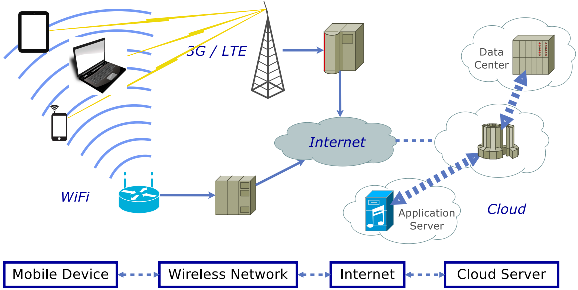
Reducing Task Completion Time in Mobile Offloading Systems through Online Adaptive Local Restart

Qiushi Wang and **Katinka Wolter**

Freie Universität Berlin
Institut für Informatik

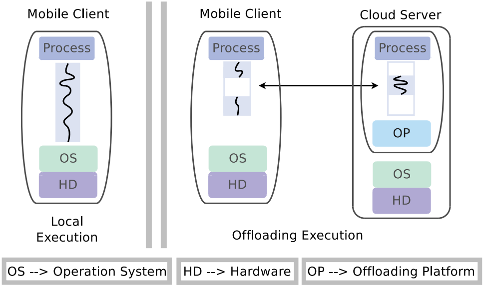
2nd February 2015

Mobile Offloading System



- Mobile cloud computing architecture.

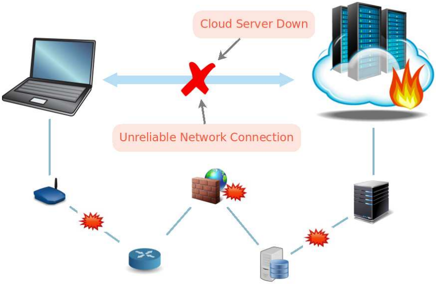
Mobile Offloading System



- Mobile cloud computing architecture.
- Offloading system $\Longrightarrow$ $Advantages:$

$$\overbrace{\text{Shorten execution time} + \text{Reduce energy consumption}}$$

Mobile Offloading System

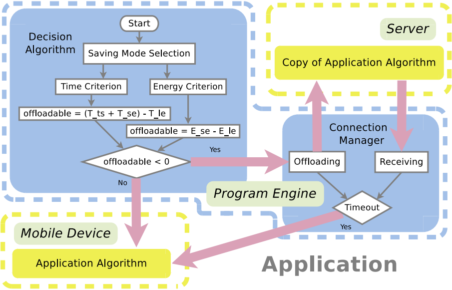


- Mobile cloud computing architecture.
- Offloading system $\Longrightarrow$ $Advantages:$

$$\overbrace{\text{Shorten execution time} + \text{Reduce energy consumption}}$$

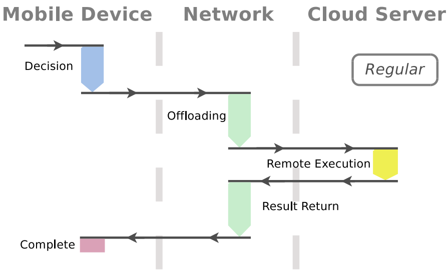
- Failures or delays can happen in the network and in the cloud server.
- how long should one wait and when to restart locally?

Offloading system architecture



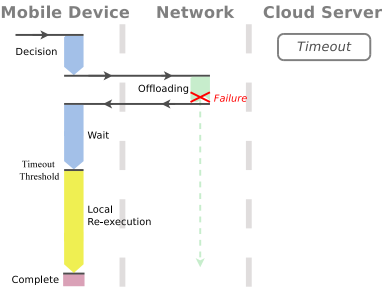
- Program engine and decision algorithm run on mobile device, that interacts with cloud server

Offloading system architecture



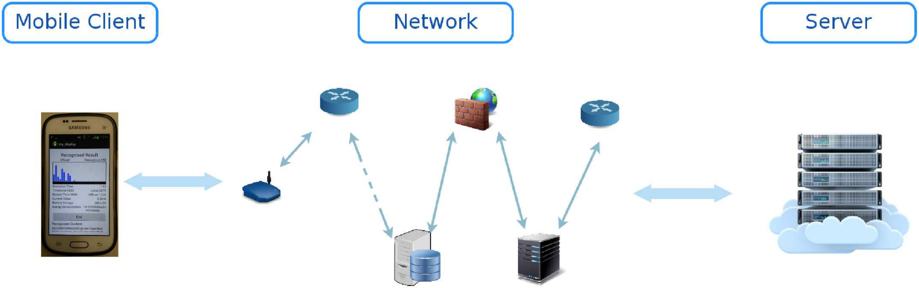
- Program engine and decision algorithm run on mobile device, that interacts with cloud server
- Offloading uses copy of algorithm in the server

Offloading system architecture



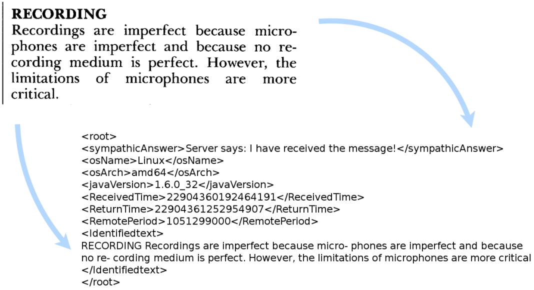
- Program engine and decision algorithm run on mobile device, that interacts with cloud server
- Offloading uses copy of algorithm in the server
- Local restart will not offload but executes in mobile device
- Online decision making needs adaptive timeout.

The test bed



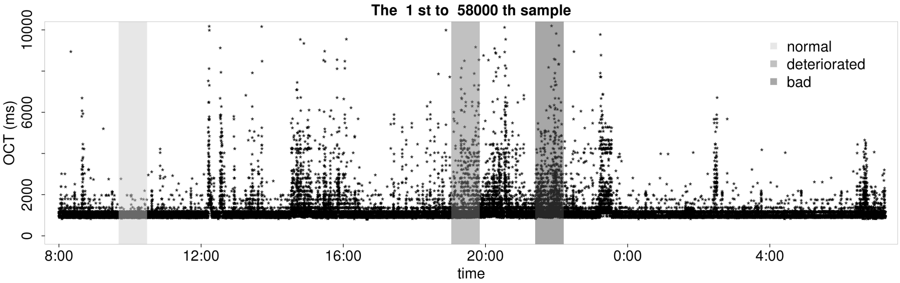
- Mobile Devices: Samsung GT-S7568, Android 4.0;
- Wireless Network: 54M/s WiFi The route passes 12 hops and the round-trip time is 82ms.
- Server: Server with 4 cores (Intel Xeon CPU E5649 2.53 GHz)

The test bed



- Mobile Devices: Samsung GT-S7568, Android 4.0;
- Wireless Network: 54M/s WiFi The route passes 12 hops and the round-trip time is 82ms.
- Server: Server with 4 cores (Intel Xeon CPU E5649 2.53 GHz)
- Sample Application: Optical Character Recognition (OCR)

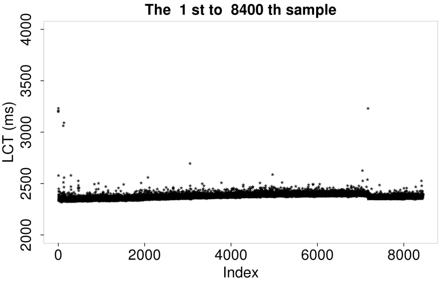
Task Completion Time



- The distribution of the sample values is not identical across time.

  OCT: Offloading Task completion time

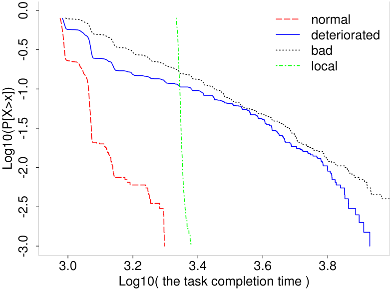
Task Completion Time



- The distribution of the sample values is not identical across time.
- Local computation is usually stable, with very few outliers.

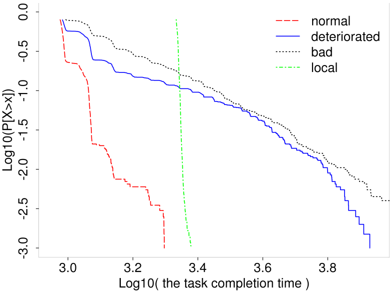
  LCT: Local Executed Task completion time

Heavy-tailed Distribution



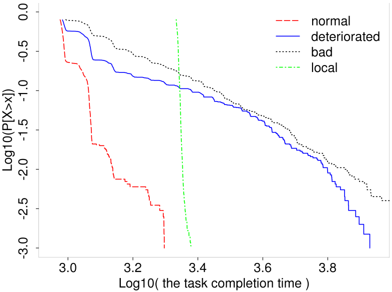
- Bad network condition: the curve has an approximately constant slope of $-2 \rightarrow$ a heavy tail.

Heavy-tailed Distribution



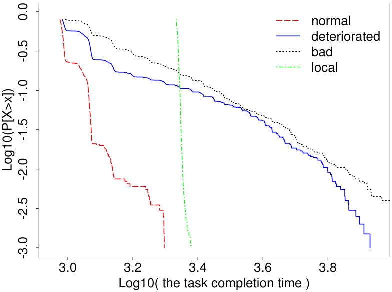
- Bad network condition: the curve has an approximately constant slope of $-2 \rightarrow$ a heavy tail.
- Deteriorated network condition: the tail has an exponential decay for long task completion times $\rightarrow$ heavy tail is not clear.

Heavy-tailed Distribution



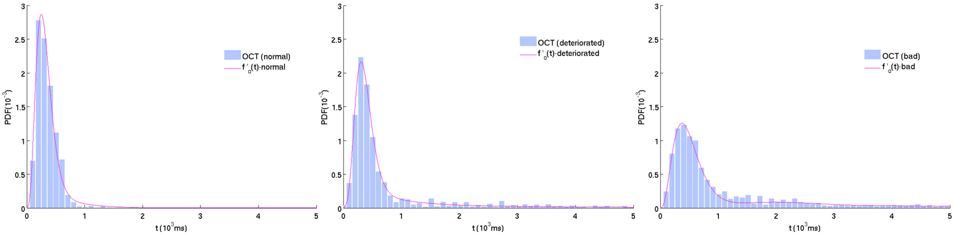
- Bad network condition: the curve has an approximately constant slope of $-2 \rightarrow$ a heavy tail.
- Deteriorated network condition: the tail has an exponential decay for long task completion times $\rightarrow$ heavy tail is not clear.
- Normal network condition: the decrease of the tail is steep $\rightarrow$ no heavy tail.

Heavy-tailed Distribution



- Bad network condition: the curve has an approximately constant slope of $-2 \rightarrow$ a heavy tail.
- Deteriorated network condition: the tail has an exponential decay for long task completion times $\rightarrow$ heavy tail is not clear.
- Normal network condition: the decrease of the tail is steep $\rightarrow$ no heavy tail.
- Local completion: the tail is almost infinite $\rightarrow$ no heavy tail.

Distribution Fitting Results



- Use Hyperstar for distribution fitting
- Task completion times has a lower threshold T^o_{min}
- Shift histogram to the left, $f_o(t) = f'_o(t - T^o_{min})$ to avoid zero density at the origin
- $f'_o(t)$ is the PH fitting result

Restart Condition

- $E[T] < E[T - \tau | T > \tau]$
  T: task completion time
  τ: restart timeout
  Expected completion time less than expected remaining time until completion

Restart Condition

- $E[T] < E[T - \tau | T > \tau]$
  T: task completion time
  τ: restart timeout
  Expected completion time less than expected remaining time until completion

- For one local restart

$$F(t) = \begin{cases} F_o(t) & (0 \leqslant t < \tau) \\ 1 - (1 - F_o(\tau))(1 - F_l(t - \tau)) & (\tau \leqslant t) \end{cases}$$

$$f(t) = \begin{cases} f_o(t) & (0 \leqslant t < \tau) \\ (1 - F_o(\tau)) f_l(t - \tau) & (\tau \leqslant t) \end{cases}$$

$f(t)$ and $F(t)$ are the density and cumulative distribution function

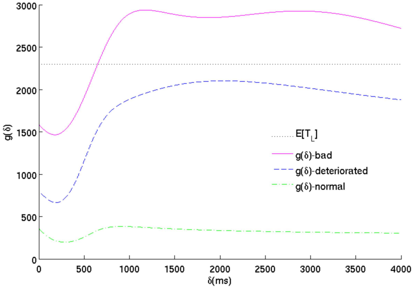
Local Restart Condition

- If $E[T] < E[T_o]$, restart is beneficial.

$$E[T_l] < \frac{\int_{\tau}^{\infty} t f_o(t) dt}{1 - F_o(\tau)} - \tau$$

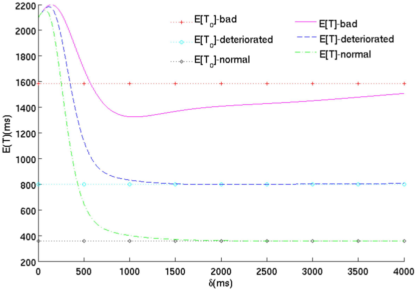
- $E[T_l] < \frac{\int_{\tau - T^o_{min}}^{\infty} t f'_o(t) dt}{1 - F'_o(\tau - T^o_{min})} - (\tau - T^o_{min})$
- $g(\delta) = \frac{\int_{\delta}^{\infty} t f'_o(t) dt}{1 - F'_o(\delta)} - \delta$
  $(\delta = \tau - T^o_{min})$

Optimal Restart Timeout



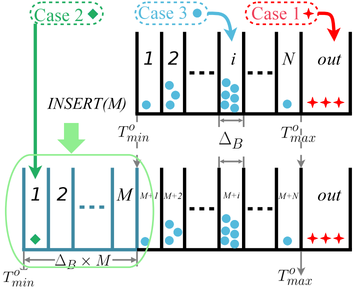
- $g(\delta) > E[T_l]$, the local restart is useful

Optimal Restart Timeout



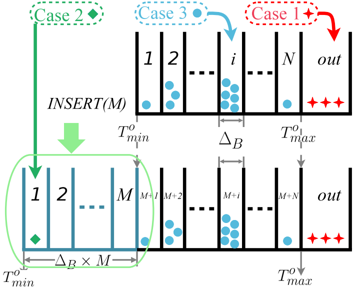
- $g(\delta) > E[T_l]$, the local restart is useful
- The optimal timeout is found when $E[T]$ is minimal.

Dynamic Restart Scheme



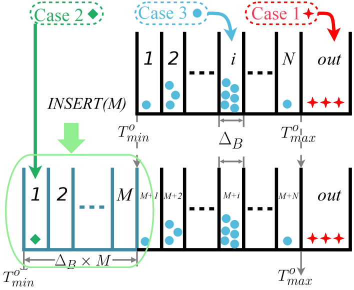
- $g(\delta)$ and $E[T]$ are estimated from the histogram

Dynamic Restart Scheme



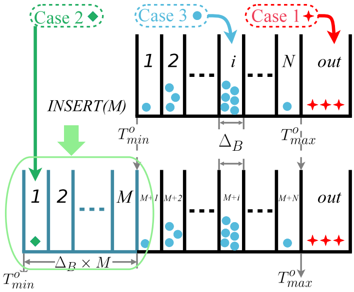
- $g(\delta)$ and $E[T]$ are estimated from the histogram
- Case 1: large items in outlier bucket,

Dynamic Restart Scheme



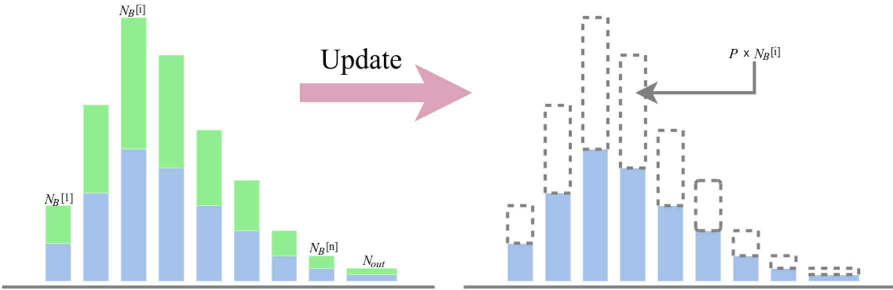
- $g(\delta)$ and $E[T]$ are estimated from the histogram
- Case 1: large items in outlier bucket, case 2: small items create new bucket,

Dynamic Restart Scheme



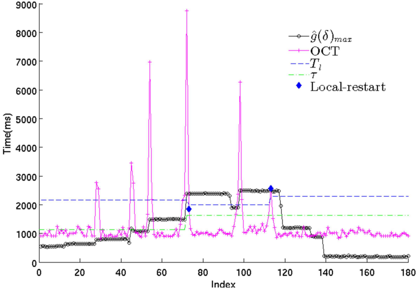
- $g(\delta)$ and $E[T]$ are estimated from the histogram
- Case 1: large items in outlier bucket, case 2: small items create new bucket, case 3: medium items go into existing buckets.

Dynamic Restart Scheme



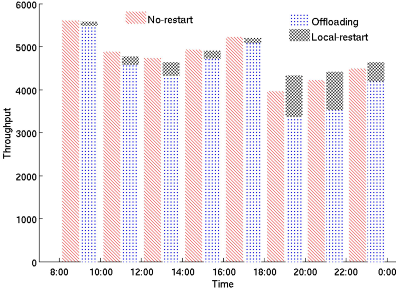
- $g(\delta)$ and $E[T]$ are estimated from the histogram
- Case 1: large items in outlier bucket, case 2: small items create new bucket, case 3: medium items go into existing buckets.
- Dynamic histogram uses partial flush

Evaluation of the Dynamic Restart



- When $\hat{g}(\delta)_{max} > T_l$ local restarts happen

Evaluation of the Dynamic Restart



- When $\hat{g}(\delta)_{max} > T_l$ local restarts happen
- At certain times (evening), dynamic local restart can increase the throughput.

Conclusions

- Mobile offloading is not always beneficial
  * Experiments illustrate the impact of network delays

Conclusions

- Mobile offloading is not always beneficial
  - Experiments illustrate the impact of network delays
- Derive expected completion time under one local restart
  - Can reduce task completion time

Conclusions

- Mobile offloading is not always beneficial
  - Experiments illustrate the impact of network delays
- Derive expected completion time under one local restart
  - Can reduce task completion time
- Dynamic local restart scheme
  - Dynamic histogram adaptively tracks variation of network quality
  - Use dynamic histogram for restart decision and timeout.

Conclusions

- Mobile offloading is not always beneficial
  * Experiments illustrate the impact of network delays
- Derive expected completion time under one local restart
  * Can reduce task completion time
- Dynamic local restart scheme
  * Dynamic histogram adaptively tracks variation of network quality
  * Use dynamic histogram for restart decision and timeout.
- General solution for other applications?

Conclusions

- Mobile offloading is not always beneficial
  - Experiments illustrate the impact of network delays
- Derive expected completion time under one local restart
  - Can reduce task completion time
- Dynamic local restart scheme
  - Dynamic histogram adaptively tracks variation of network quality
  - Use dynamic histogram for restart decision and timeout.
- General solution for other applications?
- Many restarts, mixed local and remote

Thank you.