Provenance of the Workflow Refinement Process

Ewa Deelman
USC Information Sciences Institute
Refinement provenance

- We not only consider the provenance of the executing application but also of the refinement process that maps an abstract workflow (workflow instance) onto a set of resources.
- The refinement process can be multi-staged.
- Stages of the refinement can execute on a variety of resources.
- We capture provenance of the entire workflow as well as workflow constituent.
- The representations for the refinement and workflow provenance are uniform.
The Workflow is Partitioned into 2 pieces, then Workflow 1 is refined and executed and then Workflow 2 is refined and executed.
Based on available data products Workflow 2 is reduced. In this case files Resliced Image 1, Resliced Header 1, Resliced Image 2, and Resliced Header 2 are found on Storage systems S1 and S2, so they do not need to be recomputed.
Site Selection: the location where the workflow components are to be executed are selected.

In our workflow we have the following resources available R1, R2 and R3

Workflow 5
Data stage-in and data stage out nodes are added to the workflow. Data is moved from the storage systems to the computations, data is transferred between compute resources and some intermediate data products are staged out to long term storage.
Data registration nodes are added to the workflow.
Nodes that are destined to the same resources can be clustered together to improve the overall workflow performance. The workflow is ready for execution and is sent to a workflow engine.
Upon successful execution of partition 1, partition 2 is refined, it is possible that this partition will skip some refinement steps. Here we perform site selection.
Data transfer nodes are added.

Workflow 10
Workflow 11

Data registration nodes are added

- Resliced Image 1 (Tr(S1->R1))
- Resliced Header 1 (Tr(S1->R1))
- Resliced Image 2 (Tr(S2->R1))
- Resliced Header 2 (Tr(S2->R1))
- Softmean at R1
- Atlas Image (Tr(R1->R2))
- Atlas Header (Tr(R1->R2))
- Atlas Image (Tr(R1->R3))
- Atlas Header (Tr(R1->R3))
- Slicer at R1
- Slicer at R2
- Slicer at R3
- Convert at R1
- Convert at R2
- Convert at R3
- Atlas X (Tr(R1->S1))
- Register Atlas X
- Atlas Y (Tr(R2->S1))
- Register Atlas Y
- Atlas Z (Tr(R3->S1))
- Register Atlas Z
Nodes are clustered for performance.
Chain of Refinement and Execution Steps

- Workflow 1 -> Partition id1
- Workflow 2
- Workflow 3
- Workflow 4
- Workflow 5
- Workflow 6
- Workflow 7
- Workflow 8
- Workflow 9
- Workflow 10
- Workflow 11
- Workflow 12

Control Dependency

Ewa Deelman, deelman@isi.edu
www.isi.edu/~deelman
pegasus.isi.edu
Definition of refinement and execution provenance

<object id>
  [[I/O] data input/output
   [function performed]
   [performance info]
   [optional annotations]]

Could include a justification of the reasons for the tasks performed
Provenance records relating to the refinement process

$id1$:[l: [<workflow1> O: <<workflow2>; <workflow3>]
[<partition>][<host22><2 mins@Á @]
[<planning horizon set at 5 hours>]

$id2$:[l: [<workflow2> O: <workflow4>]
]tgf wevvp@]>Á øø]>Á Á @

$id3$:[l: [<workflow4> O: <workflow5>]
]ukg"ugmvevvp@]>Á øø]>Á øø

$id4$:[l: <workflow5> O: <workflow6>]
[<data
    stagein>][<host12><10mins>..][]

$id5$:[l: <workflow6> O: <workflow7>]
]tgi kvtkvp@>Á ø]>wulpi "primary RLS host14>]

$id6$:[l: <workflow7> O: <workflow8>]
[<clustering>][<host12><12mins>]
[]

$id7$:[l: <workflow8> O: < Ø>]
[<dagman_exec>][]

Ewa Deelman, deelman@isi.edu

www.isi.edu/~deelman

pegasus.isi.edu
Provenance of the execution process
(both at the workflow and task levels)

<Workflow8>[I:<AnatomyImage3@S1><AnatomyHeader3@S1><ReferenceImage@S2><ReferenceHeader@S2><AnatomyImage4@S1><AnatomyHeader4@S1>]

[O:<WarpParams3><WarpParams4><ReslicedImage3@S1><ReslicedHeader3@S1><ReslicesImage4><ReslicedHeader4>]

[<description of tasks in workflow 8> (could be in a form of a DAX (XML-DAG used by Pegasus)),
 <task_id1_align_warp><task_id2_align_warp><task_id3_reslice><task_id4_reslice>]

[R1,R2]<20hours (cumulative time @Á 0] __
<task_id1_align_warp>[I:<AnatomyImage3@S1><AnatomyHeader3@S1>,
 <ReferenceImage@S2><ReferenceHeader@S2> O:<WarpParams3>]

]T3@3j t@Á __
<task_id2_align_warp>[I:<AnatomyImage4@S1><AnatomyHeader4@S1>,
 <ReferenceImage@S2><ReferenceHeader@S2> O:<WarpParams4>]

]T4@304j t@Á __
<task_id3_reslice@Á 0__
<task_id4_reslice@Á Â _
Some observations

- The provenance records for a workflow can be in pieces
  - for example for each partition there is a different set of records
  - need an aggregator agent to put the records together
- Sometimes a refiner or workflow engine may not give many internal details, but can still provide the basic information:
  - I/O
  - Cumulative runtime
  - others
Acknowledgments

- This provenance challenge is done in collaboration with Yolanda Gil, Jihie Kim, Gaurang Mehta, and Varun Ratnakuari
- The provenance model is being developed in collaboration with Ian Foster and Carl Kesselman
- Thanks to Luc Moreau for fruitful discussions