A Distributed Load-Based Failure Recovery Mechanism for Advance Reservation Environments

Lars-Olof Burchard
Barry Linnert
Jörg Schneider

KBS, TU Berlin
Motivation: SLA and Advance Reservations

◆ SLA provisioning for compute jobs and complex workflows in Grid environments
  ■ Resource planning
  ■ Co-allocation
  ■ Deadline guarantees
  ■ Runtime responsibility

◆ Possible solution: Advance Reservations
  ■ Start and stop time given by user
  ■ Needs support by all local RMS
Failure Recovery for Advance Reservations

- Idea: remap all admitted jobs for the broken resource

- How to determine the affected jobs?
  - Estimate failure duration
  - Underestimation: higher termination ratio
  - Overestimation: higher blocking ratio, remapping overhead
Load-Based Remapping

- Remap independently of the failure duration
- Keeping the remapping interval clean

![Diagram showing nodes and time intervals for load-based remapping with key terms: current time, remapping interval, downtime]
Load-Based Remapping

- Remap independently of the failure duration
- Keeping the remapping interval clean

![Diagram showing remapping and downtime intervals]
Load-Based Remapping

- Remap independently of the failure duration
- Keeping the remapping interval clean
Load-Based Remapping

- Remap independently of the failure duration
- Keeping the remapping interval clean
Load-Based Remapping

- Remap independently of the failure duration
- Keeping the remapping interval clean
Load-Based Remapping

- Remap independently of the failure duration
- Keeping the remapping interval clean

![Diagram showing remapping and downtime]

- **Remapping interval**: The period during which jobs are remapped.
- **Downtime**: The period when a node is not available for job execution.
- **Current time**: The time at which the remapping process is being considered.

Example jobs and nodes:
- J2, J6, J7, J8, J11
Load-Based Remapping

- Remap independently of the failure duration
- Keeping the remapping interval clean

Diagram showing nodes and time, with shaded area indicating downtime and remapping interval.
Load-Based Remapping

- Remap independently of the failure duration
- Keeping the remapping interval clean

![Diagram showing load-based remapping with J2, J6, J7, J8, and J11 nodes over time with current time, remapping interval, and downtime highlighted.]
Determine the Remapping Interval

- **Remapping interval**
  - Longer: Probability of success increases
  - Shorter: Less jobs unnecessarily remapped

- **Optimum**: remap, if in the remapping will be impossible in the next time slot

- **Calculate remapping interval using the currently booked load and a prediction off the incoming load**
Load-Based Remapping

- **Remapping Interval:** $i_{\eta}(t_0)$
  - For each $t > i_{\eta}(t_0)$ must be $\hat{l}_{t_0}(t) < \eta$
Distributed Approach

◆ Grid is separated in domains
  ■ No central instance to hold and calculate load and booking profiles
  ■ Avoid communication overhead
  ■ Information hiding between domains

◆ No global knowledge

◆ „How good performs the load based approach with reduced knowledge?“
  ■ Worst case: one resource per domain
  ■ Inhomogeneous load distribution
Evaluation

◆ Measured termination ratio and remapping overhead

◆ Simulated Grid
  ■ Resources: 32, 96, 128, 256, 512 nodes

◆ Reservations
  ■ simple, synthetic jobs
  ■ different book-ahead times / load situations
  ■ Job size: 2, 4, 8, ..., 128 nodes
  ■ Length: 250-750 time slots

◆ Failures
  ■ Constant failure duration (200, 500, 1000 time slots)
  ■ Same failure probability for each resource
Termination Ratio

$\text{load}_{\text{global}} > \text{load}_{\text{local}}$

$\text{load}_{\text{global}} < \text{load}_{\text{local}}$
Remapping Overhead

\[ \text{load}_{\text{global}} > \text{load}_{\text{local}} \]

\[ \text{load}_{\text{global}} < \text{load}_{\text{local}} \]

- Central
- Distributed
**Conclusion:**
- Remapping in advance is needed for systems using advance reservation.
- Load-based approach reach performance achieved with exact knowledge.
- Load-based approach can be used distributed.

**Future Work:**
- Checkpointing and migration to split jobs
- Adapt for other types of resources
A Distributed Load-Based Failure Recovery Mechanism for Advance Reservation Environments

Jörg Schneider, TU Berlin
komm@cs.tu-berlin.de
http://kbs.cs.tu-berlin.de/
Termination Ratio

\[ \text{load}_{\text{global}} > \text{load}_{\text{local}} \quad \text{load}_{\text{global}} < \text{load}_{\text{local}} \]

\[
\begin{array}{cccccccc}
0.10 & 0.25 & 0.40 & 0.55 & 0.70 & 0.85 & 1.00 \\
50 & 60 & 70 & 80 & 90 & 100 & 100
\end{array}
\]

\[
\eta = \text{termination ratio (\%)}
\]

- \text{central}
- \text{distributed}
- \text{exact knowledge}
- \text{50\% underestimation}
Remapping Overhead

![Graphs showing remapping overhead](image)

- **x-axis**: \( \eta \) (eta)
- **y-axis**: overhead (%)

**Legend**:
- **central**
- **distributed**
Different Load Differences

The graphs illustrate the termination ratio (%) for central and distributed load differences as a function of efficiency (η) and load difference. The graphs show the relationship between these variables for different load scenarios, with color intensity indicating the termination ratio. The central and distributed load differences are plotted on separate graphs, each with a range of load differences from -650 to 0 and efficiency values from 0.1 to 1.0.
Load-Based Remapping

◆ **Load Profile:** \( l_{t_0}(t) \)
  - For the time slot \( t_0 \): The **whole** booked load of each future time slot \( t_0 + t \)

◆ **Booking Profile:** \( b_{t_x}(t) \)
  - The load booked during time slot \( t_x \) for the future time slot \( t_x + t \)

◆ **Average Booking Profile:** \( \bar{b}(t) \)

◆ **Combined Profile:** \( \hat{l}_{t_0}(t) = l_{t_0}(t) + \bar{b}(t) \)
  - Expected load for future time slots

◆ **Load Threshold:** \( \eta \)
  - Maximum load for a high remapping probability.
Virtual Resource Manager: Main Idea

- Joint project of KBS (TU Berlin) and PC² (Paderborn University)

- Layer between “Grid” and local RMS (“close the gap”)

- Implement SLA negotiation, monitoring, allocation, planning within this layer

- Few modules with well-defined standard interfaces (-> GGF)

- Management entity providing enhanced functionality for domains of various size

- Integration of existing RMS
VRM Architecture

„The Grid“ (other VRMs, Resources, ...)

VRM

ADC

AI

RMS (Cluster)

RMS (Network)

RMS (Cluster)
Application Environment

◆ Advance Reservations
  - Time slots of fixed size
  - Book-ahead interval

◆ Reservations
  - Booked in advance
  - Start and stop time given at admission