Nimrod/K: Towards Massively Parallel Dynamic Workflows

IEEE SuperComputing 2008, Austin, Texas
November

http://dl.acm.org/citation.cfm?id=1413395&dl=ACM&coll=DL
http://sc08.supercomp.org/scyourway/conference/view/pap134.html

By
David Abramson & Colin Enticott
Faculty of Information Technology
Monash University

Ilkay Altinas
San Diego Supercomputer Center
Grid Computing

Distributed System with non-interactive workloads that involve a large number of files.

Challenges in Grid Computing

To develop a Software that supports:

• Concurre
• Highly Distributed
• Dynamic in nature

Systems have been built using mechanisms like Remote Procedure Calls, IPC, Message Passing or using Web Services.

Programming still remains at low level and is error prone task.
Workflows
A sequence of defined steps and each step achieving some objective.

Recently Workflow Engines have been leveraged as Programming Environment in the Grid.

• Various legacy or existing components are used to build complex applications.
• Can be considered as a Pipeline Architecture.
• The output of one component act as input of another component.
• Workflow Engine takes care of scheduling based on available resources.
Limitation of Existing Workflows

• Most workflows do not allow parallel execution of tasks.
• Workflow Graph structure does not change once the graph has started execution.

Nimrod Family of Tools

• Nimrod/G
  Performs complete parameter sweep over all possible combinations.
• Nimrod/O
  Used for search and optimization using non linear algorithms.
• Nimrod/E
  Used for experimental design and analysis of models.
Focus of the Paper

- Nimrod not designed to execute arbitrary workflows
  - Parameter Sweeps over Workflows
  - Workflows containing Sweeps
- Provide all Nimrod Tools under Kepler Workflow Engine.
- Use Dataflow Execution Model.

Kepler

- Modern Grid Workflow Engine that allows to design and execute scientific workflows.
- Major Components
  - Actors – an encapsulation of actions performed on input data to produce an output data.
  - Models of Computation (MoC) determines the interaction styles of actors.
  - Directors are responsible for implementing particular MoC.

Since Directors orchestrate the workflow the scheduling and overall execution of workflow could be changed by changing the director.
Available Directors

Commonly used directors for Grid programming are:

- **Process Networks**
  - Directed Graph comprising set of nodes comprising a set of directed arcs (representing one way FIFO Channels).
  - Each process executes a stand alone program and is wrapped as an actor.
  - Channels are used to transfer data.

- **Synchronous Data Flow**
  - Sequential execution order of actors are statically determined prior to execution.
  - Specifies number of token generated or consumed during each invocation.
  - Minimal Overhead & bounded memory usage.
  - Guarantees that deadlock will never occur.

Kepler provides Additional functionality to implement execution model.

Tagged Dataflow Architecture is introduced as an execution model.
Dataflow Architecture

- Multiprocessing architecture that supports parallel execution of instruction streams.
- Executes programs as graphs, graph represent machine instruction sequence for a program.
- Instructions fire when it has operands, unlike Von Neumann Model.
Tagged Dataflow Architecture

- Tagged token machine contains both data field and tag field, and tag field determines the separate threads of execution.
- Instruction fires when the machine has token belonging to same tag.
- This allows to change the amount of parallelism dynamically.
Design of Nimrod/K

- New Director using Tagged Dataflow Architecture was implemented for the Execution model.
- Uses the existing PN and SDF directors but tags tokens based on threads of execution.
- Unlike PN, SDF directors maintains separate FIFO queue for each distinct tag.
- Tokens with the same tag queue up and tokens with different tags could be consumed in parallel.
- Special Tag Manipulation Actors are used to assign tag values.
Simple Kepler Workflow

- Parameter sweep actor produces a number of tokens.
- The tokens correspond to one parameter combination.
- My Program accepts one token and generates one token as output.
- Director Orchestration
  - SDF Director executes actors in turn until every actor reports it is unable to execute.
  - PN Director activates every actor simultaneously and blocks all of them at the same time.
The TDA director executing the simple workflow

- TDA director generates one Parameter Sweep Actor but emits tokens with different tags
- Multiple My Program actors are spawned to handle the tokens.
Structure of TDA Director

Main Components:
• Token Scheduler
• Actor Manager
• Director
Token Scheduler

- Responsible for deciding the assignment of token to actor.
- Also looks over the Actor firing order.
- Scheduler is notified of all movement of tokens.
- Responsible for deciding the number of actor copies.
- Ensures all tokens are processed before completion.

Actor Manager

- Responsible for execution of an actor and all the copies of the Actor.
- Each Actor has a separate Actor Manager but the copies does not have separate Actor Manager.
- Maintains the token input and output and initialization and cleaning of Actor
TDA Director

• All Directors in Kepler are invoked in the same manner.
• Maintains the execution of workflow & reports to calling functions.
• The same workflow can have multiple directors.
• Kepler defines rules for which kind of Actors can be used together.
Relevance to Smartgrid

- Agent Based Architecture could be considered as a workflow where each agent has well defined objective.

Multiple Workflows needs to be created for separate Information Sources.
• Each agent can spawn multiple agents to process part of data, similar to the TDA Director.

• Workflow Engine is required to orchestrate the execution of workflows and managing the resources.

• Workflow Engine can provide higher level abstraction.