Case Study #3: Analysis of Replicated Data with Repair Dependency

Ing-Ray Chen and Ding-Chau Wang
The Computer Journal
Vol. 39, No. 9, 1996, pp. 767-779
Replicated data management

• Extend Case Study 1 by considering both node and link failures/recovery as well as the effect of repair dependency which occurs when many sites and links may have to share the same repairman due to repair constraints.
Dynamic voting for replicated data management

– Dynamic voting: Each site $S_i$ maintains $(VN_i, SC_i, DS_i)$ to understand if it is in the major partition

– Site $i$ is in the major partition if:
  
  • the number of copies it can access is larger than one half of $SC_i$
  
  • the number of copies it can access is exactly equal to one half of $SC_i$ and it can access the “distinguished site” indicated in $DS_i$

– If a site is in the major partition, it can update locally. After an update is done, all copies along with $(VN_i, SC_i, DS_i)$’s in the major partition are updated
System model

- Sites and links have independent failure rates $\lambda_s$ and $\lambda_l$.
- A repairman repairs a failed site with rate $\mu_s$ and a failed link with rate $\mu_l$.
- There is always an update (called an immediate update) after a failure or repair event since the update rate is much faster than the failure/repair rate.

Site subnet
- A site can be in one of four states:
  - up and current (upcc)
  - up and out-of-date (upoc)
  - down and current (downcc)
  - down and out-of-date (downoc)
When an update arrives and a major partition exists, a token will be put into ready.

Site Subnet Model: one for each site

\[ g_i \text{ true: } i \text{ is in major partition} \]
\[ \overline{g_i} \text{ true: } i \text{ is not in major partition} \]

Only one out of the six transitions can fire, all with the same priority level (3)

\[ (t_1, 3, g_i) \]
\[ (t_2, 3, g_i) \]
\[ (t_3, 3, g_i) \]
\[ (t_4, 3, g_i) \]
\[ (t_5, 3, \overline{g_i}) \]

- \( t_5 \) is the name of the transition, 3 is the priority of the transition, and \( \overline{g_i} \) is the enabling function.
- \( t_5 \) will fire if \( \overline{g_i} \) is true and site i is up and current.
- After \( t_5 \) fires, upcc->upoc and site i is up and out of date.

\[ \text{FIGURE 1. Local status update actions by site } i. \]
gi true: i is in major partition
\bar{g}_i true: i is not in major partition

FIGURE 1. Local status update actions by site i.
t0 will fire if site i is down and current
After t0 fires, site i is down and out of date (downoc)

\( g_i\) true: i is in major partition
\( \bar{g}_i\) true: i is not in major partition

FIGURE 1. Local status update actions by site i.
\( g_i \) true: i is in major partition
\( \bar{g}_i \) true: i is not in major partition

FIGURE 1. Local status update actions by site \( i \).
• Transitions \( tf \) and \( tfbar \) are given the highest priority levels (5 and 4)

• When an update event arrives, a token will be put in place “update event”

\[ f \text{ true: there is a major partition} \]
\[ \neg f \text{ true: there isn’t a major partition} \]

Each of the boxes labeled site \( i \) is the site subnet model for site \( i \)

**FIGURE 2.** Global status-update actions triggered by an update operation.
(tsc, 1) updates the site cardinality:
The multiplicity of tsc → sc is the number of sites with mark (upcc) >0 in the site subnet model.
The new site cardinality is stored as the number of tokens in place sc.
(tds, 2) updates distinguished site:
The multiplicity of $tds \rightarrow ds$ is the maximum $(upcc)$ value among all the sites in the major partition
The site ID of the new distinguished site will be stored as the number of tokens in place $ds$

**FIGURE 2.** Global status-update actions triggered by an update operation.
Independent Repairman Model

- This subnet describes the effect of site i’s failure and repair on the system state.
- Site i can only be in one state at a time, so only one transition out of these two subnets is possible at any time.

**Independent Repairman Model**

- subscript i refers node i
- failure events: $\text{upcc}_i \rightarrow \text{dwcci}$ and $\text{upoci} \rightarrow \text{dwoci}$ with rate of $\lambda_s$
- repair events: $\text{dwcci} \rightarrow \text{upcc}_i$ and $\text{dwoci} \rightarrow \text{upoci}$ with rate of $\mu_s$

**FIGURE 3.** Site failure/repair events.
Independent Repairman Model

Link failure/repair subnets for each link $ij$

- subscript $ij$ refers to the link between nodes $i$ and $j$
- failure events: $\text{uplink}_{ij} \rightarrow \text{dwlink}_{ij}$
  - With rate of $\lambda_i$
- repair events: $\text{dwlink}_{ij} \rightarrow \text{uplink}_{ij}$
  - With rate of $\mu_i$

**FIGURE 4.** Link failure/repair events.
## TABLE 1. Meanings of places.

<table>
<thead>
<tr>
<th>Place</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$upcc_i$</td>
<td>$Copy_i$ is up and current</td>
</tr>
<tr>
<td>$downcc_i$</td>
<td>$Copy_i$ is down and current</td>
</tr>
<tr>
<td>$upoc_i$</td>
<td>$Copy_i$ is up and out of date</td>
</tr>
<tr>
<td>$downoc_i$</td>
<td>$Copy_i$ is down and out of date</td>
</tr>
<tr>
<td>$uplink_{ij}$</td>
<td>$Link_{ij}$ is up</td>
</tr>
<tr>
<td>$dwlink_{ij}$</td>
<td>$Link_{ij}$ is down</td>
</tr>
<tr>
<td>$update_event$</td>
<td>An update is initiated</td>
</tr>
<tr>
<td>$sc_ready$</td>
<td>An SC is initiated</td>
</tr>
<tr>
<td>$sc$</td>
<td>$#(sc)$ indicates the SC</td>
</tr>
<tr>
<td>$ds_ready$</td>
<td>A DS change is initiated</td>
</tr>
<tr>
<td>$ds$</td>
<td>$#(ds)$ indicates the ID of the DS</td>
</tr>
<tr>
<td>$ready_i$</td>
<td>A local update at site $i$ is in process</td>
</tr>
</tbody>
</table>
### TABLE 2. Arc multiplicity functions.

<table>
<thead>
<tr>
<th>Arc</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$sc \rightarrow tsc$</td>
<td>$(sc)$</td>
</tr>
<tr>
<td>$tsc \rightarrow sc$</td>
<td># of sites in the major partition with mark($upcc$) &gt; 0</td>
</tr>
<tr>
<td>$ds \rightarrow tds$</td>
<td>$(ds)$</td>
</tr>
<tr>
<td>$tds \rightarrow ds$</td>
<td>Max #($upcc$) among all sites in the major partition</td>
</tr>
</tbody>
</table>
TABLE 4. Enabling functions.

<table>
<thead>
<tr>
<th>Tr.</th>
<th>Enabling function</th>
</tr>
</thead>
</table>
| $tf$ | $f()$
  
  {IF $\exists$ a partition $\mathcal{M}$ with sum equal to
   
   # of sites in $\mathcal{M}$ with mark($upcc$) > 0;
   
   AND IF (sum $>$ #(sc)/2) OR
   
   (sum $=$ #(sc)/2 AND
   
   mark($upcc_{#(ds)}$) AND site #(ds) $\in \mathcal{M}$)
   
   THEN RETURN 1;
   
   ELSE RETURN 0} |
| $t_{2i}$ | $g_i()$
  
  {Look at mark($uplink_{jk}$) $\forall j \forall k$ to
determine site $i$’s partition;
IF site $i$’s in the major partition $\mathcal{M}$
THEN RETURN 1;
ELSE RETURN 0} |
| $t_{3i}$ | $g_i()$ |
| $t_{4i}$ | $\bar{g}_i()$ {1 $-$ $g_i()$} |
| $t_{5i}$ | $\bar{g}_i()$ |
FIFO repairman model

- We can make use of the independent repairman model and modify the repair rates to account for repair dependencies.
- FIFO repairman model has only one repairman.
- The repair rate is “deflated” by the total number of failed sites and links to account for the effect of repair resource sharing.
- If a state has 3 failed entities: two failed sites and one failed link,
  - For the independent repairman model, repair rates are $\mu_s$, $\mu_s$ and $\mu_l$.
  - For the FIFO repairman model, repair rates are $\mu_s / 3$, $\mu_s / 3$ and $\mu_l / 3$. 


228
**TABLE 5.** Rates of timed transitions for FIFO repair.

<table>
<thead>
<tr>
<th>Timed Tr.</th>
<th>Rate value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$tccf_i$</td>
<td>$\lambda_s$ $\mu_s$</td>
</tr>
<tr>
<td>$tccr_i$</td>
<td>$\lambda_s$ $\sum_{j,k,k\neq j} (\text{downcc}_j + \text{downoc}<em>j + \text{dwlink}</em>{jk})$</td>
</tr>
<tr>
<td>$tocf_i$</td>
<td>$\lambda_s$ $\mu_s$</td>
</tr>
<tr>
<td>$tocr_i$</td>
<td>$\lambda_s$ $\sum_{j,k,k\neq j} (\text{downcc}_j + \text{downoc}<em>j + \text{dwlink}</em>{jk})$</td>
</tr>
<tr>
<td>$tlinkf_{ij}$</td>
<td>$\lambda_l$ $\mu_l$</td>
</tr>
<tr>
<td>$tlinkr_{ij}$</td>
<td>$\lambda_l$ $\sum_{j,k,k\neq j} (\text{downcc}_j + \text{downoc}<em>j + \text{dwlink}</em>{jk})$</td>
</tr>
</tbody>
</table>
Linear-order repairman model

- Repairing failed site/link in a prescribed order
- Giving a higher repair priority to a higher linearly ordered site to increase the chance of having a major partition
- Creating a new enabling function associated with each repair transition
- Only one enabling function at any state returns TRUE based on the prescribed linear order and all others return FALSE
Linear-order repairman model

- Example:
A 5-site ring topology with the linear repair order being sites 5,4,3,2,1 followed by links 45,51,43,32,21.

Suppose sites 2 and 4, and link 51 are down.
Site 4 is chosen to be repaired first.
Linear-order repairman model

- Enabling functions associated with sites 4 and 2 and link 51 will return TRUE, FALSE and FALSE, respectively, meaning that site 4 will be repaired first over site 2 and link 51.
<table>
<thead>
<tr>
<th>Transition</th>
<th>Enabling function</th>
</tr>
</thead>
</table>
| $tccr_i, toc_r_i$ | $h_{\text{site}}(i)$  
{IF site $i$ failed and the repair rank of site $i$ is higher than those of other failed sites or links in the linear order THEN RETURN TRUE; ELSE RETURN FALSE} |
| $t\text{link}_{ij}$ | $h_{\text{link}}(i, j)$  
{IF link $ij$ failed and the repair rank of link $ij$ is higher than those of other failed sites or links in the linear order THEN RETURN TRUE; ELSE RETURN FALSE} |
Best-first repairman model

• Preference is given to the site or link which can lead to the existence of a major partition after its repair with respect to the current state

• If there are more than one failed sites or links whose repair would lead to the existence of a major partition, then a tie-breaker rule will be applied to select one to be repaired next.
Best-First Repair Strategy
Tie-Breaker Rules

• Choosing a failed entity such that after repair it will lead to the largest SC in the major partition (i.e., the more sites in the major partition, the better).

• Choosing a site (among failed sites) that is the highest linearly ordered site, so it has a better chance to become the DS.

• Choosing a failed entity that will stay alive for a longer time after repair. For example, when choosing between a failed site vs. a failed link, if $\frac{\mu_s}{\lambda_s} \geq \frac{\mu_l}{\lambda_l}$, then repair the failed site, otherwise repair the failed link.
Best-First Repair Example

Effective:
- best-first repair (SC is 4 after repair)

Ineffective:
- linear-order repair (SC is 3 after repair)
<table>
<thead>
<tr>
<th>Transition</th>
<th>Enabling function</th>
</tr>
</thead>
</table>
| $tccr_i, tocr_i$ | $h_{1_{site}}(i)$  
{IF site $i$ failed and the hypothetical site availability after repairing site $i$ is higher than those of other failed sites or links in the system  
THEN RETURN TRUE;  
ELSE RETURN FALSE} |
| $tlinkr_{ij}$ | $h_{1_{link}}(i, j)$  
{IF link $ij$ failed and the hypothetical site availability after repairing link $ij$ is higher than those of other failed sites or links in the system  
THEN RETURN TRUE;  
ELSE RETURN FALSE} |
Evaluation

• Tested with a 5-site ring topology

• Four repairman models:
  – Independent repair
  – Dependent repair
    • FIFO
    • Linear-order
    • Best-first
Model complexity: number of states

<table>
<thead>
<tr>
<th></th>
<th>Independent</th>
<th>FIFO</th>
<th>Linear-order</th>
<th>Best-first</th>
</tr>
</thead>
<tbody>
<tr>
<td># of states in the underlying Markov model</td>
<td>8674</td>
<td>8674</td>
<td>5429</td>
<td>3821</td>
</tr>
</tbody>
</table>
Performance metrics and reward assignments for calculation

<table>
<thead>
<tr>
<th>System Availability</th>
<th>Definition</th>
<th>Reward Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The steady-state probability that a major partition exists</td>
<td><strong>Reward rate = 1</strong> for those states in which enabling function $f()$ is evaluated to TRUE. Reward rate = 0, otherwise</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Site Availability</th>
<th>Definition</th>
<th>Reward Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The probability that an update arriving at an arbitrary site will succeed</td>
<td><strong>Reward rate = $1\times k/n$</strong> for those states in which enabling function $f()$ is evaluated to TRUE where $k$ is the number of up and current copies in the major partition. Reward rate = 0, otherwise</td>
</tr>
</tbody>
</table>

$k$: # of ‘up and current’ site in the major partition in a particular state  
$n$: total number of sites in a system (n=5 in a 5-site ring topology)
Results:

independent repairman model

\[ \mu_i \cdot \lambda_i \uparrow \quad \text{site availability} \uparrow \]

\[ \mu_s \cdot \lambda_s \uparrow \quad \text{site availability} \uparrow \]

Fallible sites assumption (as in Case Study 1) will overestimate the site availability unrealistically.

FIGURE 5. Site availability of five-site ring under independent repairman model.
Results: Comparison of repairman models

- Independent Repair > Dependent Repair
- Dependent Repair:
  - Best-first > Linear-Order > FIFO

**FIGURE 6.** Site availability of five-site ring under various repairman models.
Results: Comparison of repairman models

- System availability is higher than the corresponding site availability
- Independent > Best-first > Linear-Order > FIFO also holds for the system availability metric

FIGURE 7. System availability of five-site ring under various repairman models.