Reliable Workflow Execution in Distributed Systems for Cost Efficiency

Young Choon Lee[1], Albert Y. Zomaya[1] and Mazin Yousif[2]

[1]Centre for Distributed and High Performance Computing
School of Information Technologies
The University of Sydney

[2]Global Technology Services
IBM Canada

FACULTY OF ENGINEERING & INFORMATION TECHNOLOGIES

The Joys of Real Hardware

- Typical first year for a new cluster

~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)

~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)

~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)

~1 network rewiring (rolling ~5% of machines down over 2-day span)

~20 rack failures (40-80 machines instantly disappear, 1-6 hours to come back)

~5 racks go wonky (40-80 machines see 50% packetloss)

~1000 individual machine failures

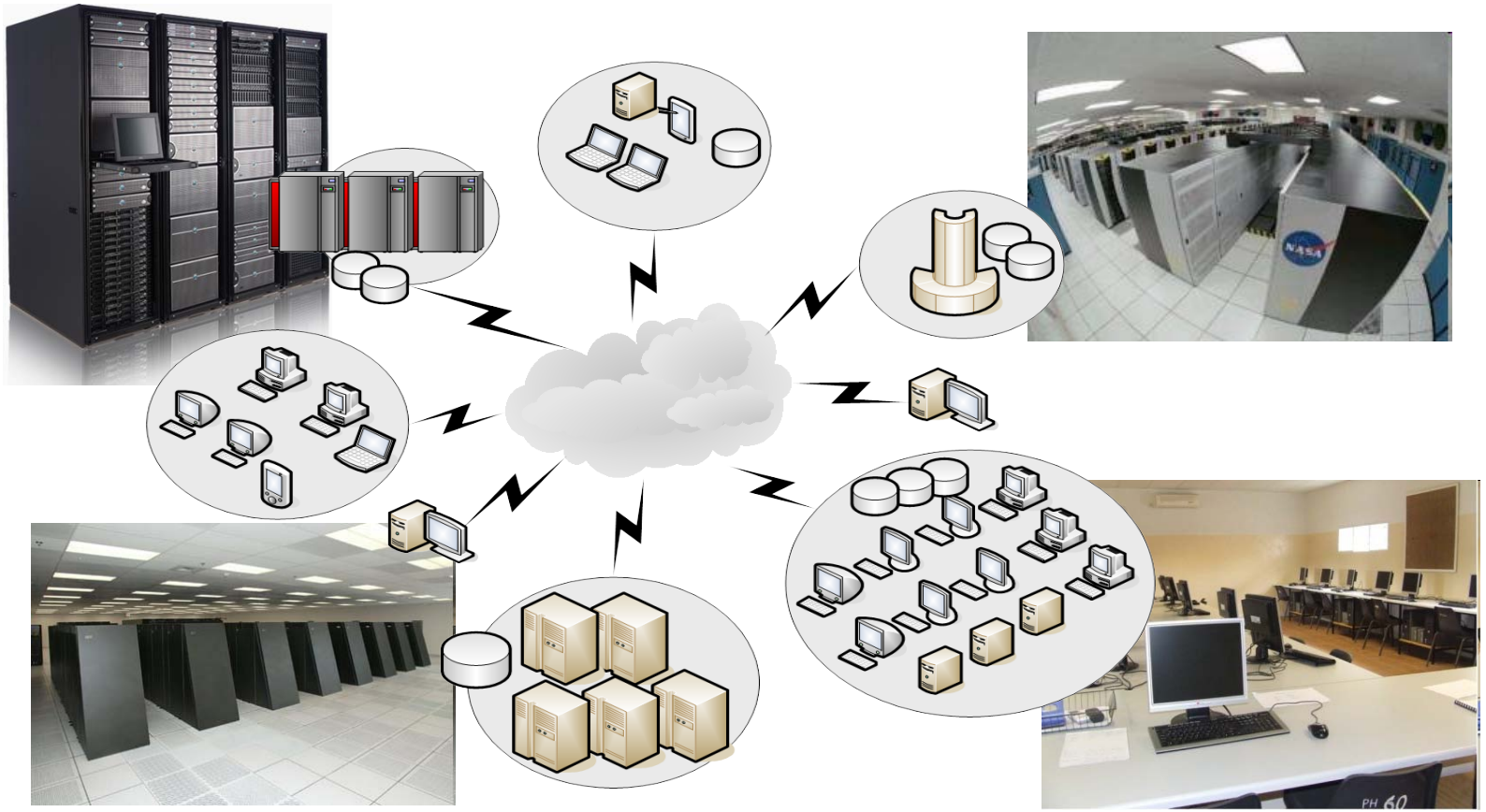
~thousands of hard drive failures

And much more

*Source :Designs, Lessons and Advice from Building Large Distributed Systems by **Jeff Dean**, Google Fellow at LADIS 2009*

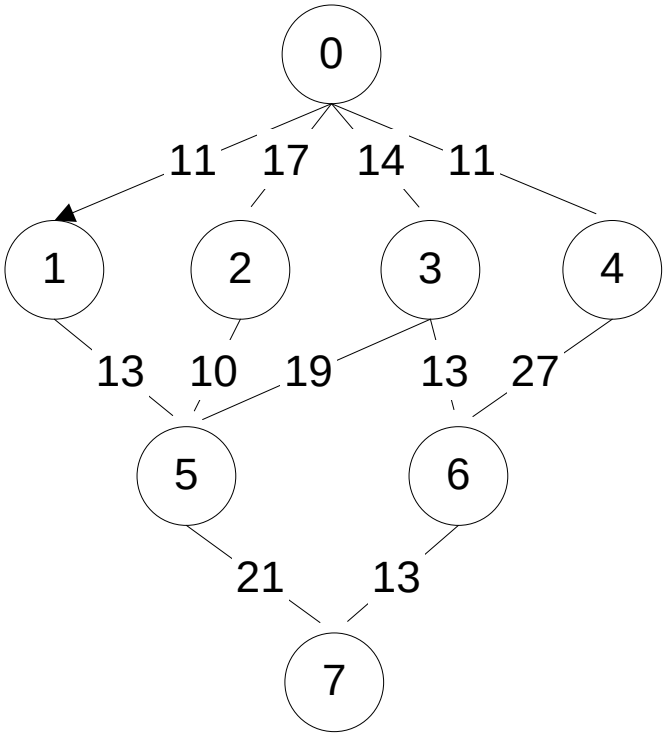
Large-Scale Distributed Systems (LSDSs)

- Multi-tenant LSDSs like grids and clouds may exhibit various unreliability factors and more complicating effects



Workflow Applications

› A DAG, *G* = *(N, E)*, consists of a set *N* of *n* nodes, and a set *E* of *e* edges



- *Task 0: Entry task*
- *Task 7: Exit task*

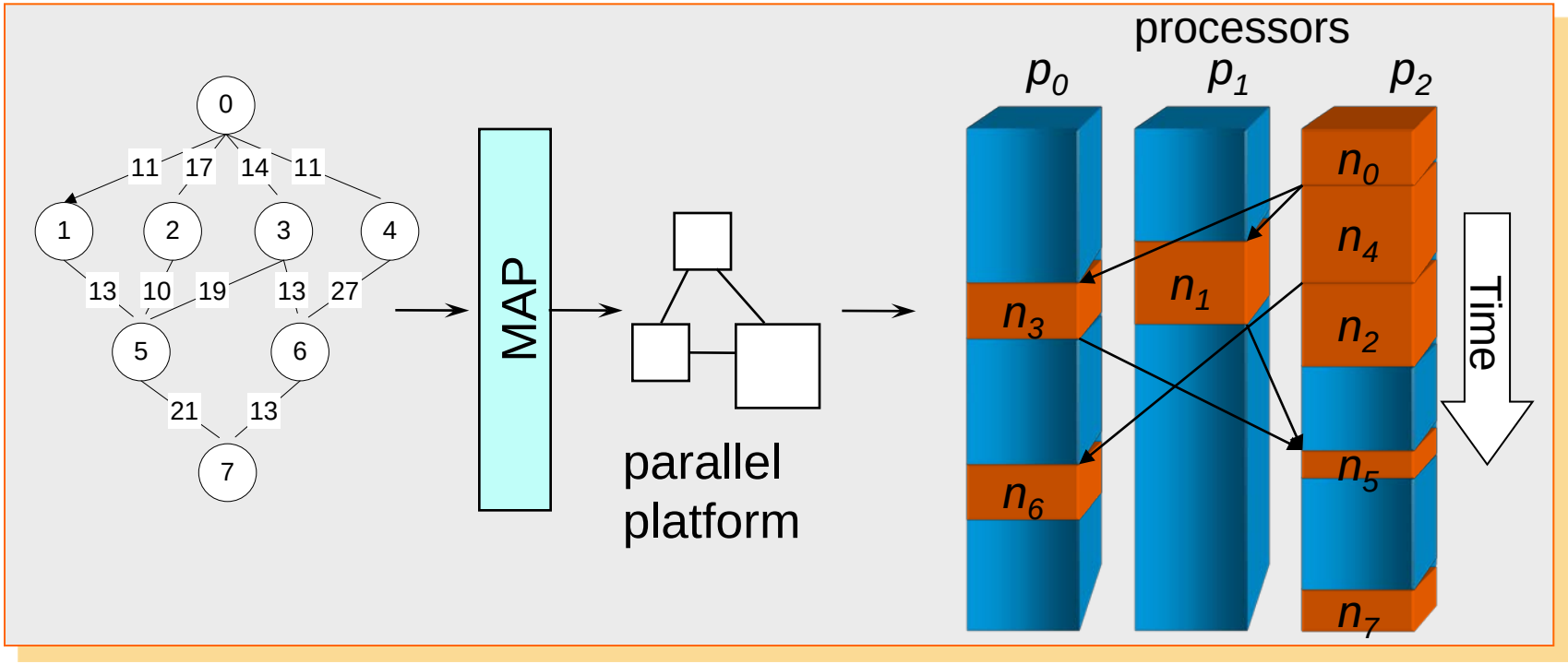
Node/Task

Edge/Interprocessor communication/ Precedence constraint

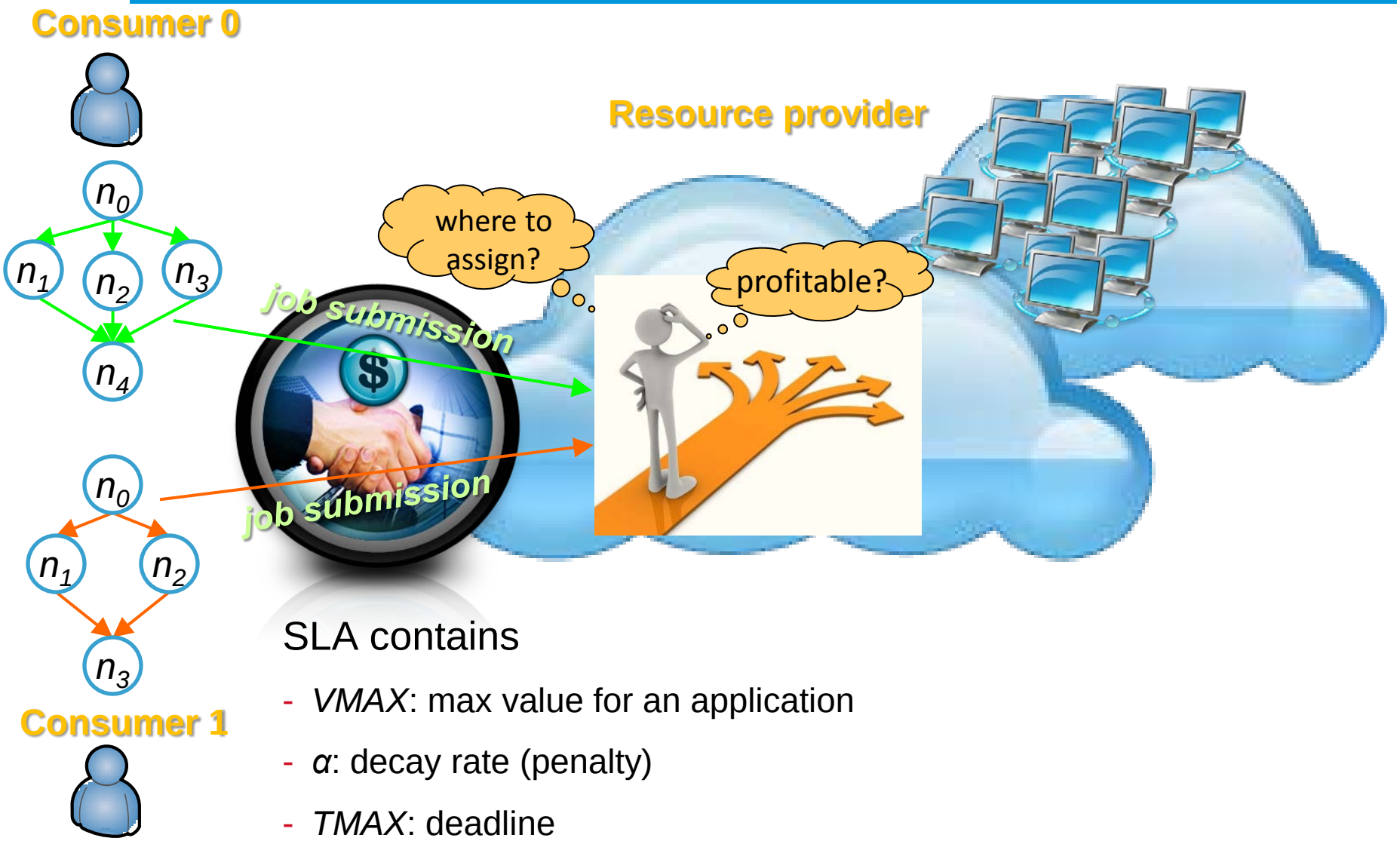
n — Interprocessor communication cost

Workflow Scheduling

› How to assign interdependent tasks to computing resources (processors) aiming to minimize makespan without violating precedence constraints



Workflow Scheduling in Market-based LSDSs



SLA contains

- *VMAX*: max value for an application
- *α*: decay rate (penalty)
- *TMAX*: deadline

Market-based Distributed Computing Systems

- Advertised cost (*ac*) per unit time per resource occupancy is charged to the resource consumer:

$$ac = oc \cdot (1 + pr^{net})$$

where *oc* is the net operating cost and pr^{net} is the net profit rate set by a provider.

- Resource occupancy: the resource is exclusively allocated for a particular request.
- The amount of a resource occupancy includes:
  - actual task execution time
  - idle time occurring due to inaccurate execution time estimate, and
  - waiting time caused by delays (from resource failures) in the completion of predecessor tasks

SLA parameters

- Key SLA parameters of an application are:
  - *VMAX*: maximum value

$$VMIN = \sum_{i=1}^{n} w_i \cdot ac$$

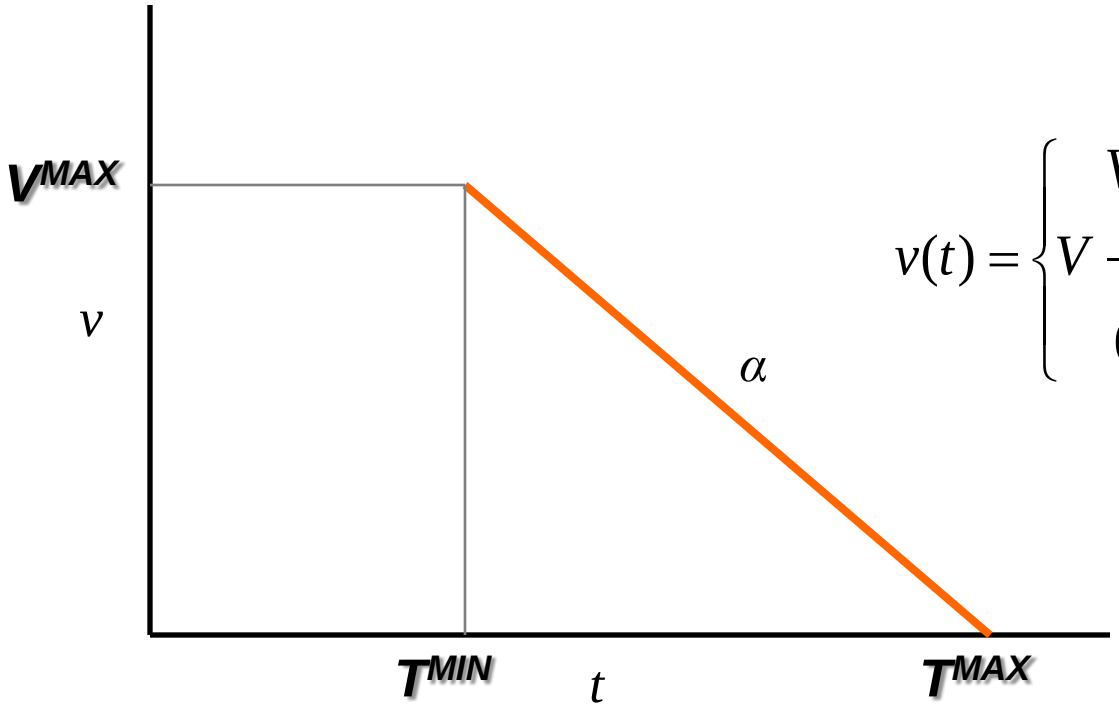
$$VMAX = VMIN + \Delta v$$

  - α: value decay rate

$$\alpha = \frac{VMAX}{(TMAX - TMIN)}$$

Utility Function

› Value (profit) ***v*** is inversely related to processing time ***t***



$$v(t) = \begin{cases} V & , \quad t \le TMIN \\ V - \alpha t & , \quad TMIN \le t \le TMAX \\ 0 & , \quad t > TMAX \end{cases}$$

Reliable Workflow Scheduling in Market-based LSDSs

- A resource provider attempts to guarantee the quality of service to a certain degree in the way its profit is maximized.
- In this scenario, resource failures should be explicitly and proactively addressed, or QoS is much compromised and profit is reduced accordingly.
- Those reserved resources can be seen as a result of resource co-allocation using advance reservation at the arrival of the workflow in grids, or that of resource provisioning for the next time frame (epoch) at the finalization of a given bidding event in clouds.
- We investigate the reliability of workflow execution in the context of scheduling and its effect on operating costs in LSDSs
- The main objective is the maximization of profitability reducing costs incurred specifically by resource failures.

Reliability for Profit Assurance (RPA)

- The rationale behind our *RPA* algorithm is that the cost efficiency in workflow scheduling is heavily dependent on the reliable execution of high impact tasks (normally, small in number) who have a large number of successor tasks.
- Task replication is adopted to deal with resource failures.
- However, replicas created in response for increasing reliability are clearly a major source of resource wastage.
- Replication is guided by “replicability parameters”: loss, actual loss and replication cost.

Reliability for Profit Assurance (RPA)

Algorithm 1: *RPA*

1: Let $Q = \emptyset$
2: **while** $\exists n_i \in N \mid n_i$ is not scheduled **do**
3: Let n^* = the first ready task in N to be scheduled
4: Let p^* = processor in P on which n^*'s *EFT* is the smallest
5: Reserve p^* for n^*
6: Enqueue n^* to Q
7: **end while**
8: Let *ms* = makespan based on the current schedule
9: **while** $Q \neq \emptyset$ **do**
10: Let n_i **=** the first task in Q
11: Dequeue Q
12: Let cd_i = compute_cumulative_delay(n_i, $w_i/2$)
13: Compute mi_i based on *ms* *// makespan increase*
14: Let *loss* = $VMAX - (mi_i \cdot \alpha + cd_i \cdot oc)$
15: Let *actLoss* = $loss \cdot ur(w_{i,*})$
16: Let p_j = processor in P on which n_i's *EFT* is the smallest
17: Let *repCost* = $w_{i,j} \cdot oc$
18: **if** actLoss > *repCost* **then**
19: Let n_i' = a replica of n_i
20: Assign n_i' to p_j
21: **end if**
22: **end while**

Initial scheduling

Task replication

Reliability for Profit Assurance (RPA)

Function: compute_cumulative_delay

Input: a task $n_i \in N$ and the estimated delay ed_i

1: **for** $\forall n_{i,j} \in succ(n_i)$ **do**

2: Let $w_{i,j} = EFT(n_{i,j}) - EST(n_{i,j})$

3: Update $EFT(n_{i,j})$ considering pd_i

4: **if** ($EFT(n_i) > EST(n_{i,j})$) **then**

5: Let $EST(n_{i,j}) = EFT(n_i)$

6: **end if**

7: Let $d_{i,j} = EFT(n_{i,j}) - EST(n_{i,j}) - w_{i,j}$

8: Let $cd_i = cd_i + d_{i,j}$ *// recursive function call*

9: Let $cd_i = cd_i +$ **compute_cumulative_delay**$(n_{i,j}, ed_i)$

10: end for

11: Return cd_i

Performance Evaluation

› The total number of experiments conducted is 1,012,500 (337,500 for each *algorithm—Dynamic*, *Reactive* and ***RPA***)

	Parameter	Value
Global	# different workflow applications	500
	mean for workflow inter-arrival times	U(10, 100)
	simulation duration	{2,000, 4,000, 8,000, 16,000, 32,000}
	operating cost (*oc*)	10
	net profit rate (pr^{net})	{20%, 40%, 60%, 80%, 100%}
	reservation delay (*rd*)	U(1, 64)
Application	# tasks per workflow (*n*)	U(4, 128)
	maximum width	{2, 4, 8, 16, 32}
	out degree of a node	U(0, 8)
	comm. to comp. ratio (CCR)	≤ 0.2
Resource	# resources	U(4, 64)
	failure rate	{0.1%, 0.5%, 2.0%}
	heterogeneity	{100%, 200%, 400%}
	mean time to repair (MTTR)	U(10, 100)

Performance metrics

- Profit rate

$$pr_i = \frac{v(t)_i - (oc\sum_{j=1}^{n} w_{j,*} + pd_j)}{VMAX_i - (oc\sum_{j=1}^{n} w_{j,*})} \qquad \overline{pr} = \frac{\sum_{i=1}^{M} pr_i}{M}$$

- Response rate

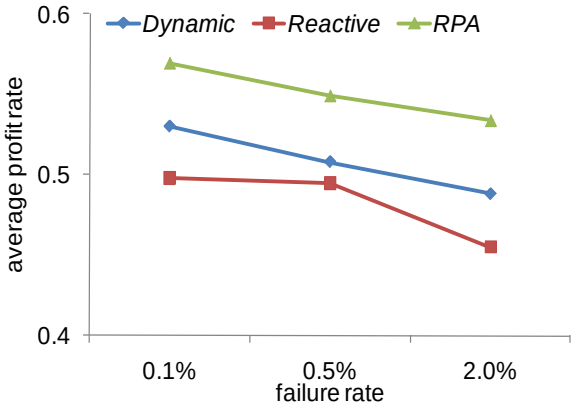
$$rr_i = \frac{TMIN_i}{t_i} \qquad \overline{rr} = \frac{\sum_{i=1}^{M} rr_i}{M}$$

where *M* is the total number of workflow applications processed

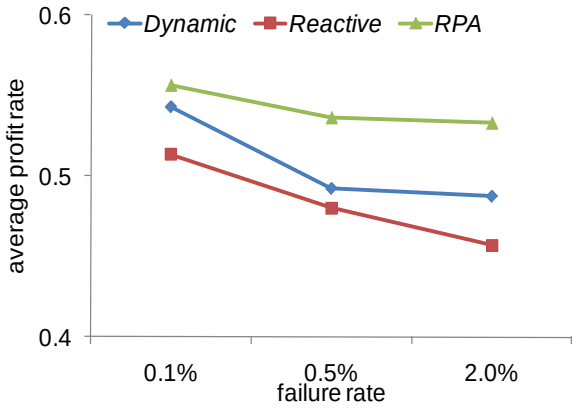
Results: profit rate

- Average profit rates
  - ***RPA* : 55%**
  - Dynamic: 49%
  - Reactive: 48%

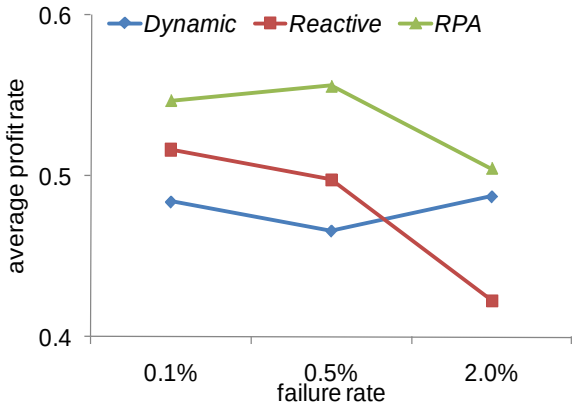
- Different resource heterogeneity values



(a) 100%



(b) 200%

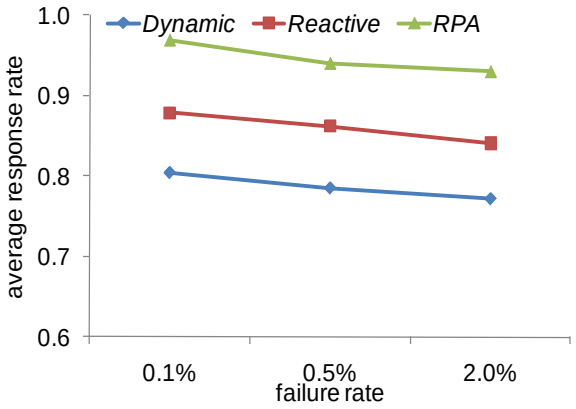


(c) 400%

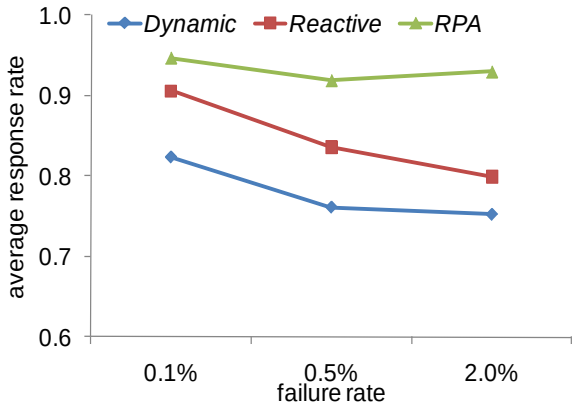
Results: response rate

- Average response rates
  - ***RPA* : 93%**
  - Dynamic: 77%
  - Reactive: 85%

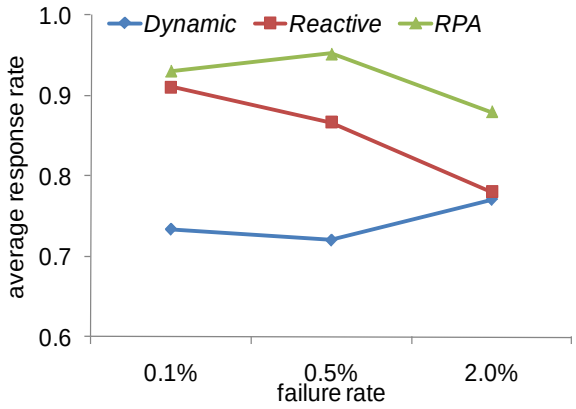
- Different resource heterogeneity values



(a) 100%



(b) 200%



(c) 400%

Related Work

- Market-based resource allocation
  - Pricing-based methods can reveal the true needs of users and allocate resources efficiently [Sigecom Ex. 05]

- Time-varying resource valuation
  - Changing values for prioritizing & scheduling jobs
  - Penalty when the value is not realized [CCGrid 02, HPDC 04]
  - In our model, penalty is implicitly reflected in the cost of physical resource usage
  - Our utility function is time-varying and dependency aware

Conclusion

- A resource failure causes delay not only in the completion of the task running on it, but also in the completion of that task's successors and eventually the makespan of corresponding workflow recursively.
- This recursive impact of resource unreliability on workflow execution is a serious limiting factor in cost efficiency in LSDSs.
- Replication is an appealing solution for reliable workflow execution.
- The cost-aware replication incorporated into RPA is an intuitive cost optimization technique for workflow execution in LSDSs.