STROLL: A Universal Filesystem-based Interface for Seamless Task Deployment

Abdulrahman Azab, Hein Meling, Josef Kejzlar
University of Stavanger

Abstract

Developing applications for solving compute intensive problems is not trivial. And despite recent availability of a range of Grid computing platforms, domain specialist programmers and scientists only rarely take advantage of these new computing facilities. One reason for this is the complexity of Grid computing, and the need to learn a new programming environment to interact with the Grid. Typically, only a few programming languages are supported, and often scientists use special-purpose languages that are not supported by most Grid platforms. Moreover, Grid users cannot easily deploy their compute tasks to multiple Grid platforms without rewriting their program to use different task submission interfaces. In this paper we present Stroll, a universal file system-based interface for seamless task submission to one or more Grid computing facilities. The user interacts with the Grid through simple read and write file-system commands. Stroll allows all categories of users to submit and manage compute tasks both manually, and from within their

programs, which may be written in any language. We have implemented Stroll on both Windows and Linux, and we demonstrate that we can submit the same compute tasks to both Condor and Unicore clusters. In the evaluation, we show that the overhead of Stroll is negligible. We also compare the code complexity of the compute task, written using our Stroll interface with an implementation using a custom Java-based Grid API.

To submit jobs to a Grid system you need to learn how to:

1. Prepare your input files 2. Write a detailed submission script. 3. Submit your jobs through the front end.
4. Monitor the execution. 5. Collect the results.

Do scientists have time for this ?

Current Solutions

Solution	Examples	Drawbacks
Grid Portals	WebSphere, WebLogic, GridSphere, GridPortlets	Useful for manual submission. In many cases, it is required to perform job submission automatically from a user code.
Web Services	Birdbath (condor), GRAM (Globus), GridSAM	The programming language has to support the technology and the user must have the proper experience. This is not the case for many low level special purpose languages and most of the scientists
Grid APIs	DRMAA, HiLA, CondorAPI, GridR	

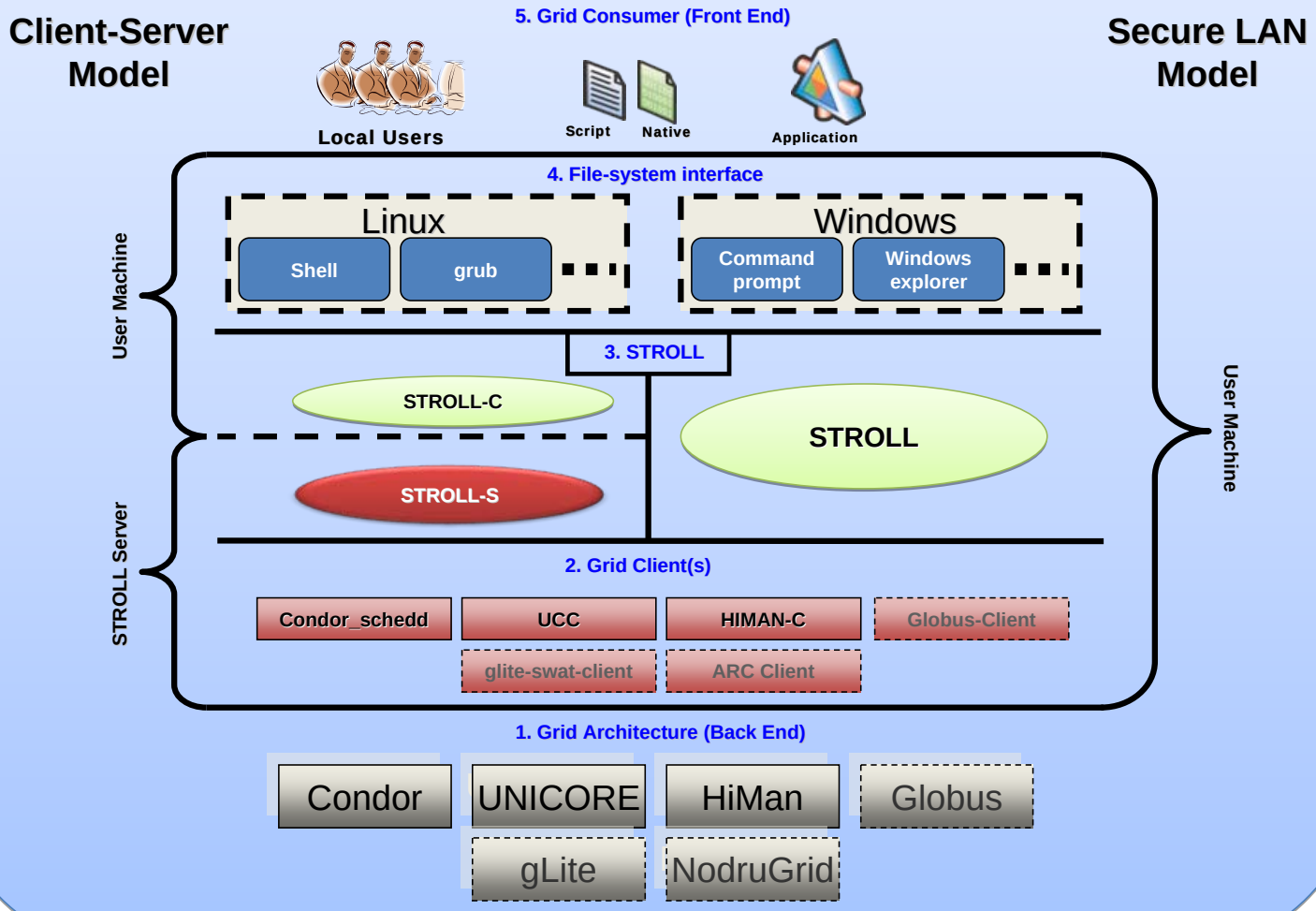
Proposed Solution

Grid Access File system Interface (Stroll)

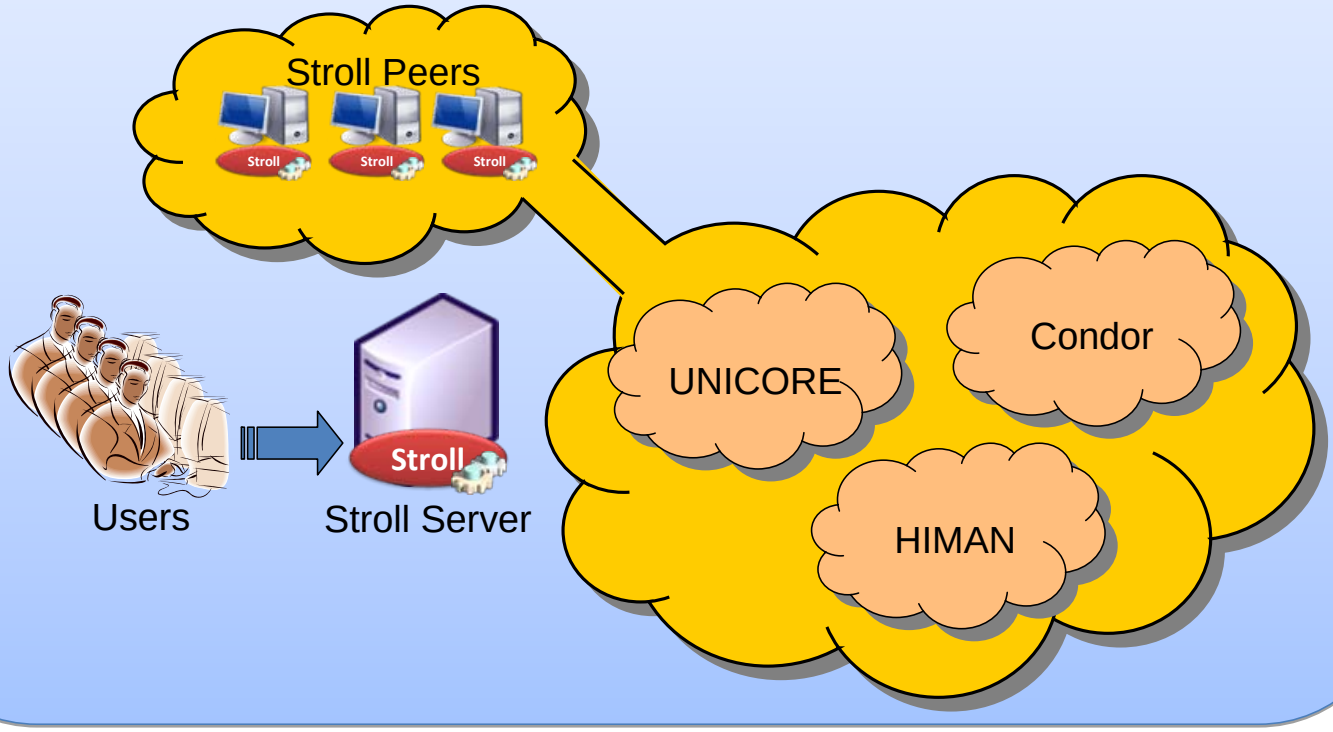
submission and management of grid jobs is carried out by executing **simple** read() and write() file system commands.

- This technique allows all categories of users to submit and manage grid jobs both manually and from their codes which may be written in any language.

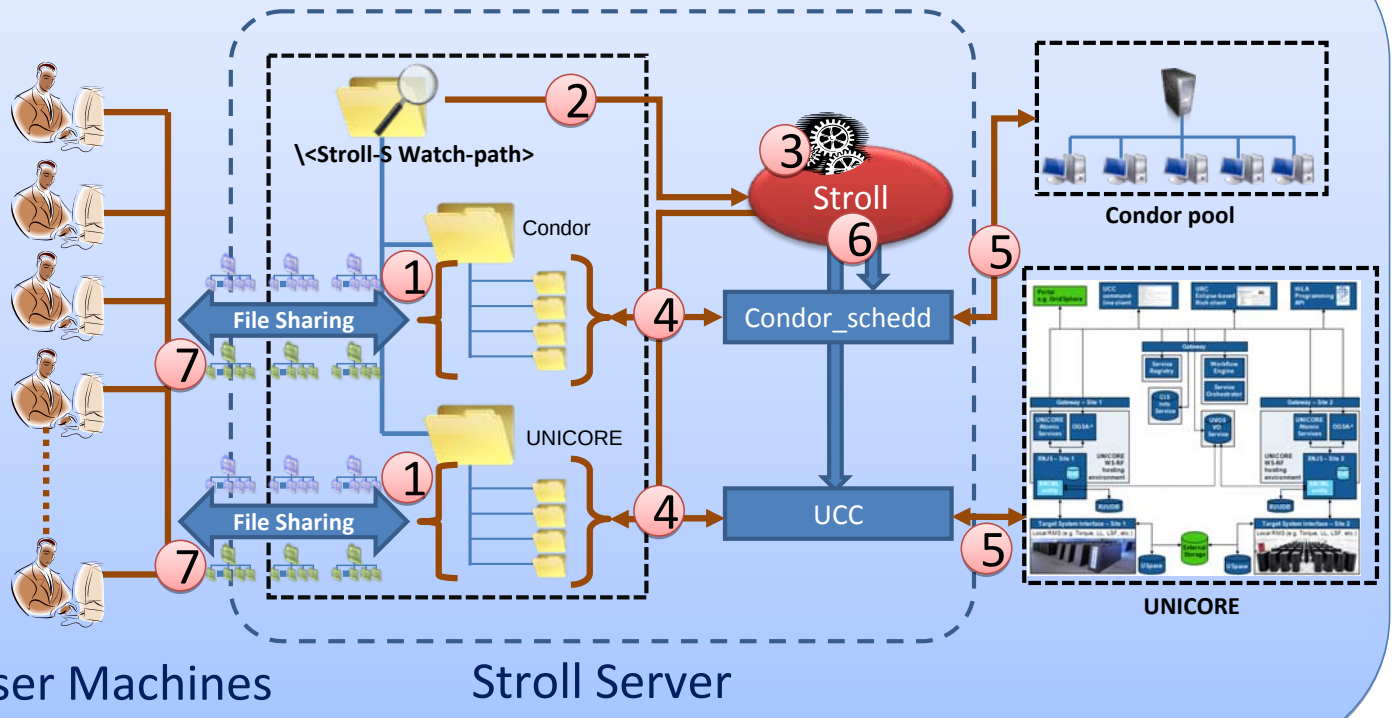
System Architecture



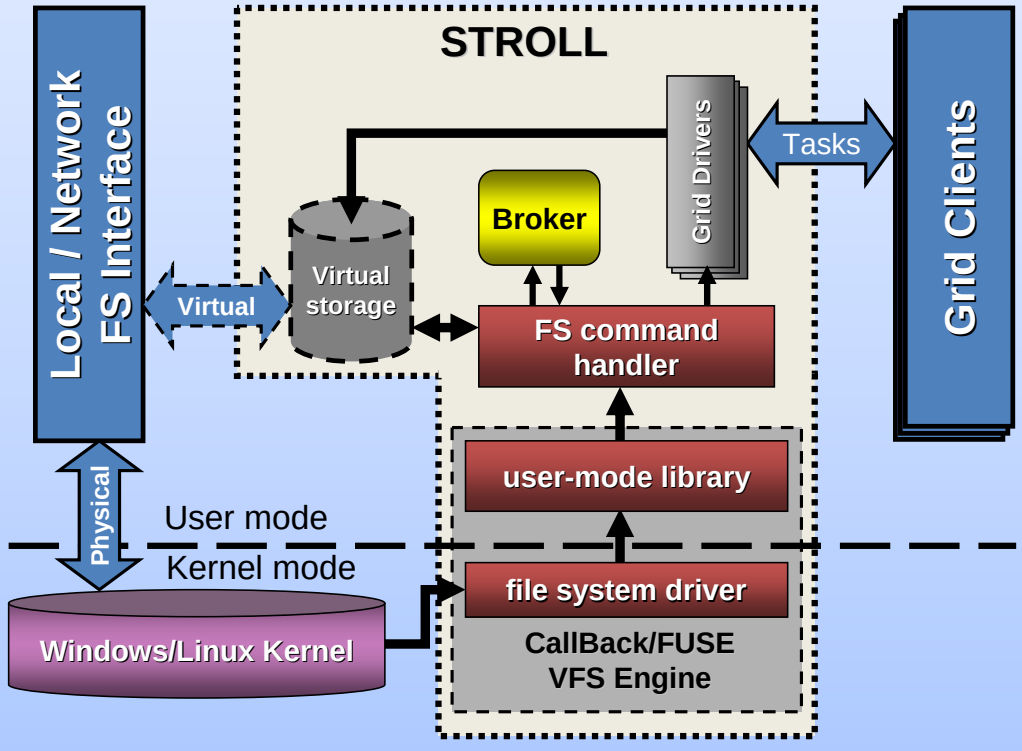
Environment

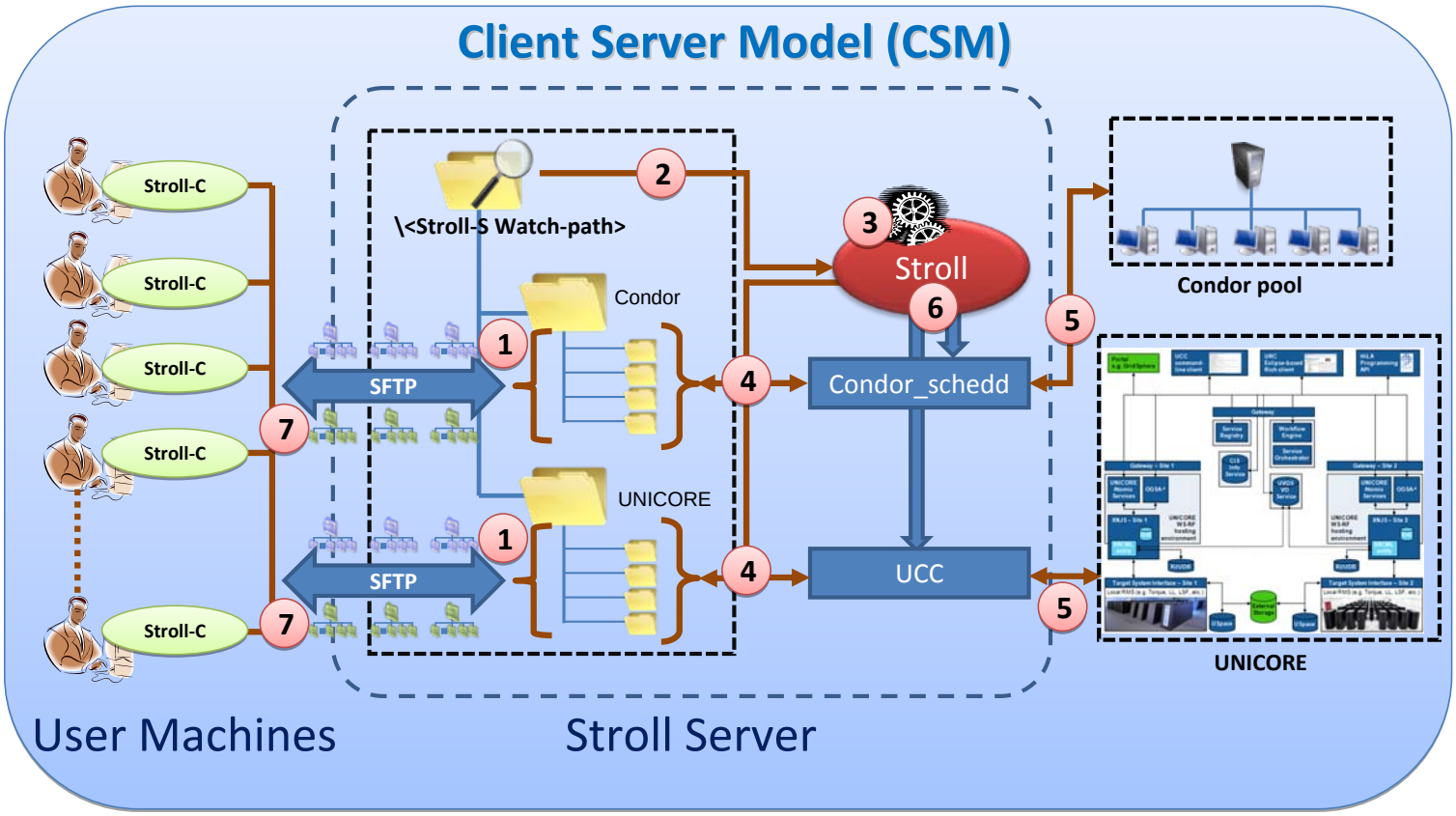


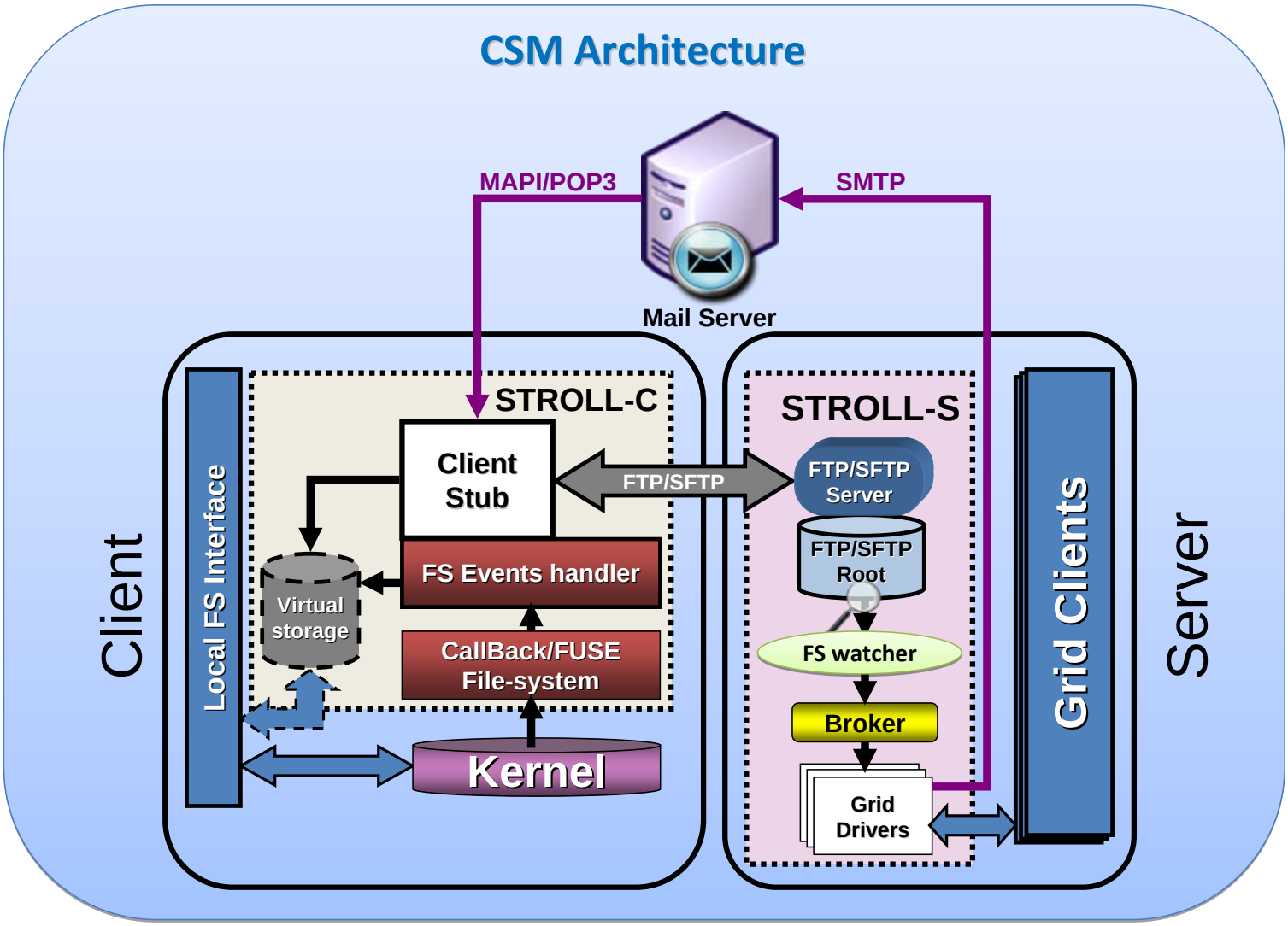
Secure LAN Model (SLM)



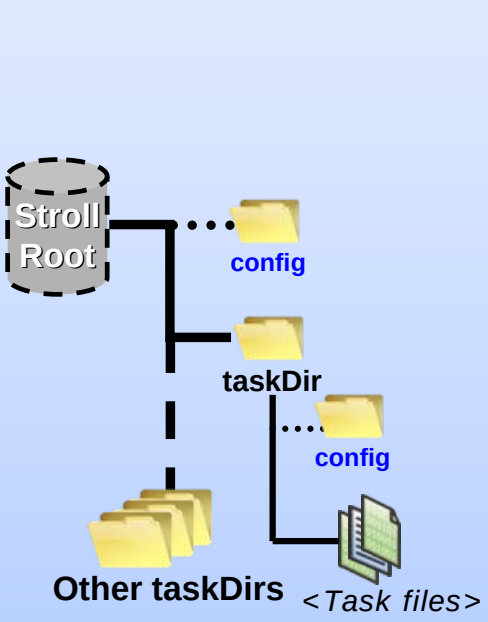
SLM Architecture

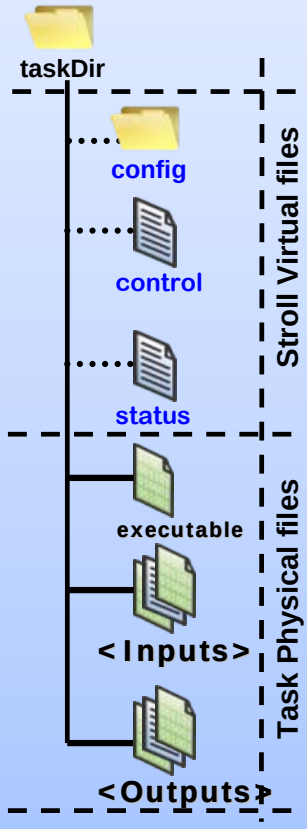



Client Server Model (CSM)
Stroll-C
Stroll-C
Stroll-C
Stroll-C
Stroll-C
\<Stroll-S Watch-path>
Condor
UNICORE
SFTP
SFTP
Stroll
Condor_schedd
UCC
Condor pool
UNICORE
1
2
3
4
5
6
7
User Machines
Stroll Server

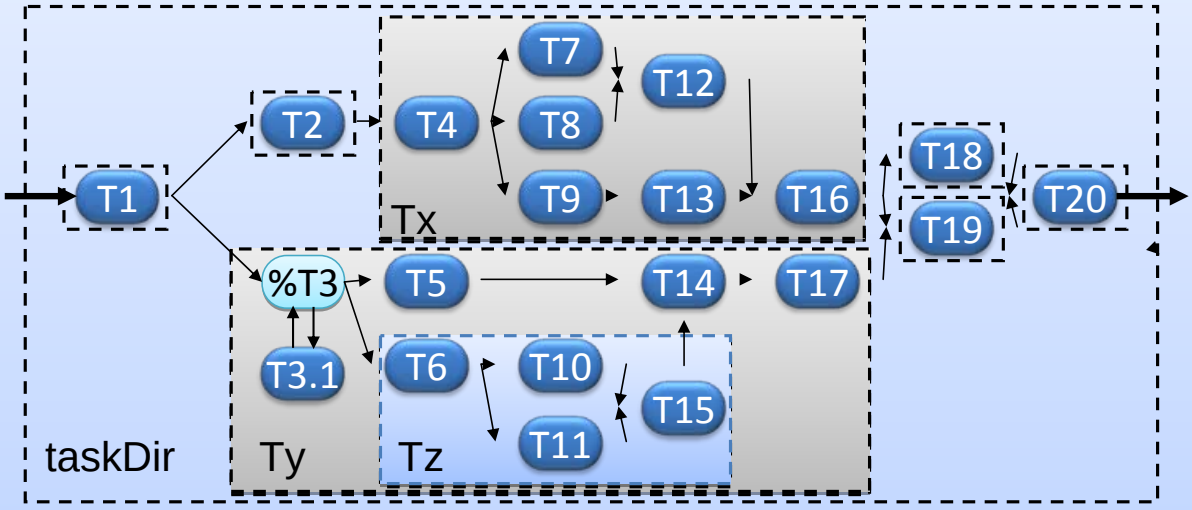

CSM Architecture
MAPI/POP3
SMTP
Mail Server
Client
Local FS Interface
STROLL-C
Client Stub
FTP/SFTP
FS Events handler
Virtual storage
CallBack/FUSE File-system
Kernel
STROLL-S
FTP/SFTP Server
FTP/SFTP Root
FS watcher
Broker
Grid Drivers
Grid Clients
Server

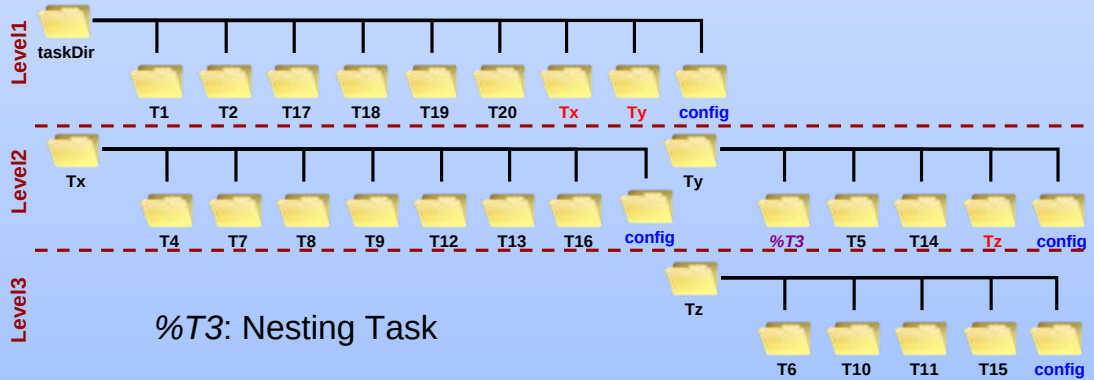
Task Structure





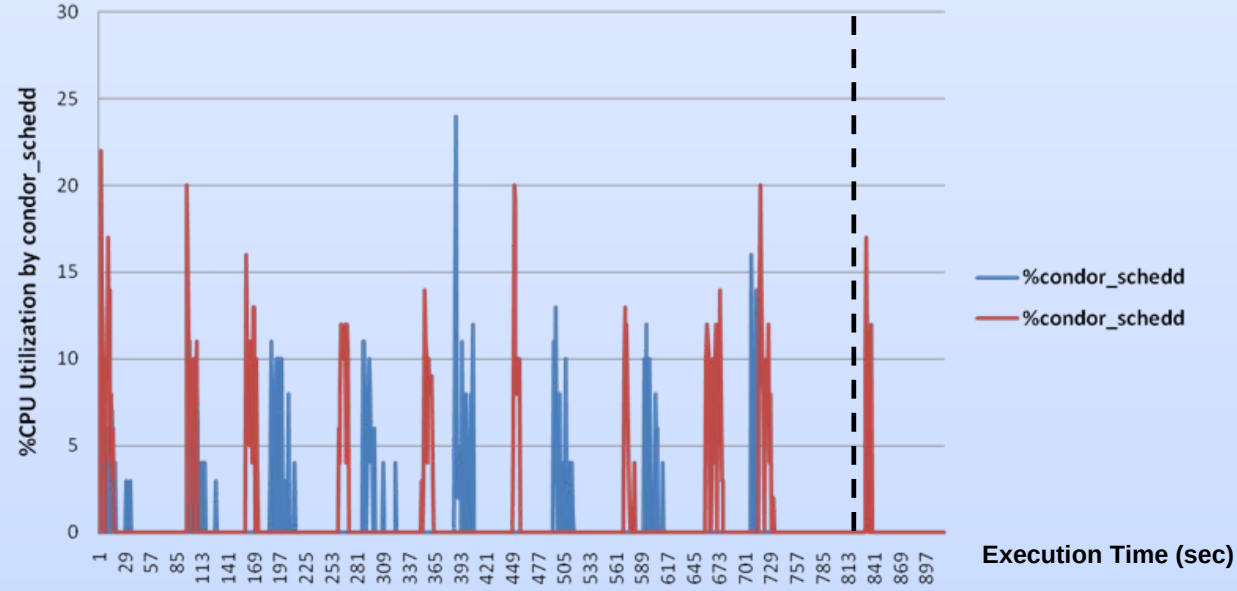
Composite Tasks

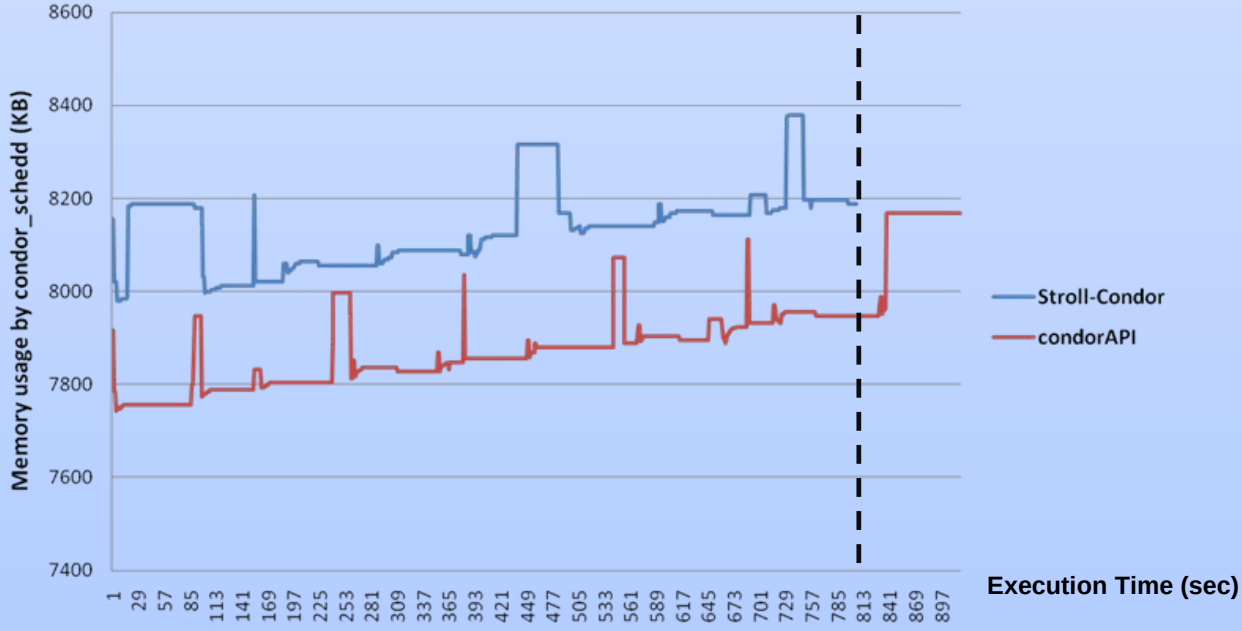




Performance evaluation (Condor)

CPU utilization of R process during the execution of a parallel version PSM.estimate() statistical modeling function on Condor





Performance evaluation (UNICORE)

CPU utilization of R process during the execution of a parallel version PSM.estimate() statistical modeling function on UNICORE