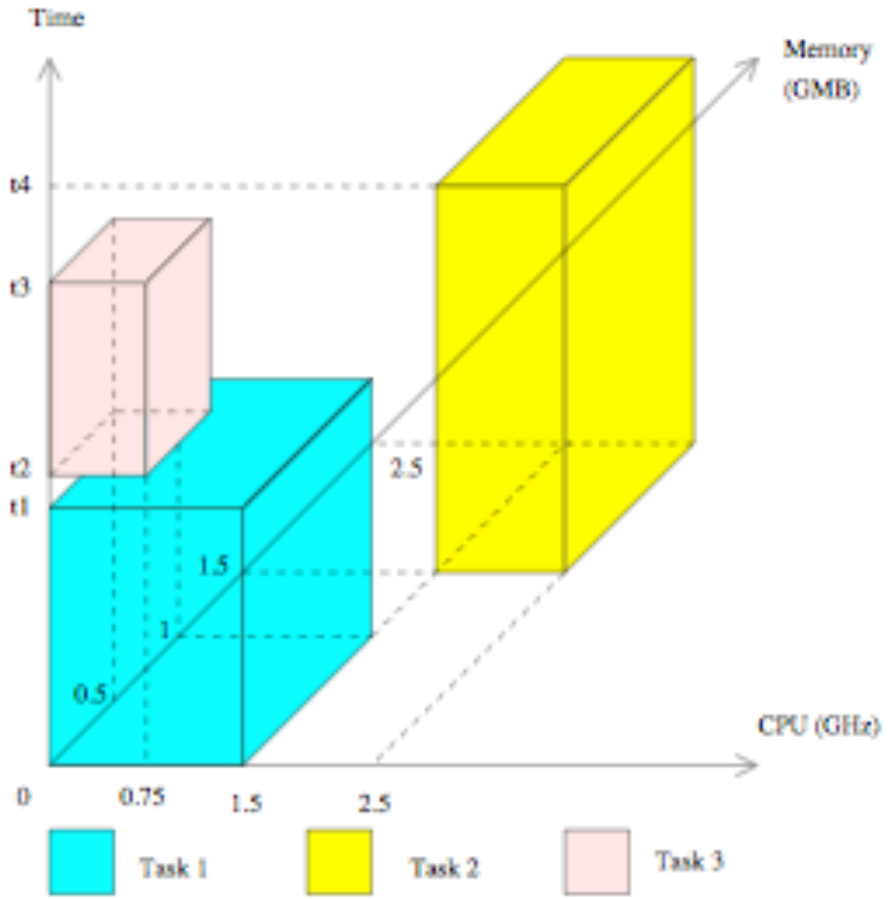
The scheduling framework



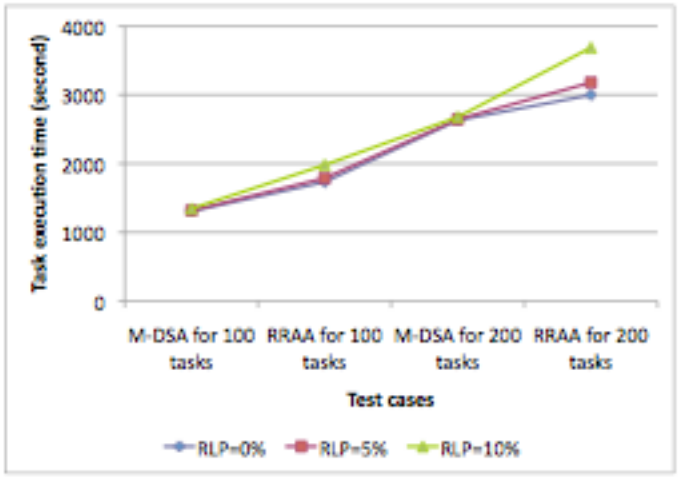
Task scheduling algorithm

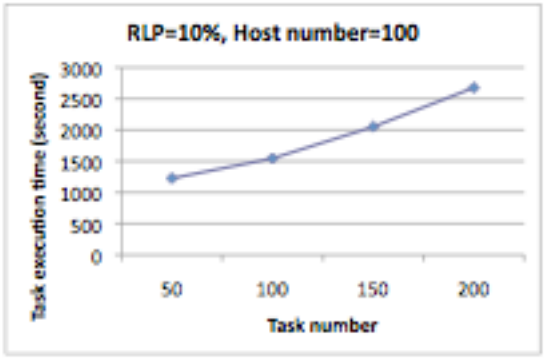
- Generate DAGs form parallel tasks
- Sort jobs in a parallel task in priority with job dependency
- Scheduling algorithm
  - Greedy algorithm
  - Get the first job from sorted job list
  - Find a resource that fit job's requirement in multi-demension

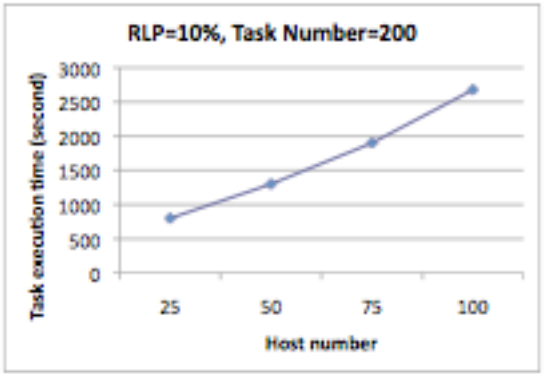
An demonstration example



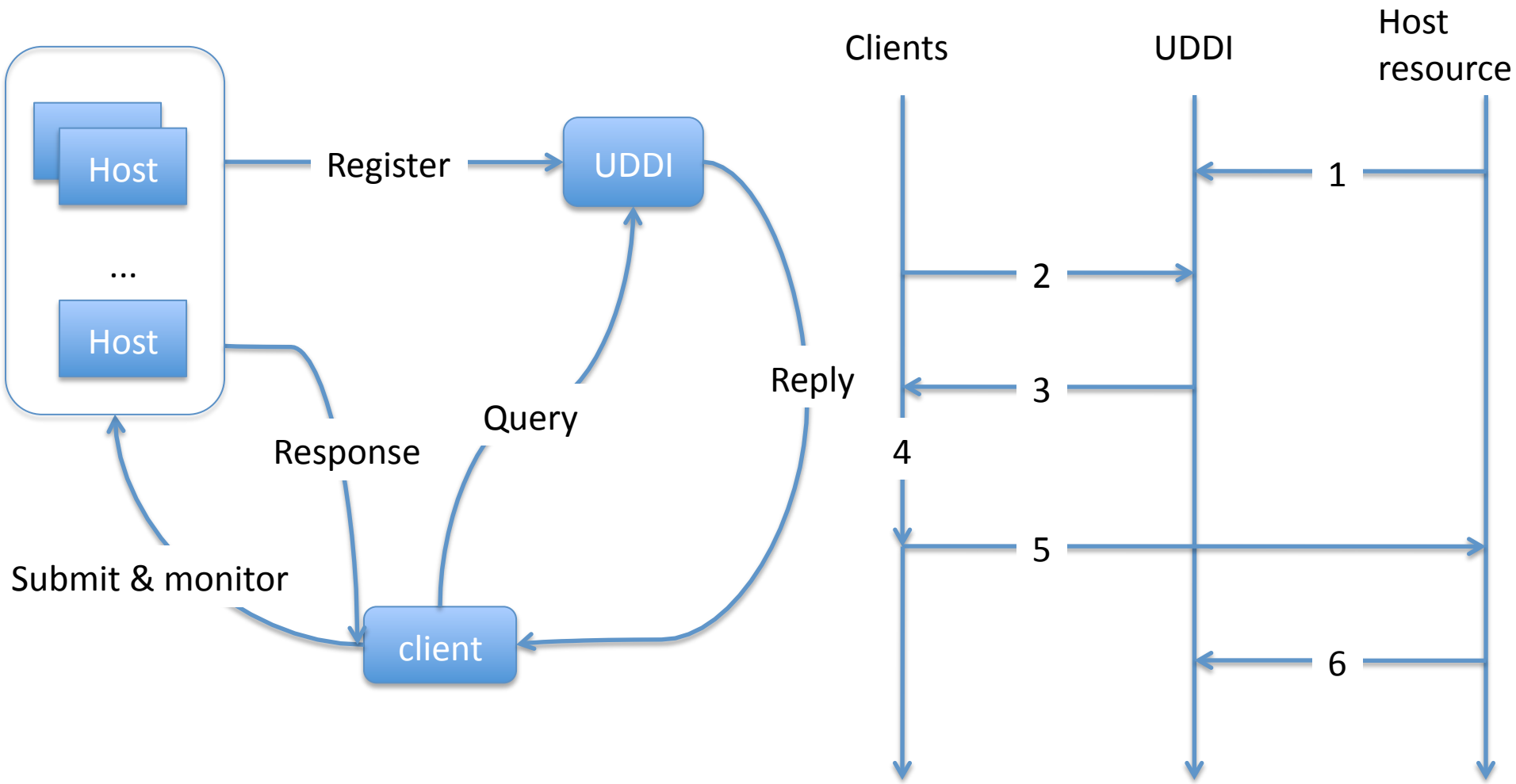
Simulation results



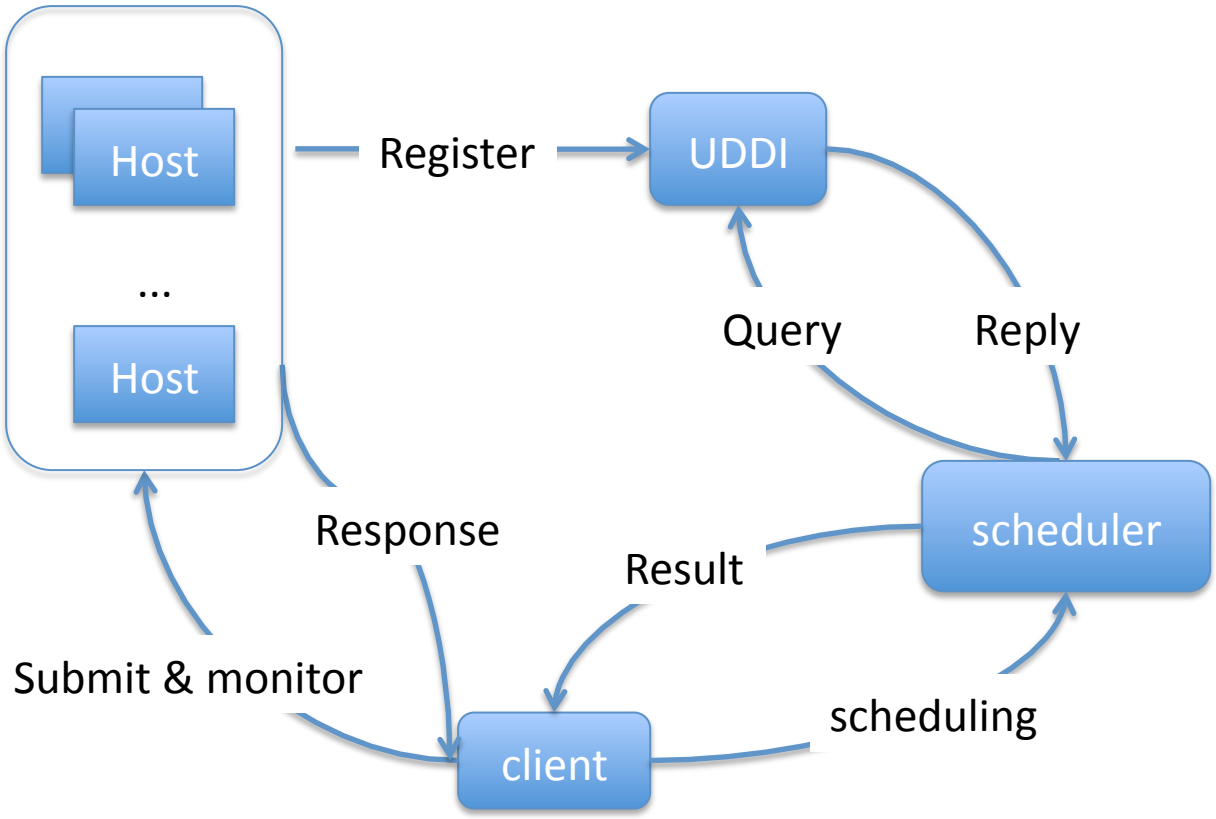




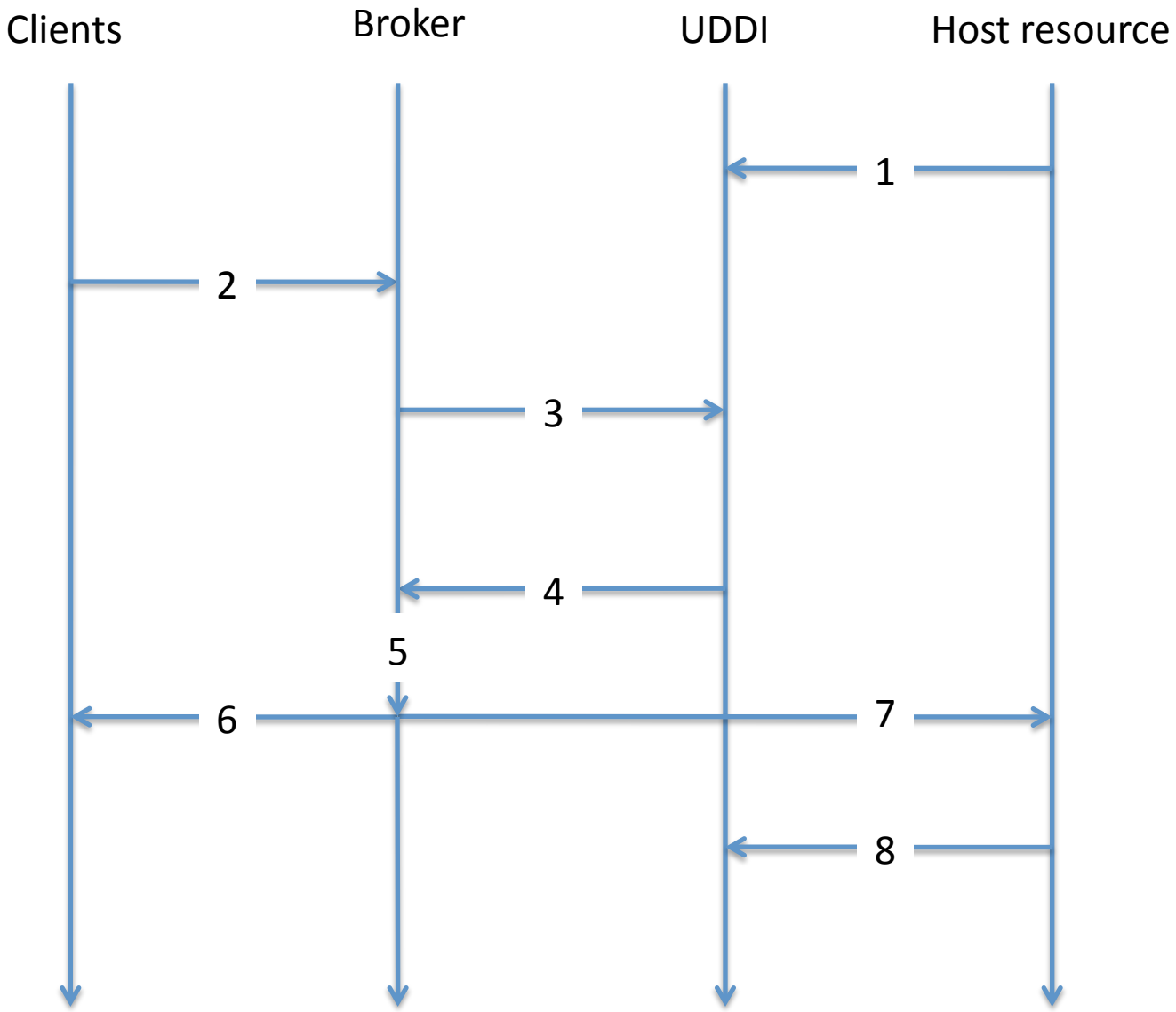
A test SOA environment



Implementation



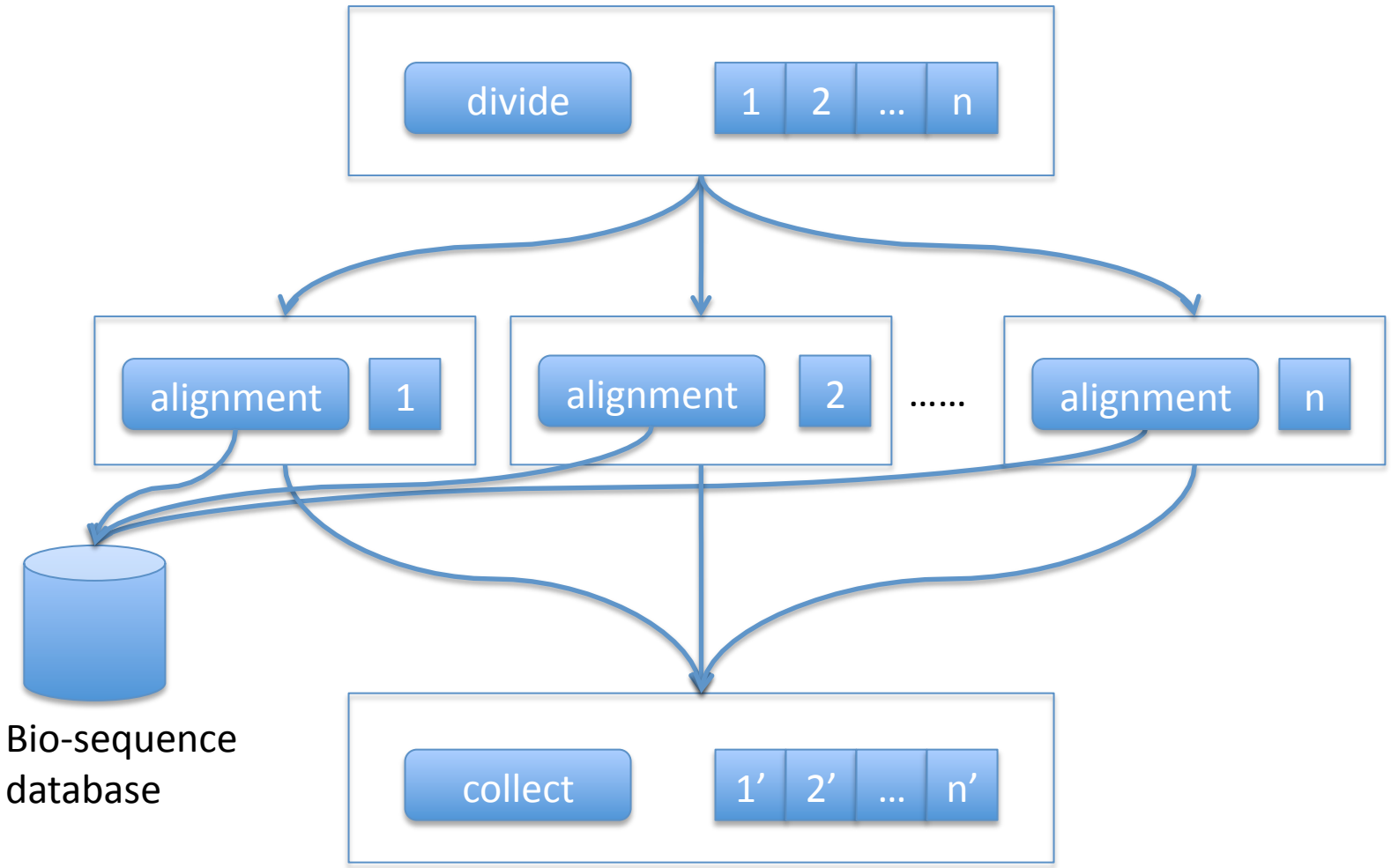
Implementation SOA execution sequence diagraph



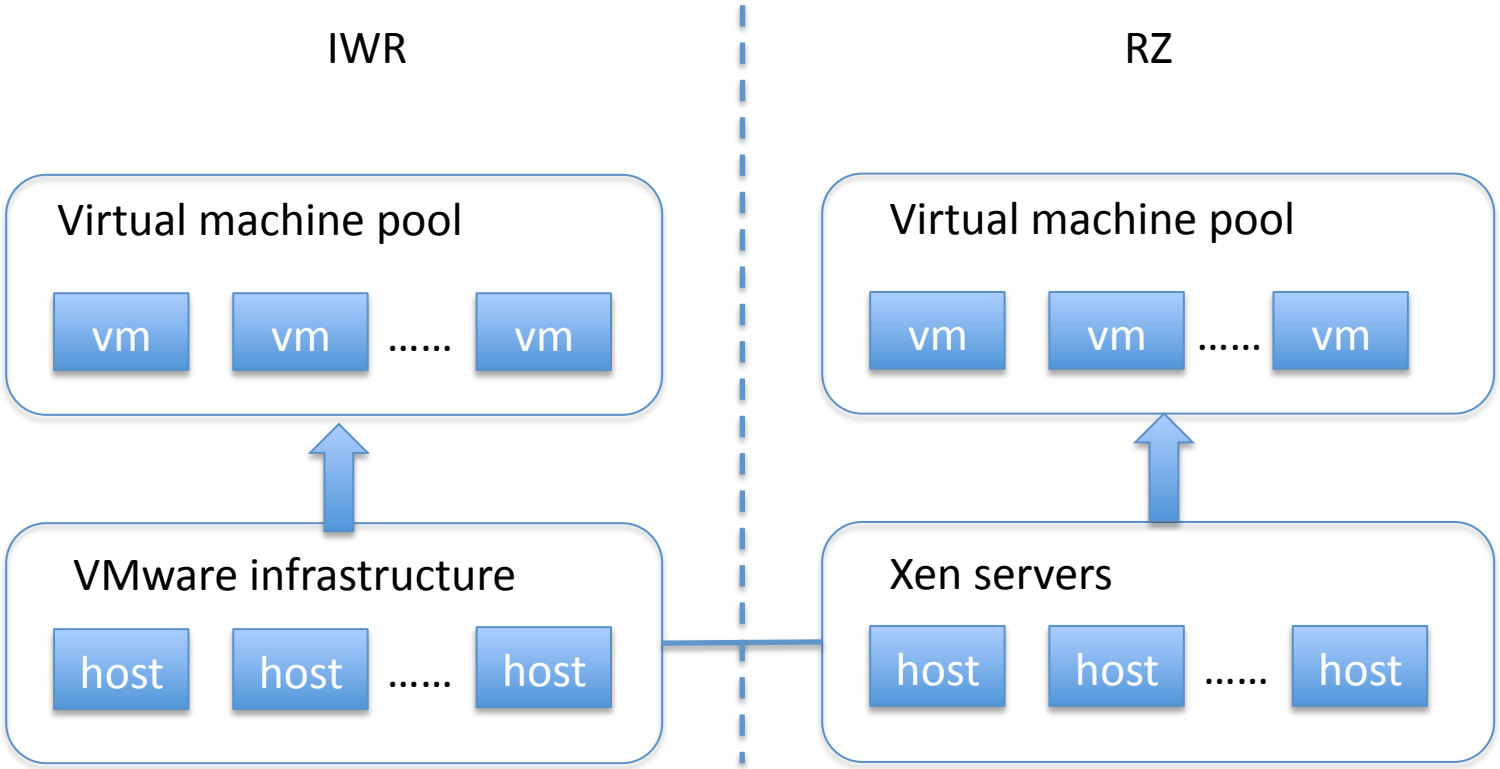
Bio-sequence Alignment: a Use Case

- The process of bio-sequence alignment is to execute the Smith-Waterman algorithm and compare the query sequence with the bio-database.
- Master-slave paradigm
- Use distributed VM to execute slave jobs

Bio-sequence Alignment: a Use Case



Test bed


IWR
RZ
Virtual machine pool
vm
vm
.....
vm
Virtual machine pool
vm
vm
.....
vm
VMware infrastructure
host
host
.....
host
Xen servers
host
host
.....
host

Test result

Table 1: Test results of bio-sequence alignment

Task Name	CPU (GHz)	Memory (MB)	Req. TET (Second)	Actual TET (Second)
divide	0.5	500	60	52
$align_x$	1	1000	5800	5632
collect	0.5	500	60	44

Conclusion

- Multiple dimension VM scheduling in a SOA
- Implement in a SOA environment
- Make a performance study in a wide area