Scheduling for workflows with security sensitive intermediate data by selective Tasks duplication in clouds

Abstract:

Cloud computing has become a revolutionary paradigm by enabling on-demand provisioning of applications, platforms, or computing resources for customers based on a "pay-as-you-go" utility computing model. In this model, cloud service providers manage large-scale heterogeneous virtual machines (VMs) to process customers' applications.

Existing system:

Qiu et al. proposed an ILP-based method to choose cryptographic instances for the data generated by executing workflows and meet their schedulability. However, the existing security approaches do not make full use of idle time slots to duplicate workflow tasks so as to advance the start time of workflow tasks and workflows' make spans. In addition, these approaches rarely exploit tasks' laxity time to encrypt their output data which makes the encryption time seriously delay workflows' make spans.

Disadvantages:

1. It takes lot of time to process the applications
2. Duplicate data is also going for processing there is no method for avoiding this.

Proposed system:

In the proposed system we present a novel task-scheduling framework for security sensitive workflows with three novel features. First, we provide comprehensive theoretical analyses on how selectively duplicating a task's predecessor tasks is helpful for preventing both the data Transmission time and encryption time from delaying task's start time. Then, we define workflow tasks' latest finish time, and prove that tasks can be completed before tasks' latest finish time by using cheapest resources to reduce monetary cost without delaying tasks' successors' start time and workflows' make spans. Based on these analyses, we devise a novel scheduling approach with selective tasks duplication, named *SOLID*, incorporating two

important phases: 1) task scheduling with selectively duplicating predecessor tasks to idle time slots on resources; and 2) intermediate data encrypting by effectively exploiting tasks' laxity times. We evaluate our solution approach through rigorous performance evaluation study using both randomly generated workflows and some real-world workflow traces.

Advantages:

1. Application performance has been improved by allocating the tasks to the different schedulers.
2. Processing data will have some pre checking methods for the duplications and others.

SYSTEM REQUIREMENTS

H/W System Configuration:-

- Processor - Pentium –III
- RAM - 256 MB (min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

S/W System Configuration:-

- Operating System : Windows95/98/2000/XP
- Application Server : Tomcat5.0/6.X
- Front End : HTML, Jsp
- Scripts : JavaScript.
- Server side Script : Java Server Pages.
- Database : MySQL 5.0
- Database Connectivity : JDBC