Fault Tolerance in the Virtual Grid

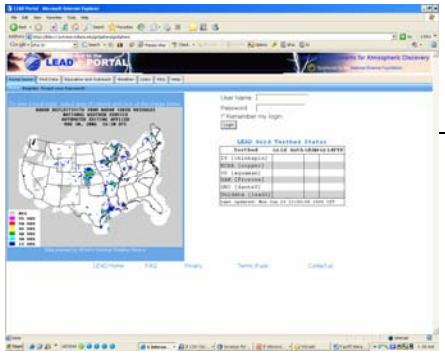
Lavanya Ramakrishnan, Daniel A. Reed

Renaissance Computing Institute

Renci
Renaissance Computing Institute

VGrADS
Virtual Grid Application Development Software Project

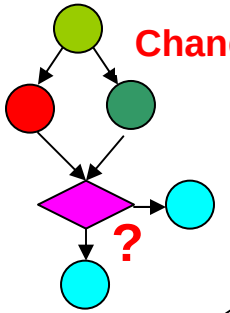
The Need For Fault Tolerance



Launch, configure and control

Application Interface

Change in weather



Change in workflow

Workflow service

Change in service

Grid Portals

App Instance

Open Grid Service Architecture Layer

Registries and Name binding

Reservations And Scheduling

Policy

Event/Message Service

Security

Administration & Monitoring

Logging

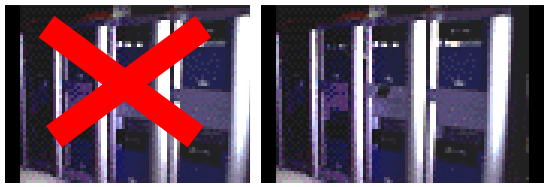
Data Management Service

Accounting Service

Grid Orchestration

Open Grid Service Infrastructure (web service component model)

Resource Layer
(from PCs to Supercomputers)

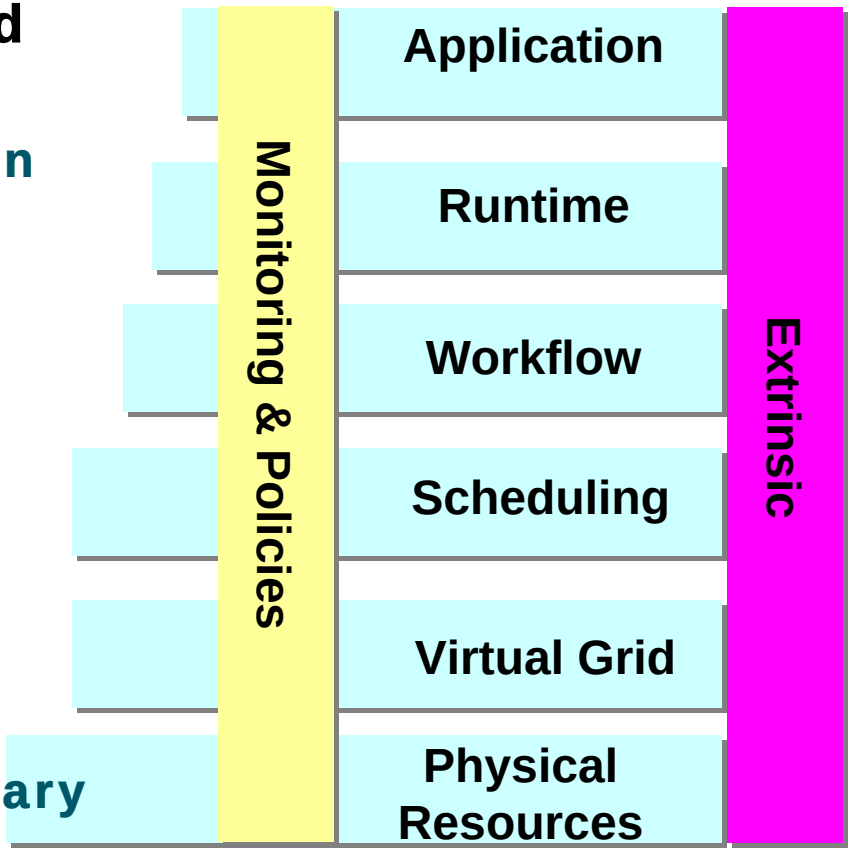


Change in resources (quantity, quality)

Source: Dennis Gannon

Multi-level Fault Tolerance

- **Measurement, monitoring and estimation**
  - **sequential / parallel application**
    - SvPablo, HPCView, ...
  - **distributed workflow**
    - BPEL, ...
  - **system / network**
    - NWS, HAPI, ...
  - **virtual Grid**
    - MORTAR, ...
- **Decision making**
  - **application code, runtime library**
  - **workflow engine, task / job scheduler(s)**
  - **virtual Grid, extrinsic to virtual Grid**



Renci
Renaissance Computing Institute

Example – Master Worker [1/2]

- **Original request for resources for an mpiBLAST run**

  mpiBLAST1 = MasterNode = {memory ≥ 4GB, disk > 20GB }
  highBW LooseBagOf **<WorkerNode> [4:32]; WorkerNode = {memory >= 4GB}**

- **Require the network link between the master and the worker to have "good" reliability.**

  mpiBLAST2 = MasterNode = {memory ≥ 4GB, disk > 20GB }
  goodReliability LooseBagOf <WorkerNode> [4:32]; WorkerNode = {memory >= 4GB}

- **In addition to the network being reliable, the request could specify that the master node be a highly reliable node**

  mpiBLAST3 = HighReliabilityBag<MasterNode>
  goodReliability LooseBagOf <WorkerNode> [4:32]; WorkerNode = {memory >= 4GB}; MasterNode = {memory ≥ 4GB, disk > 20GB}

Renci
Renaissance Computing Institute

Example - SPMD [2/2]

- For a simple WRF run, the request is for a cluster with 8 to 32 nodes, each with at least 4 GB of memory

  wrf1= WRFBag = *TightBagOf*<CNode>[8:32]; CNode = {memory>=2GB}

- Nodes need to be highly reliable and the network between them to be very reliable as well.

  wrf2= WRFBag = HighReliabilityBag<ManyNodes>[1:1]; ManyNodes = *TightBagOf*<CNode>[8:32]; CNode = {memory>=2GB}

- If the model has been running for more than 6 hours, set the expected reliability level to high.

  (reliabilityLevel = high) ← ([runningTime > 6 hours] AND [reliabilityLevel < Good])

Renci
Renaissance Computing Institute

Virtual Grid Interfaces

- **Interfaces to support the ability to**
  - **describe collective qualitative reliability or specific quantitative requirements for resource selection**
    - E.g. need high reliability set of nodes
  - **adjust fault tolerance levels and expectations at run-time,**
    - E.g. watch for reliability fluctuations
  - **register a callback, where application intervention might be required when certain constraints are violated.**
    - E.g. if reliability drops, notify the application

Renci
Renaissance Computing Institute

Some Research Questions

- How does one manage potentially conflicting resource selection goals such as performance and reliability?
- What performability guarantees can be made to applications?
- How can one balance multi-level fault tolerance strategies?
- How can one optimize resource selection based on performance and reliability constraints in the context of a workflow?

Renci
Renaissance Computing Institute

Virtual Grid Description Language

- **vgDL provides application-level resource abstraction**
  - **aggregates or collections**
    - ClusterOf (Homogeneous, Tightly Coupled)
    - TightBag (Heterogeneous, Tightly Coupled)
    - LooseBag (Heterogeneous, Loosely Coupled)
  - **individual resource attributes (extensible)**
    - CPU, Speed, Memory, Disk, Software, Hostname, etc
  - **couplers**
    - HighBW, LowBW, Close, Far
- **Preferences**
  - **scalar ranking function and arithmetic on attributes**
- **Advanced and extent reservation**
  - **start time, duration**
- **Resource quantity (service units)**

Renci
Renaissance Computing Institute

Source: UCSD/ISI

Reliability Extensions to vgDL

- **Reliability Associators (Collective for a set of nodes)**
  - **HighReliabilityBag: (90-100 %)**
  - **GoodReliabilityBag: (80-89%)**
  - **MediumReliabilityBag: (70–79%),**
  - **LowReliabilityBag: (60-69%)**
  - **PoorReliabilityBag: (59-0%)**
- **Reliability Operators (network link)**
  - **highReliability: (90-100%)**
  - **goodReliability: (80-89%)**
  - **mediumReliability: (70-79%)**
  - **lowReliability: (60-69%)**
  - **poorReliability: (59-0%)**

Renci
Renaissance Computing Institute

Compound Operator Expressions

- Request a set of nodes with “highBW and good reliability”
  - Not possible today
- *BNF for vgDL*
  - *Rdl-expression ::= Rdl-subexpression | [“(“ Rdl-expression “)” op “(“ Rdl-expression “)”]**
  - op := close | far | highBW | lowBW
- Alternatives
  - op field needs to support operators such as ***and, or***, etc to connect operators together.
  - develop new operators with compound function
    - **e.g. closeHighReliability, farHighReliability**

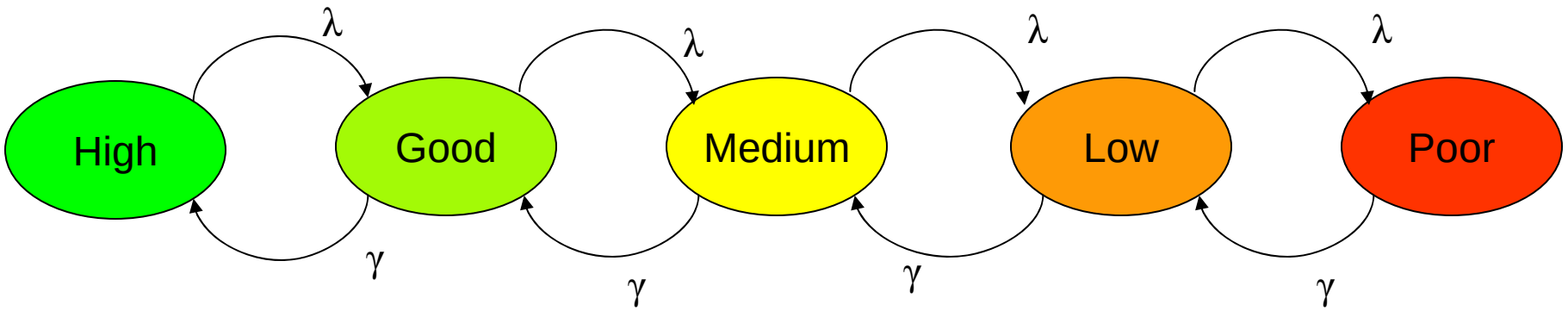
Renci
Renaissance Computing Institute

Performability

- A composite measure for a system's performance and dependability
- Combined Analysis [JMeyers 1980]
  - "the probability that the system reaches an accomplishment level y over a utilization interval (0,t)"
  - Markov reward models to define range

Renci
Renaissance Computing Institute

Performability Model



λ – Failure Rate or reliability degradation rate

γ – Repair Rate for machine

- Assumptions
  - The reliability degradation rate and repair rate are uniform from one state to the other for one single machine
  - A fixed reward for a given state for a given machine

- Accumulated reward in the interval [0,t)

$$Y(t) = \int_0^t Z(\tau)\mathrm{d}\tau$$

t is the time for task from performance model

- Availability = MTTF / (MTTF + MTTR)

Renci
Renaissance Computing Institute

Next Steps

- **Language implementation as extensions to vgDL**
- **Model application to different programming model patterns**
- **Application to LEAD workflows**
  - **Performability as a criteria for resource selection in the workflow planning**
  - **Evaluation of real-time monitoring and adaptation**
    - Using over-provisioning, replication, etc.

Renci
Renaissance Computing Institute