Fault-Tolerant Scheduling with Dynamic Number of Replicas in Heterogeneous Systems

Laiping Zhao[1], Yizhi Ren[1,3], Yang Xiang[2], Kouichi Sakurai[1]

[1] Department of informatics, Kyushu University, Japan,
[2] School of Information Technology, Deakin University, Australia.
[3] School of Software, Dalian University of Technology, China.

This paper is partly Supported by the Grant of Graduate School of ISEE, Kyushu University for Supporting Students' Overseas Traveling.
The first author of this research is supported by the governmental scholarship from China Scholarship Council.

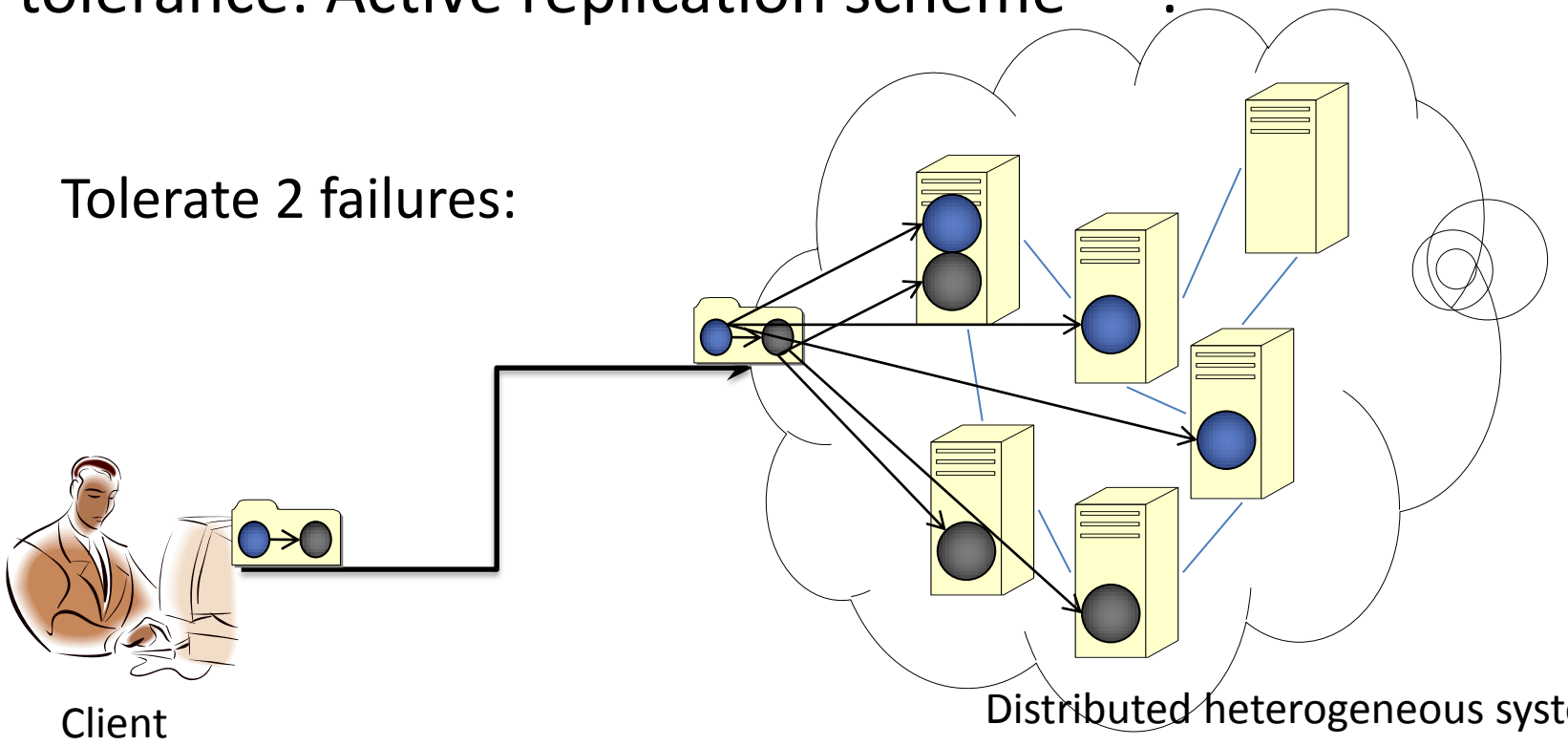
Outline

- 1. Introduction
  - Active replication
  - Problem
- 2. System model
- 3. The MaxRe scheduling algorithm
- 4. Experiments
- 5. Conclusion and future works

1. Introduction

Fault tolerance: Active replication scheme[1-2]:

Tolerate 2 failures:



The job will be completed even with maximum 2 failures occur.

[1] A. Benoit, et al. Parallel Computing, 2009.
[2] A. Girault et al. DSN2003.

1. Introduction

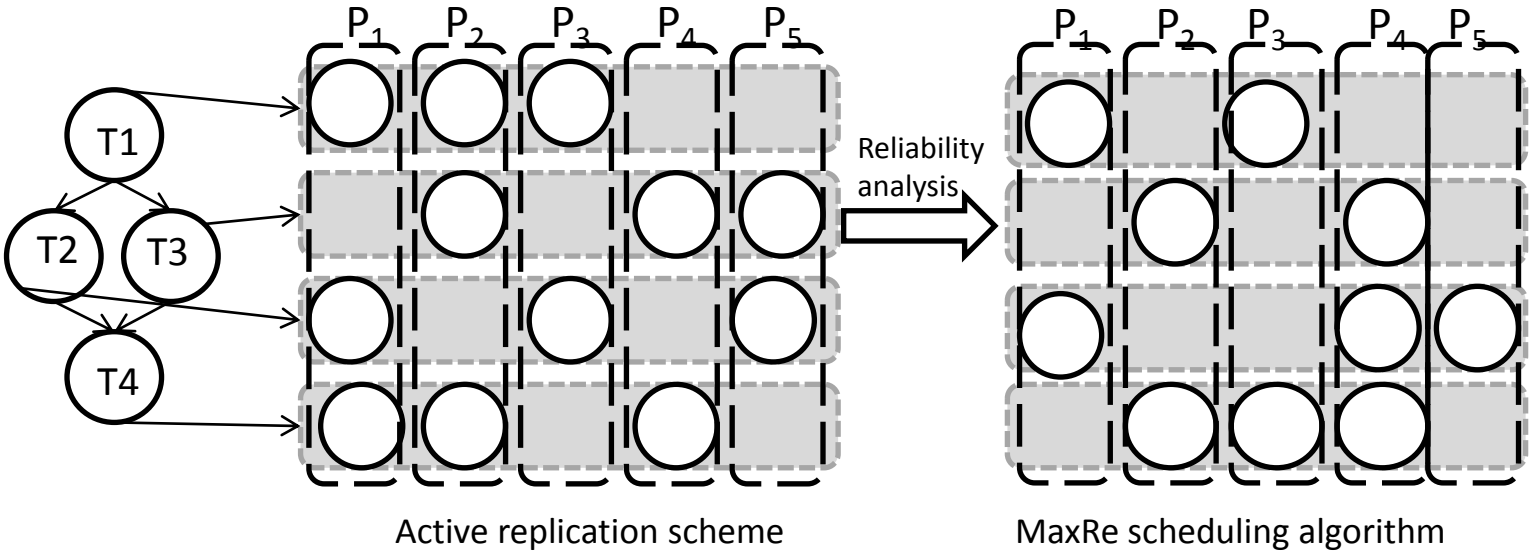
- In order to tolerate *t* failures, the active replica scheme has to schedule *t+1* processors for each task.
- The large resource redundancy problem:
  - Energy consumption
    - Electric power, etc.
    - According to EPA (Environmental Protection Agency) report 2007, Servers and data centers consumed 1.5 percent of total U.S. electricity consumption in 2006.
  - Economic cost
    - To store x gigabytes data, it has to use 3x gigabytes storage space. Assuming the economic cost of each drive is y, the extra 2y cost is believed to be passed on to the customers eventually.
    - E.g. Facebook utilizes 4 replicas, and Google utilizes 3 replicas for each data volume.
- The problem is how to achieve the corresponding reliability with less resources (replicas).

1. Introduction

Our basic idea:

Combining the processors' reliability analysis within the replica scheduling, and make use of less resources to achieve the 2-failures tolerating comparable reliability.

Tolerate 2 failures:



2. System Model

- Processor model: $P = \{p_0, p_1, p_2, \ldots, p_{m-1}\}$
  - The processor is fault free during its idle time;
  - The arrival of failures follows a Poisson distribution:

$$f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

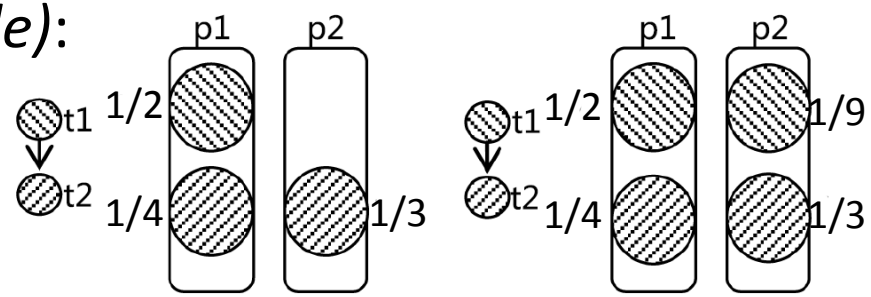
- Job model:
  - A job is represented as a weighted directed acyclic graph (DAG): $G = (V, E)$
  - $n = |V|$ is the number of tasks.
  - $e = |E|$ is the number of edges.

2. System Model

Proposition 1:

When submitting a DAG-based workflow to a heterogeneous system, the more replicas for each task may not lead to a higher reliability.

Proof (one example):



(a) (b)

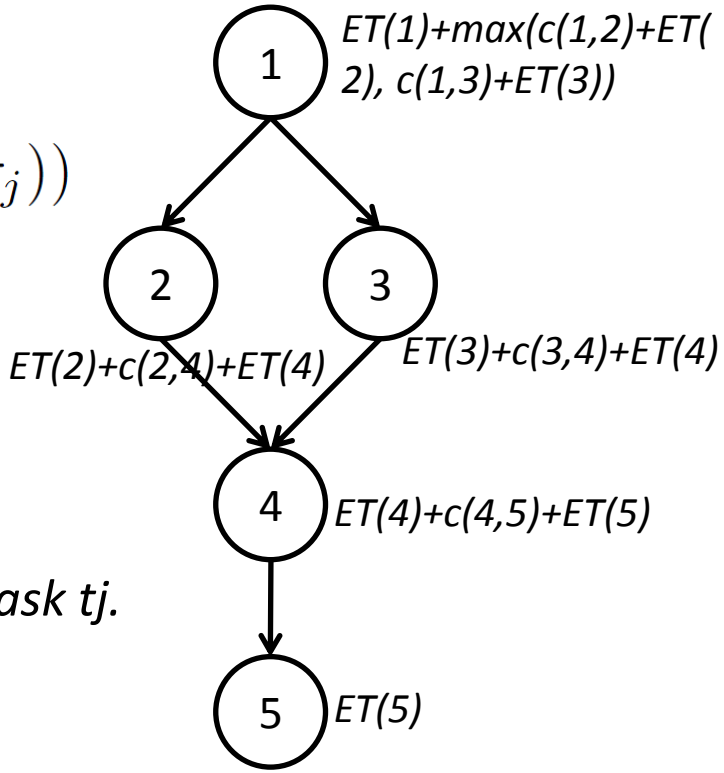
(a): $\frac{1}{2} \times \left(1 - \left(1 - \frac{1}{4}\right) \times \left(1 - \frac{1}{3}\right)\right) = 0.25$

(b): $1 - \left(1 - \frac{1}{2} \times \frac{1}{4}\right) \times \left(1 - \frac{1}{9} \times \frac{1}{3}\right) \approx 0.1574$

3. MaxRe scheduling algorithm

- Task priority[1]:

$$rank(\tau_i) = \overline{w_i} + \max_{\tau_j \in succ(\tau_i)} (\overline{c_{i,j}} + rank(\tau_j))$$



Where:

rank(t): priority value of task t;

$\overline{w_i}$: *the average computation cost of task t;*

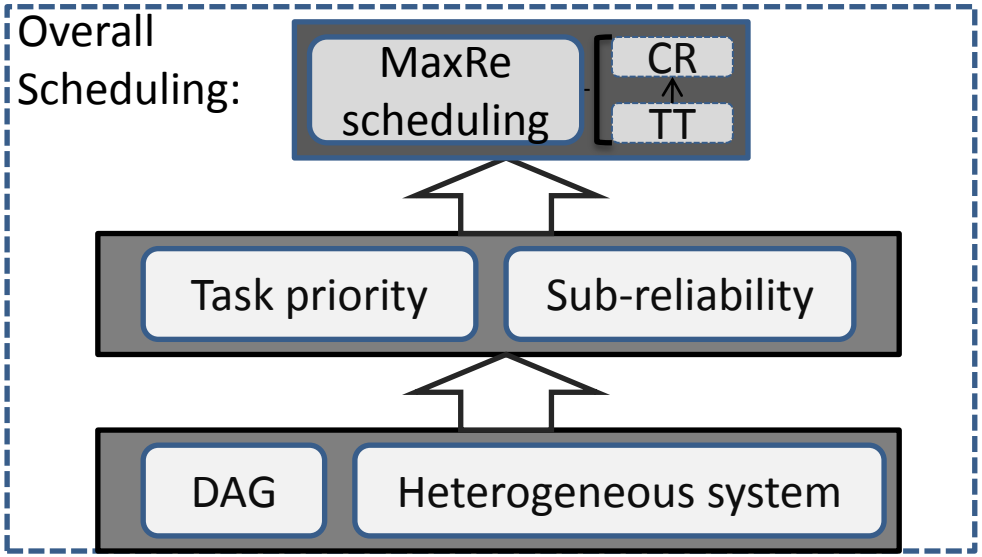
succ(t): the successor tasks of task t;

$\overline{c_{i,j}}$: *average communication cost from task ti to task tj.*

[1] H. Topcuoglu, et al. IEEE Trans. on Parallel and Distributed System,2002.

3. MaxRe scheduling algorithm

- For each task's sub-reliability: $r = \sqrt[n]{R}$
  - Where *r* is the sub-reliability for each task;
  - *n* is the number of tasks;
  - *R* the the overall reliability requirement.
- MaxRe:



3. MaxRe scheduling algorithm

Definition TT (Total Time):

$$TT(p_j) = ET(\tau_i, p_j) + \sum_{\tau_k \in on(p_j)} ET(\tau_k, p_j)$$

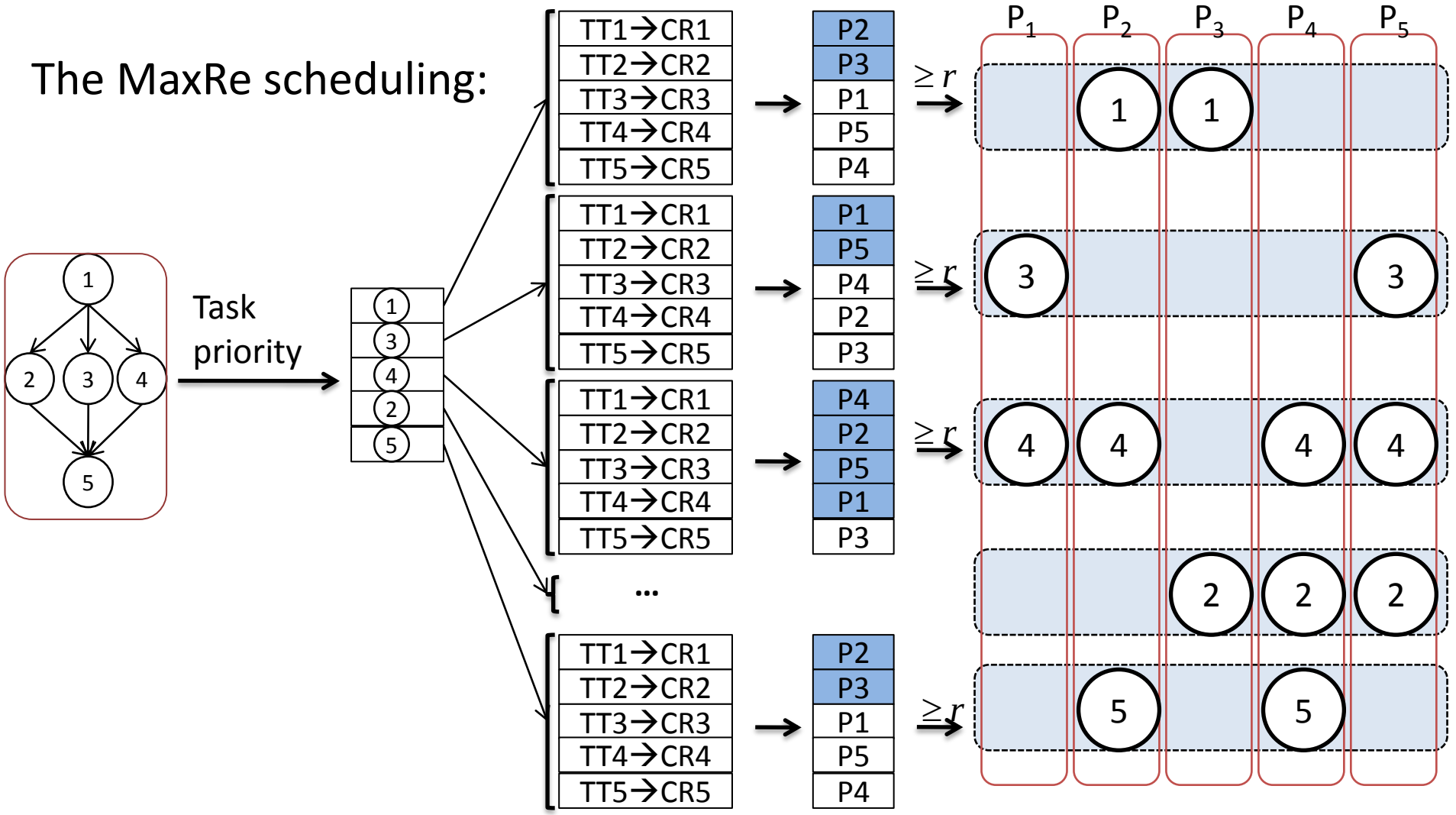
where $ET(\tau_i, p_j)$ *is the execution time when scheduling task* τ_i *to processor* p_j. $on(p_j)$ *is the previous tasks that have already been scheduled on processor* p_j.

Definition CR (Current Reliability):

$$CR(p_j) = e^{-\lambda_j TT[p_j]}$$

3. MaxRe scheduling algorithm

The MaxRe scheduling:



3. MaxRe scheduling algorithm

- Analysis:

Suppose the reliability value provided by the MaxRe algorithm is R(MaxRe). When the required number of replicas for each task does not exceed the total number of processors, we have: R <= R(MaxRe).

Proof (key point):

$$R = r^n <= \prod_{i=0}^{n-1} 1 - \prod_{k=0}^{k<\xi} (1 - CR(\tau_i, p_k))$$

$CR(\tau_i, p_k)$ implies the probability that, for task i, the replica p_k does not encounter failures from the very beginning to the i's finish.

3. MaxRe scheduling algorithm

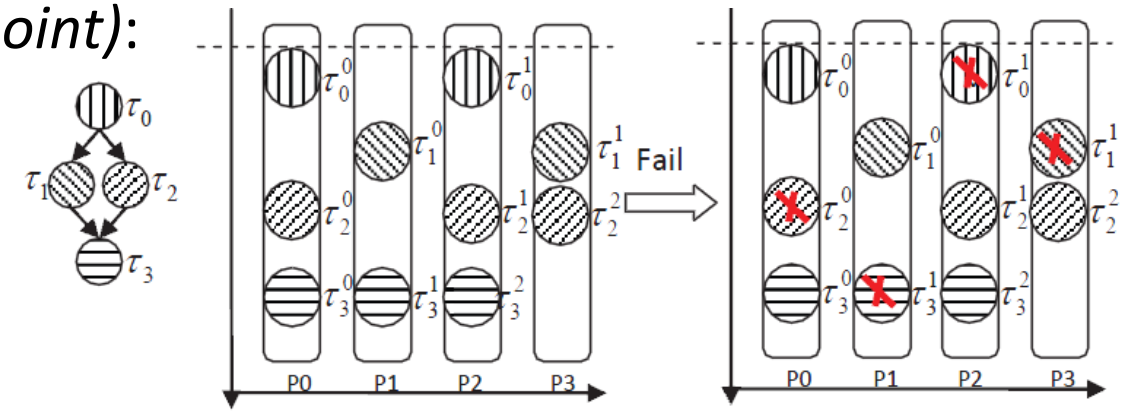
- The reliability of the MaxRe's result:

Proposition 3:

The reliability R(MaxRe) provided by the MaxRe scheduling algorithm meets the following conditions:

$$R <= R(MaxRe) <= \prod_{i=1}^{i<n} (1 - \prod_{p_j \in sche(\tau_i)} (1 - R(\tau_i, p_j)))$$

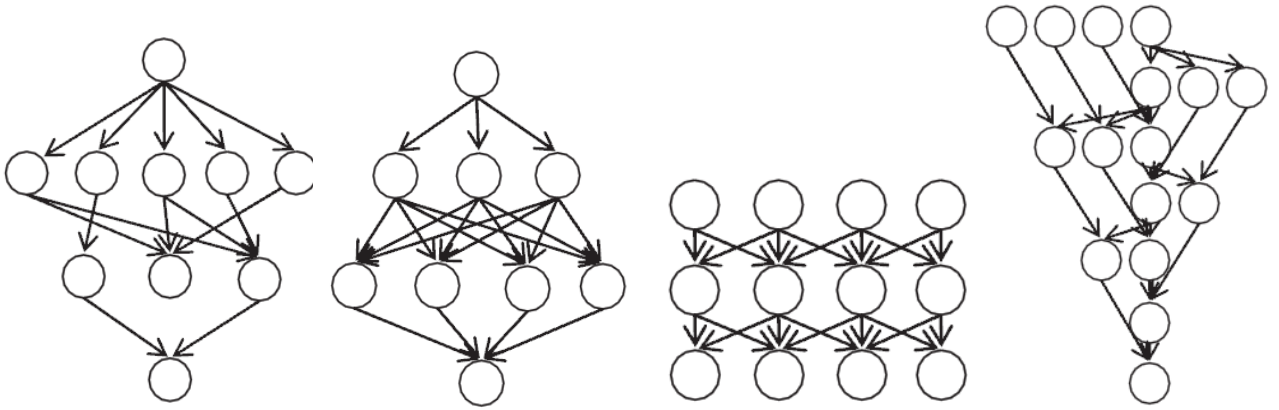
Proof (key point):



$R(\tau_i, p_j)$ implies the probability that no failures occur during the time period that task T_i is executed on processor p_j.

4. Experiments Results

- The 4 workflow examples:



(a) Sample 1 (b) LQCD (c) Stencil (d) Doolittle

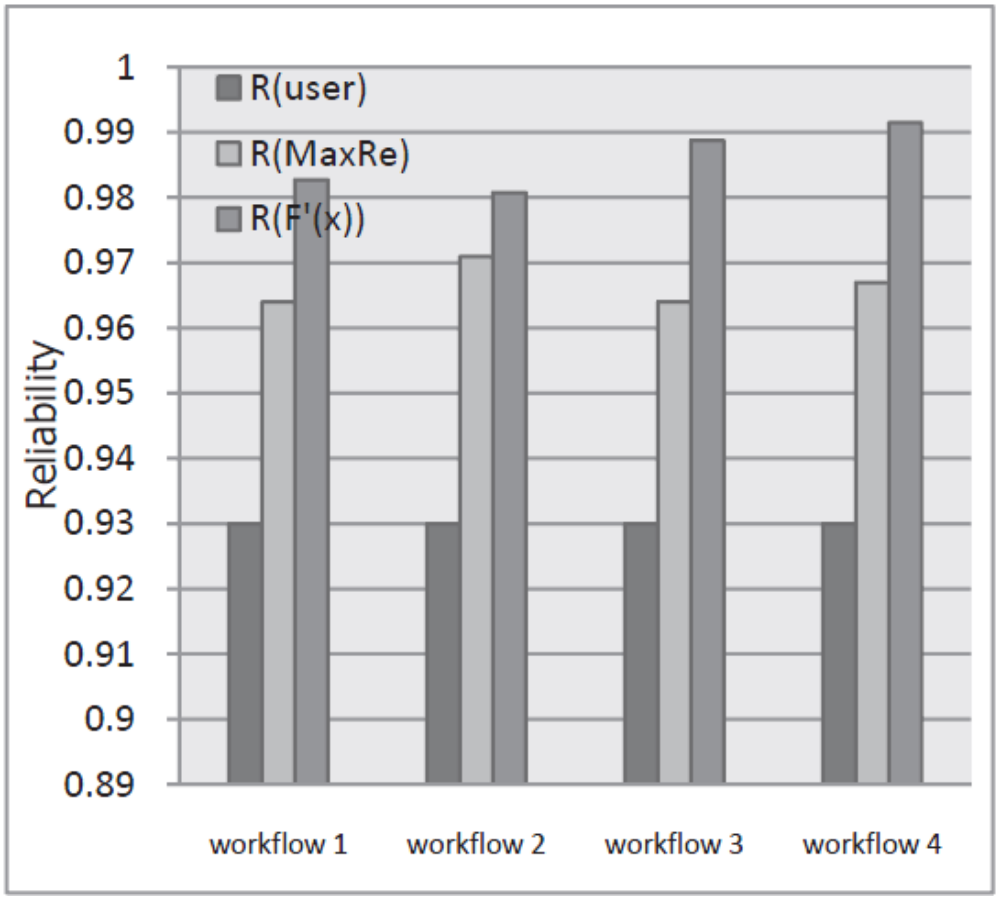
- The Parameters for the simulation platform

Task			Processor		
Load	Com_load	No.	Speed	$\lambda \times 10^3$	Com_speed
$100 \sim 500$	$9 \sim 29$	$10/20$	$5 \sim 19$	$2 \sim 8$	$0.8 \sim 1.2$

[1] H. Topcuoglu, et al. IEEE Trans. on Parallel and Distributed System,2002.

4. Experiment Results

- (1) The verification to Proposition 3:



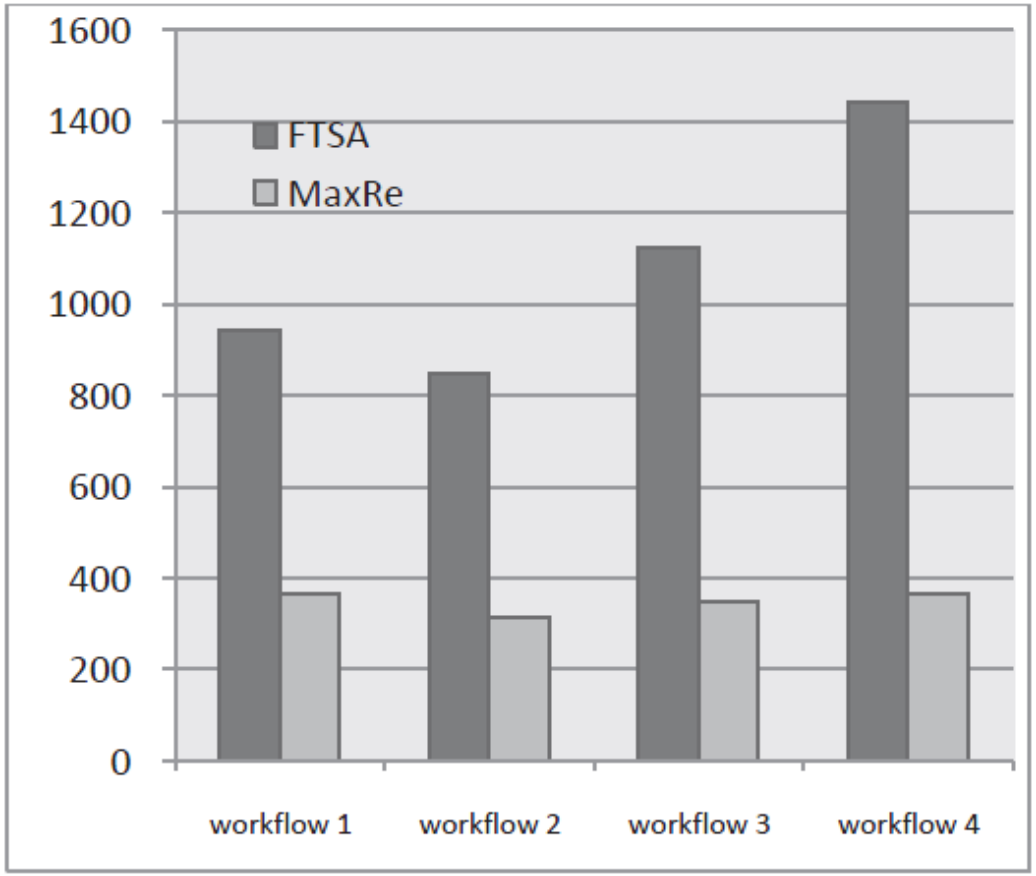
(1)R = 0.93.

(2) R(MaxRe) > 0.96;

(3) $R(\prod_{i=1}^{i<n} F'(\tau_i)) > 0.98$

4. Experiment Results

- (2) The resource usage with comparison to the FTSA algorithm[1]



Resource usage metric:

$$CUA = \sum_{i=0}^{i<n} \sum_{j=0}^{j<m} (ET(\tau_i, p_j) \times x_{ij})$$

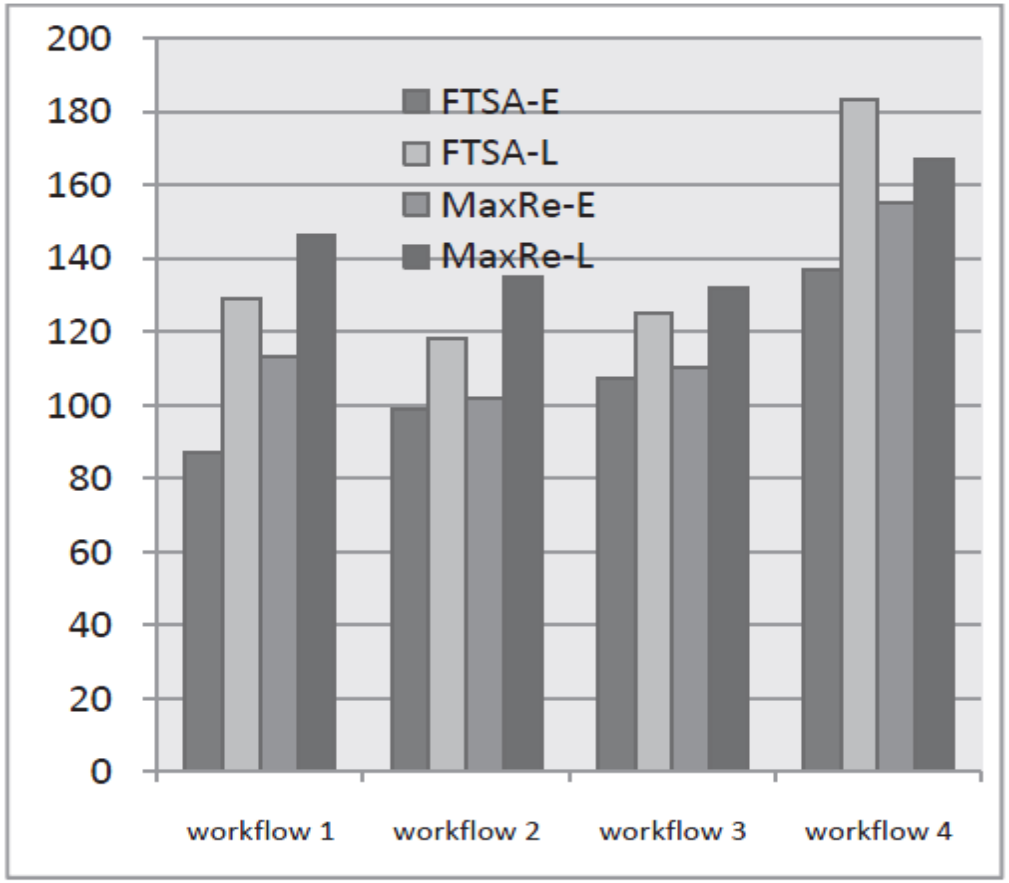
(1)The MaxRe consumes at most 70% fewer resources than the FTSA algorithm.

(2)To tolerant more failures, FTSA algorithm needs more resources than the MaxRe algorithm.

[1] A. Benoit et al, APDCM'2008.

4. Experiment Results

- (3) The execution time with comparison to the FTSA algorithm



The earliest finish time: The earliest finish time of MaxRe's schedule is at most 20 percent longer than FTSA.

The latest finish time: The MaxRe scheduling algorithm even performs better than the FTSA Algorithm.

5. Conclusions and future works

Contributions:

We propose the MaxRe scheduling algorithm which satisfy the user's reliability requirement with the minimum resources:

- (1) The theoretical analysis and experiments prove that MaxRe algorithm can satisfy the user's reliability requirement.
- (2) The resource usage by MaxRe algorithm is *at least 30 percent* fewer than by FTSA algorithm (at most 70%).
- (3) The earliest finish time of MaxRe's schedule is *at most 20 percent* longer than FTSA, while the latest finish time of MaxRe's schedule is even shorter than FTSA.

5. Conclusion and future works

- The MaxRe algorithm is suitable when:
  - The user has special requirements on higher reliability, however, the resources are limited.
- Future works:
  - The consideration on the output schedule's execution time;
  - The system performance analysis.

- Thank you for your attention,
- Questions?