A Hybrid Simulation Approach for Competitive Open Software Development Process

Sponsor: DASD(SE)

By
Razieh Saremi
6th Annual SERC Doctoral Students Forum
November 7, 2018
FHI 360 CONFERENCE CENTER
1825 Connecticut Avenue NW
8th Floor
Washington, DC 20009

www.sercuarc.org
Table of Content

- Background
- Research Question and Methodology
- Research Design and Analysis
- Conclusion
Open Competitive Software Workflow

Open Market Software Development [Weiss 2005]:

- Potentially large number of unknown workers
- Have access to the internet
- Collaborate and coordinate on the tasks
- Workers take the work they choose

Scheduling

Re-Work

Open

Competitive
Task Scheduling in Software Engineering

✓ Scheduling a software project:
   ➢ Setting sequence of time dependent tasks that make a project [Mingozi et al 1998]
   ➢ Assign tasks to workers to be done in specific time frame [Alba et al 2007]

✓ Complex tasks require higher cooperation amongst co-workers [Malon 1994].

➢ Challenges in OSD Scheduling: [Gao et al 2015, Barreto et al 2008]
   ▪ Not knowing workers in person,
   ▪ Working from different time zone,
   ▪ Workers interest in other tasks among pool of open tasks

❑ Understanding the crowd workers’ sensitivity and performance to the arrival task [Difallah 2016]
   ▪ Zero registration,
   ▪ Zero submission and
   ▪ Not qualified submission.

▼ Improper scheduling would result in task failure [Faradiani et al 2011]
Motivation Example

# Similar Open Tasks

Project Duration: 110 Days
Project Failure Ratio: 57%
Limitations of Existing Methods

Task Context
- Task Similarity, Resource Reliability
- Required Skill-Set
- Same Batches of Tasks, Switch Context
- Resource Availability, Task Size, Task Priority

Worker Availability
- Flash Organization [Valentine et al 2017]
- Fair Scheduling /Hadoop-Yarn [Ghodsi 2011]
- Weighted Fair Sharing
- Fair Sharing
- Game with a purpose [Zaharia 2010]
- Basic Space Sharing [chi 2011]
- HIT Bundle [Difallah 2016]
- FIFO
- Shorter Job Fair
- Round Robin

Required Skill-Set
Is it feasible to provide a more effective automated task scheduling framework in competitive open software development environments in order to reduce task failure ratio?

Research Approach:

Trained Data Set:
- CSD data from Jan 2014 to Feb 2015
- extracted from TopCoder website
- 403 individual projects
- 4907 component development tasks
- 8108 workers, 5062 active workers
- 60433 worker-task participation records

Presented Model:

Hybrid simulation model:
 ✓ Systems Dynamic
 ✓ Discrete Event
 ✓ Agent Based Model
CSD Market Place

CSD Market

Demand:
CSD Mini-Tasks per week

- 76 New Arrival
- 2 Cancel
- 1 Starve
- 8 Fail

Supply:
CSD Workers per week

<table>
<thead>
<tr>
<th>Belt</th>
<th>Rating Range (X)</th>
<th># Workers</th>
<th>% Workers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray</td>
<td>X&lt;900</td>
<td>4557</td>
<td>90.02%</td>
</tr>
<tr>
<td>Green</td>
<td>900&lt;X&lt;1200</td>
<td>146</td>
<td>2.88%</td>
</tr>
<tr>
<td>Blue</td>
<td>1200&lt;X&lt;1500</td>
<td>273</td>
<td>5.39%</td>
</tr>
<tr>
<td>Yellow</td>
<td>1500&lt;X&lt;2200</td>
<td>78</td>
<td>1.54%</td>
</tr>
<tr>
<td>Red</td>
<td>X&gt;2200</td>
<td>8</td>
<td>0.16%</td>
</tr>
</tbody>
</table>
Study Design

Relaxed Assumptions:

- Workers’ trust factor is constant
  - Tasks will not be cancelled by client requests
    - Tasks will not face zero registration scenario
Micro level: Agent Model

Knowledge:

- 59% of workers respond to a task call
- 24% of the workers will make submissions

Agent’s Decision Making:

- Registering:
  \[ P(R_j) = \begin{cases} 
  1 & \text{Regs} \leq 18 \\
  \text{Bernoulli}(0.3) & \text{Regs} > 18 
  \end{cases} \]

- Submitting:
  \[ P(S_j) = \begin{cases} 
  0.6 & j \in \text{Red} \\
  0.6 & j \in \text{Yellow} \\
  0.39 & j \in \text{Blue} \\
  0.25 & j \in \text{Green} 
  \end{cases} \]

Workers’ Reliability (Re): Pert(0, 1, 0.19)
Meso level: Task Completion Model

Task Score/Quality: uniform (0,100)

\[ P(TS_i) = \begin{cases} 0 & \text{AS Massege} \neq 1 \\ 1 & \text{AS Massege} = 1 \end{cases} \]

\[ FPS_k = 0.0473(TSR_k) + 0.014 \]

\[ P(TR_i) = \begin{cases} 0 & \text{AR Massege} \neq 1 \\ 1 & \text{AR Massege} = 1 \end{cases} \]

\[ FPR_k = \begin{cases} \frac{\sum_i R_{ej} * P_j}{3} & \sum_i R_{ej} > 2 \\ \frac{\sum_i R_{ej} * P_j}{2} & 1 < \sum_i R_{ej} < 2 \\ \frac{\sum_i R_{ej} * P_j}{1} & \sum_i R_{ej} < 1 \end{cases} \]

\[ P(TW_i) = \begin{cases} Cancelled & \text{Score} < 75 \\ Complete & \text{Score} \geq 75 \end{cases} \]
Workers’ Arrival: Poisson(18, 800, 20, 0, 1)
Workers’ Experience: Beta (1, 5, 0, 3000)
Tasks’ Arrival: Task Model Schedule
Task Similarity: uniform (0.33, 0.98)
Model Accuracy

Success Ratio ~ 71%
Failure ratio ~ 13%

Failure prediction in Submissions (SFP) ~ 15%
Failure Prediction in Registration (RFP) ~ 6.5%

V.S.
Actual Failure Ratio ~ 14%

MRE (SFP) = 2%
MRE (RFP) = 1.1%
Scenario 1: How diverse?

Scenario 2: How open?

Task Similarity level < 60%

Close availability of midlevel experienced worker
Scenario 1: Agents’ Diversity

- Failure% ↓
- Submissions% ↑

PM only can manage the diversity of registrants’ experience

Scenario ran 30 times

Failure Prediction in Registration Phase

Time(hr)

Failure Prediction in Submissions Phase

Registration

Submission

Actual Failure

Gray Belt

Green Belt

Blue Belt

Yellow Belt
Scenario 2: Task Openness

- **Task Openness**
- **Failure Prediction in Registration Phase**
- **Failure Prediction in Submissions Phase**
- **Simulation Failure Prediction**

### Task Status

<table>
<thead>
<tr>
<th>Scenario Status</th>
<th>Scenario 1</th>
<th>60% Similarity</th>
<th>70% Similarity</th>
<th>80% Similarity</th>
<th>90% Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail</td>
<td>18</td>
<td>22</td>
<td>25</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Success</td>
<td>12</td>
<td>8</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Failure Prediction</td>
<td>60%</td>
<td>73%</td>
<td>83%</td>
<td>77%</td>
<td></td>
</tr>
</tbody>
</table>

- Scenario ran 30 times
- PM can manage openness of the pool of tasks only
Conclusion and Future Work

**Conclusion:**

This study provides a hybrid simulation model to help providing more insights in order to have a more efficient task scheduling in OSD.

- Attracting higher number of middle ranked agents to compete on the task, would provide lower chance of task failure in general.

- Similarity level of 60% and lower in the pool of available tasks, provides lower chance of task failure.

**Future Work:**

- Updating the model with schedule available projects form the entire data set and report the recommendation metrics
Limitation

- Presented model created based on competitive crowdsourcing only
- No access to the management dataset and overheads
- No access to the actual task sequential per project
- Assuming that monitory prize and task duration represents task complexity
- Different factors that may influence workers’ decision-making process
Thank you!