An Integrated Framework for Parameter-based Optimization of Scientific Workflows

Vijay S Kumar, *P. Sadayappan*

THE OHIO STATE UNIVERSITY

G. Mehta, K. Vahi, V. Ratnakar, Jihie Kim, Ewa Deelman, Yolanda Gil

ISI Information Sciences Institute

Mary Hall

THE UNIVERSITY OF UTAH

Tahsin Kurc, Joel Saltz

EMORY UNIVERSITY | Center for Comprehensive Informatics

Motivations

- Performance of data analysis applications is influenced by **parameters**
  - optimization ➔ search for optimal values in a multi-dimensional parameter space

- A systematic approach to:
  - enable the tuning of performance parameters (i.e., select optimal parameter values given an application execution context)
  - support optimizations arising from **performance-quality trade-offs**

Contributions of this paper

- No auto-tuning yet (work in progress)

- Core framework that can
  - support workflow execution (with application-level QoS) in distributed heterogeneous environments
  - enable manually tuning of parameters simultaneously
  - allow application developers and users to express applications semantically
  - leverage semantic descriptions to achieve performance optimizations
    - customized data-driven scheduling within Condor

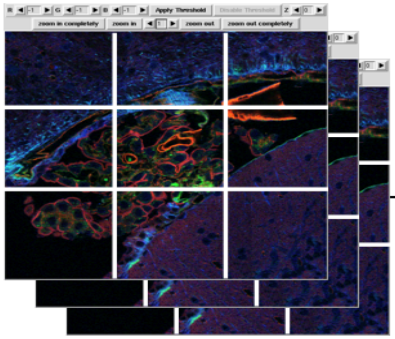
Application characteristics

- Workflows: Directed Acyclic Graphs with well-defined data flow dependencies
  - mix of sequential, pleasingly parallelizable and complex parallel components
  - *flexible* execution in distributed environments

- Multidimensional data analysis
  - data partitioned into chunks for analysis
  - dataset elements bear *spatial relationships, constraints*
  - data has an inherent notion of *quality* ➔ applications can **trade accuracy of analysis output for performance**

- End-user queries supplemented with application-level QoS requirements

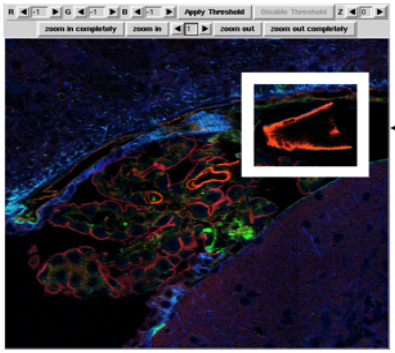
Application scenario 1: No quality trade-offs



correctional tasks

zproject → prenormalize → normalize → autoalign

mst → stitch → reorganize → warp

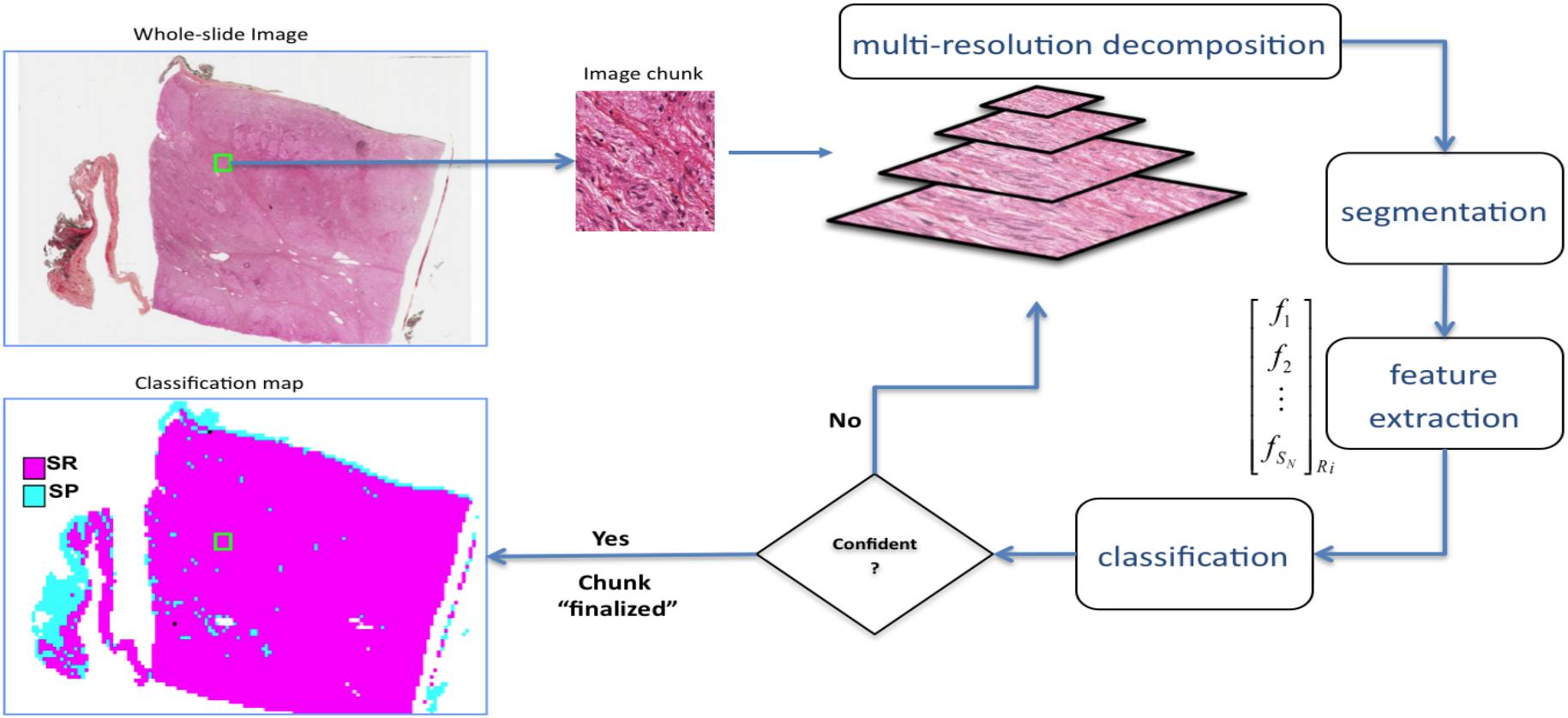


preprocessing tasks

prefix sum generation → tessellation → thresholding

- Minimize makespan while *preserving highest output quality*
- Scale execution to handle terabyte-sized image data

Application scenario 2: Trade quality for performance



- Support queries with application-level QoS requirements
  - *"Minimize time to classify image regions with 60% accuracy"*
  - *"Maximize classification accuracy of overall image within 30 minutes"*

Performance optimization decisions

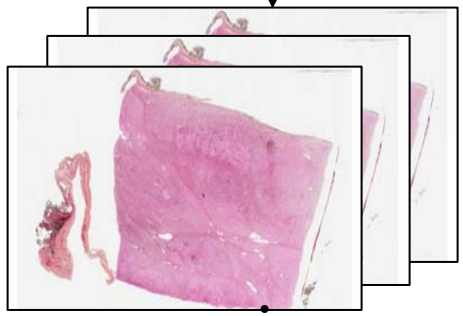
	Component-level decisions	Workflow-level decisions
Quality – preserving decisions	• What algorithm to use for this component? • What data-chunking strategy to adopt?	• Where to map each workflow component? • Which components to merge into meta-components?
Quality – trading decisions	• What is the quality of input data to this component ? • What is the processing order of the chunks?	• Which components need to perform at lower accuracy levels?

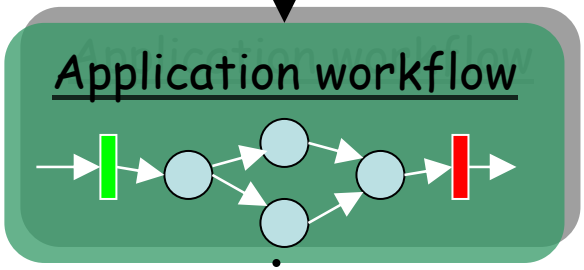
View each decision as a parameter that can be tuned

Conventional Approach

workflow design

datasets





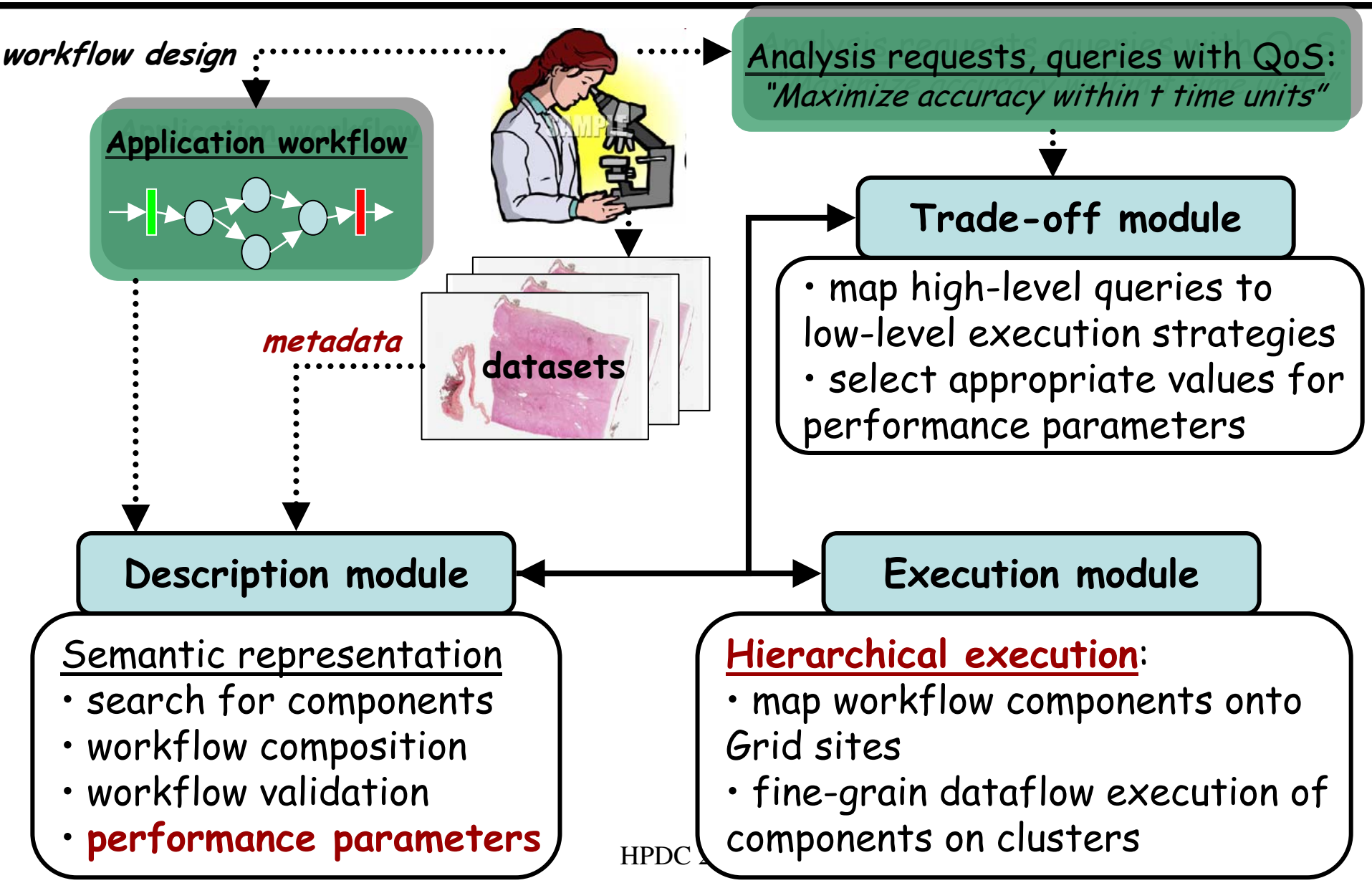
Workflow Description

Semantic representation
- component discovery
- workflow composition
- workflow validation

Workflow Execution

- clusters, the Grid or SOA
- task-based / services-based
- batch mode / interactive

Proposed approach: extensions

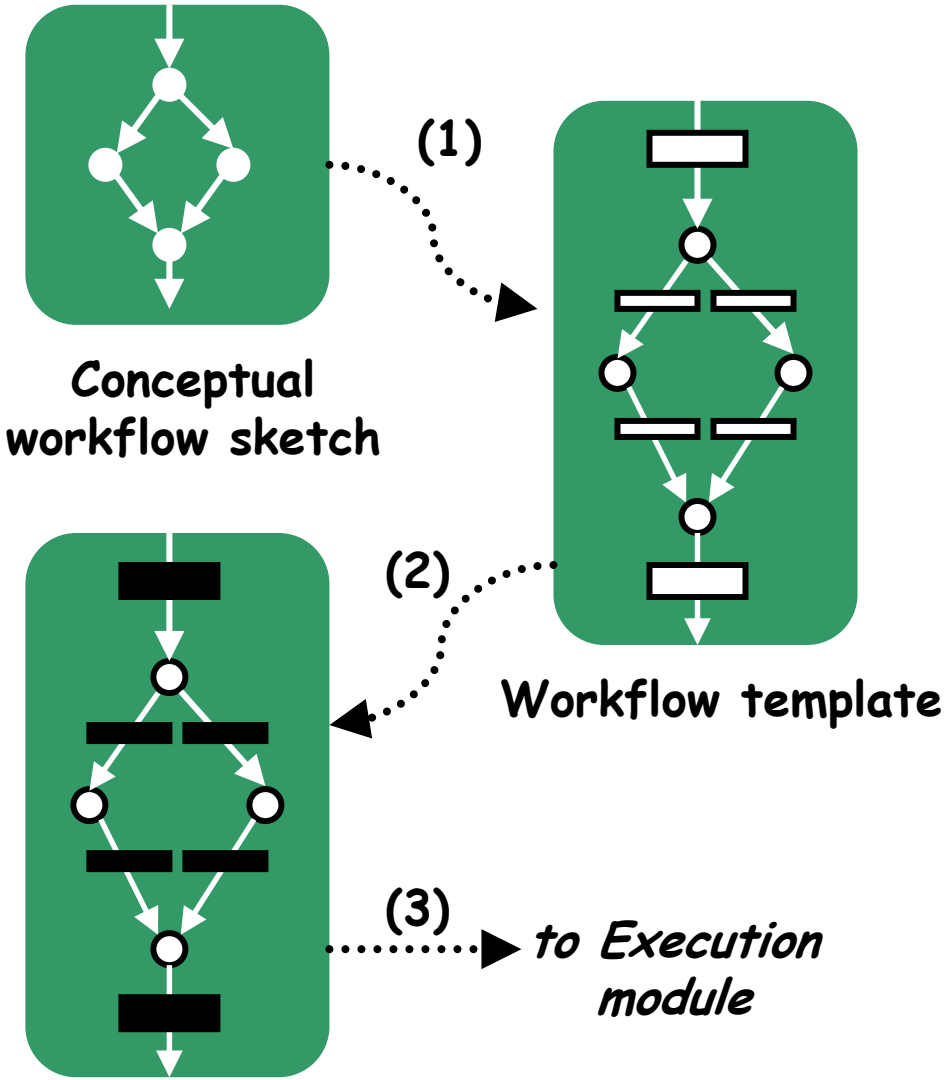


An instance of our proposed framework

PARAMETERS

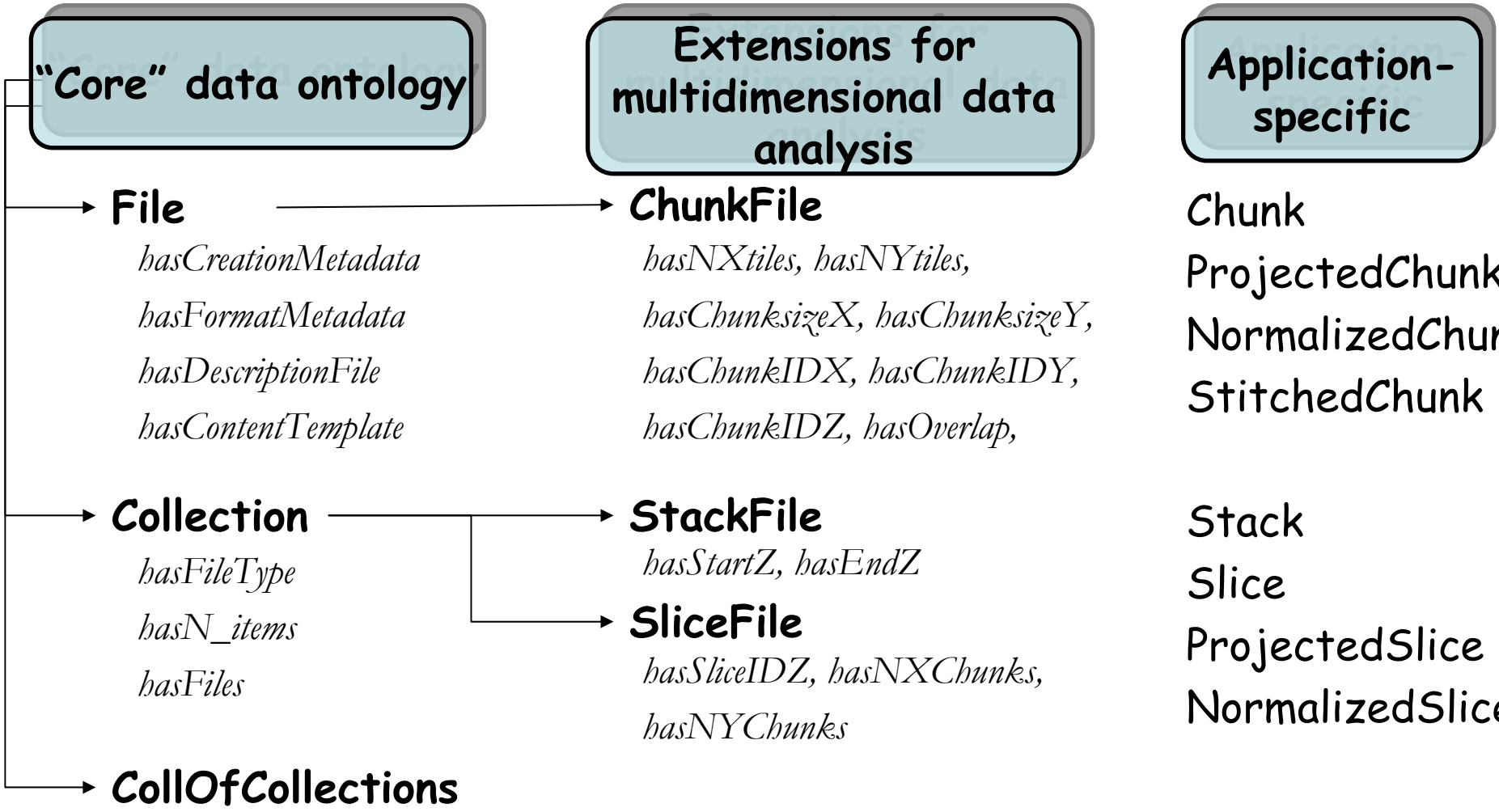
- **Description module** → **WINGS** (Workflow INstance Generation and Selection)
- **Execution module** → **Pegasus WMS**, **DataCutter**, **Condor, DAGMan**
- **Trade-off module** → Interacts with the description and execution modules

Description Module: WINGS
(Workflow Instance Generation and Selection)



- **Layered workflow refinement**
- Workflow Template:
  - abstract description
  - dataset-independent
  - resource-independent
- Compact workflow Instance:
  - contains mappings to actual datasets
  - resource-independent
- Expanded workflow instance

Extensions to WINGS data ontology



- Relations between entities, constraints on metadata
- Automatic description, naming of intermediate data products

Execution Module

Pegasus WMS (*http://pegasus.isi.edu*)

- Coarse-grain mapping of workflow tasks onto Grid sites
- Submits sub-workflows to DAG schedulers at each site
- Automatic data transfer between sites (via GridFTP)

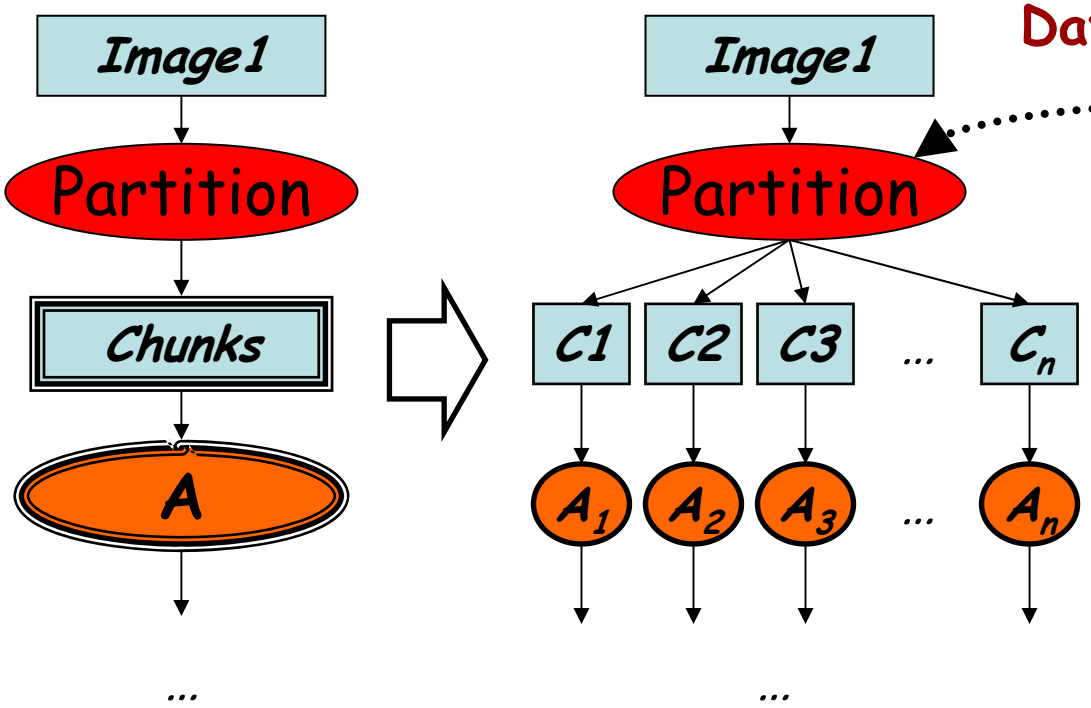
DataCutter (*http://datacutter.osu.edu*)

- Fine-grain mapping of components onto clusters
- Filter-stream model, asynchronous delivery
- Each filter executes as a thread (could be C++/Java/Python)
- Pipelined dataflow execution: Combined task- and data- parallelism
- MPI-based version (***http://bmi.osu.edu/~rutt/dcmpi***)

Condor (*www.cs.wisc.edu/condor*)

can now execute DataCutter jobs within its "parallel universe"

Quality-preserving parameters



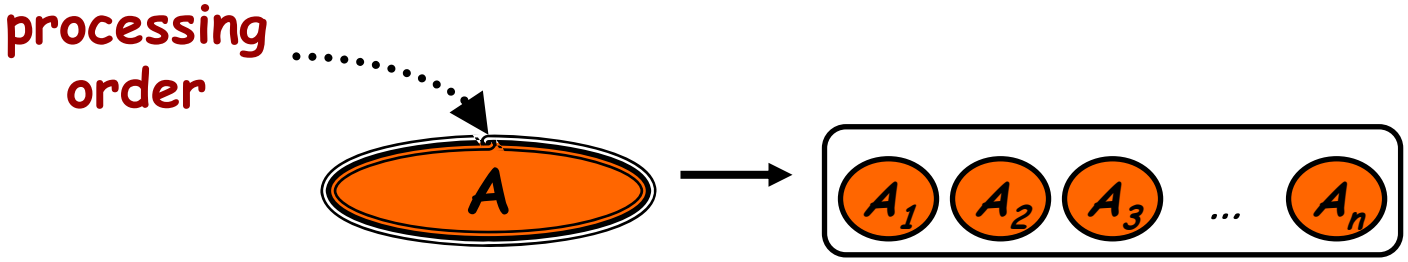
- algorithmic variant of a component
- component placement
- grouping components into meta-components
- task-parallelism and data streaming within meta-component

Quality-trading Parameters

- **Data approximation**
  - **e.g. spatial resolution of chunk**
  - **higher resolutions ➔ greater execution times, but does not imply higher accuracy of output**

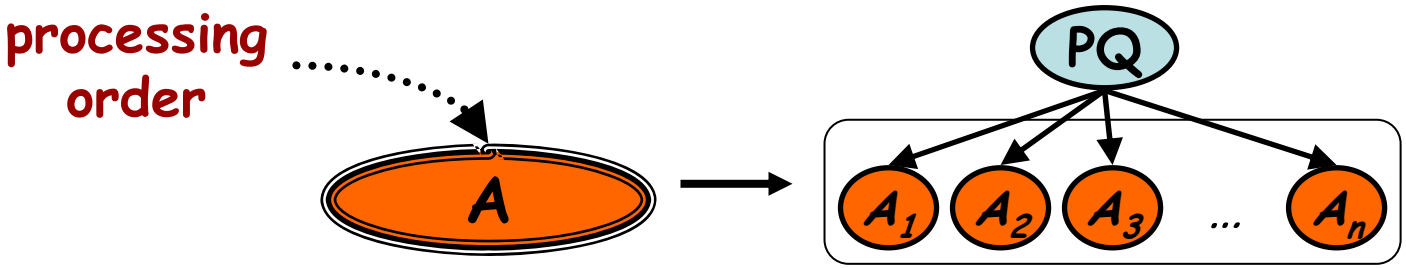
- **Processing order of chunks**
  - the order in which data chunks are operated upon by a component collection
  - can process "favorable" chunks ahead of other chunks

Processing order



- Tasks within a component collection treated as a batch
  - Condor: executes them in FIFO order
- Implemented a **priority-queue based heuristic** for reordering task execution for a component collection
  - "favorable" chunks are processed ahead of other chunks
  - different QoS requirements ➔ change the insertion scheme
- Can the execution of the bag-of-tasks be reordered dynamically?
  - *condor_prio* alone is not suitable

Customized scheduling in Condor



- Customized job scheduling within Condor to support performance-quality trade-offs for application-level quality-of-service (QoS)
  - implements the priority queue scheme (overrides the FIFO scheme)
  - executes within Condor's "scheduler" universe

- **Associates tasks with the spatial coordinates** of the respective chunks that are being processed
  - uses the automated naming of data products (metadata propagation) brought about by semantic descriptions

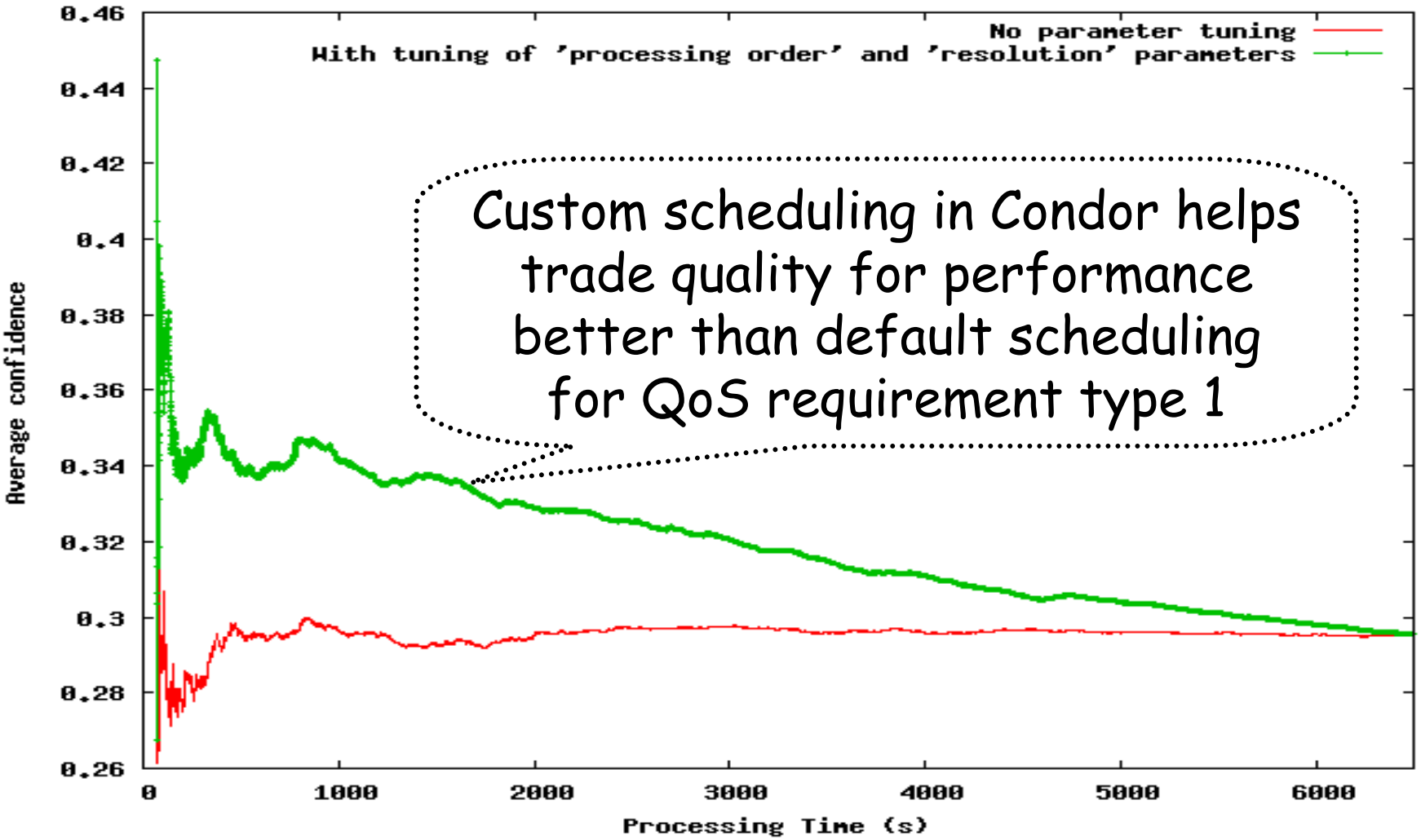
Experimental setup: Test bed

- RII-MEMORY
  - 64 node Linux cluster
  - Dual-processor 2.4 GHz Opteron nodes
  - 8GB RAM, 437 GB local RAID0 volume
  - Gigabit Ethernet
- RII-COMPUTE
  - 32 node Linux cluster
  - 3.6 GHz Intel Xeon processors
  - 2GB RAM, 10 GB local disk
  - Gigabit Ethernet and Infiniband
- Wide-area 10 Gbps connection

Performance Evaluation

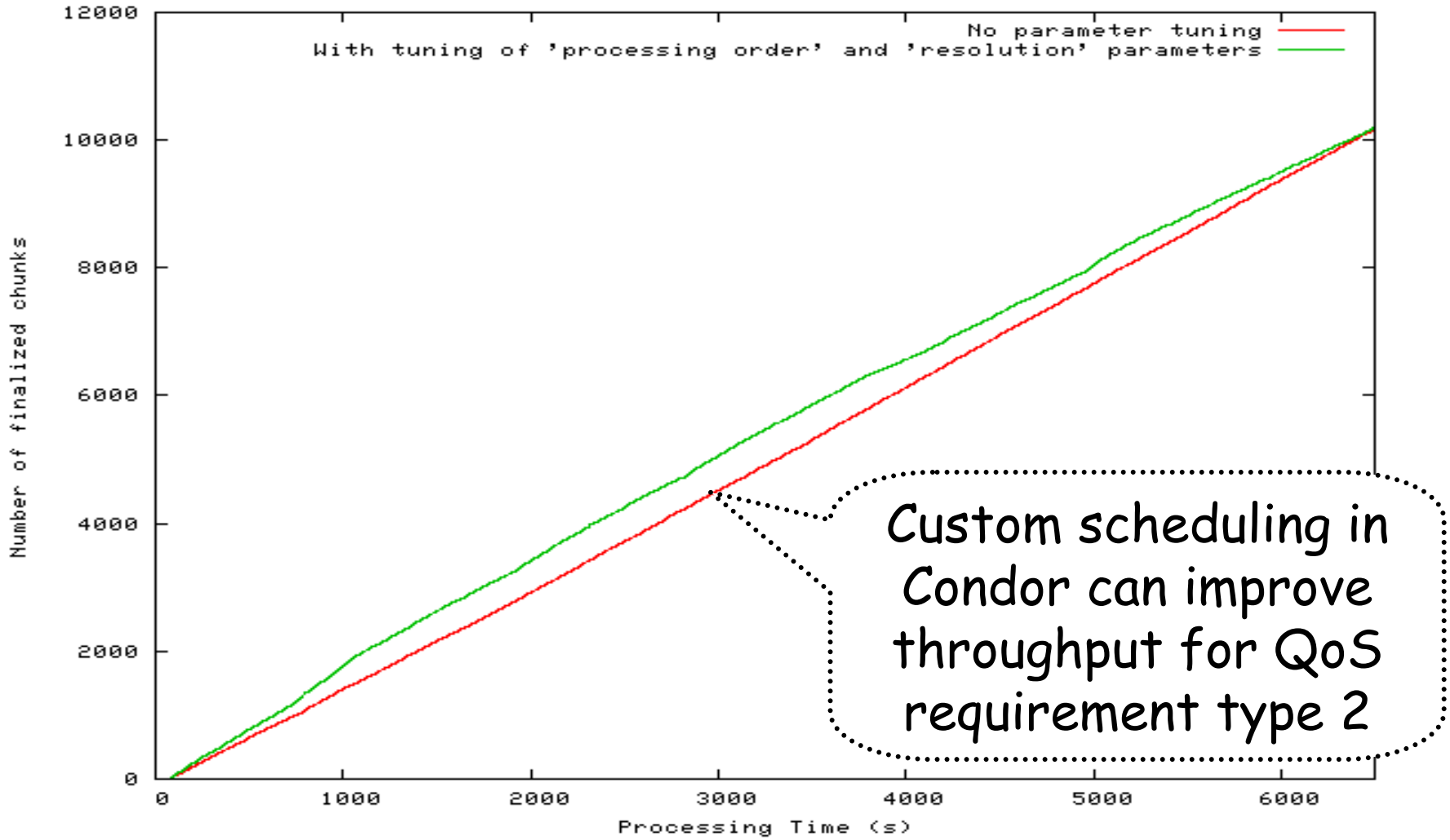
- Focus on performance-quality trade-offs
- *Neuroblastoma* Classification workflow:
  - "Maximize overall confidence of classification within *t* time units"
  - "Maximize number of data chunks processed within *t* time units"
- How to tune quality-trading parameters to achieve high performance?
  - Data resolution
  - Processing order of chunks

Parameters: resolution, processing order



- 32 nodes, 21 GB image, confidence threshold = 0.25
- "Maximize overall classification confidence within time *t* units"

Parameters: resolution, processing order



- 32 nodes, 21 GB image, confidence threshold = 0.25
- "Maximize data chunks processed within *t* time units"

Conclusions

- Performance optimization for workflows: search for values in a multidimensional parameter space
- Instance of our proposed framework allows users to manually express values for many performance parameters (simultaneously):
  - quality-preserving & quality-trading
- Semantic representations of domain data and performance parameters can be leveraged
  - Data chunking strategy and data approximation can help restructure workflow for a given resource configuration
  - Customized job scheduling within Condor can scalably support application-level QoS

Current and Future work

- Use semantic representations to map high-level queries onto low-level execution strategies
- Techniques to efficiently navigate the parameter space
  - Assume high data cardinality ➔ Uniformity of application context over time
  - Use information from sample runs to build statistical models

An Integrated Framework for Parameter-based Optimization of Scientific Workflows

Thanks!

Vijay S Kumar (vijayskumar@bmi.osu.edu), P. Sadayappan

Tahsin Kurc, Joel Saltz
(*www.**cci.emory**.edu*)

Ewa Deelman, Yolanda Gil (*www.**isi**.edu*)