FLBT

0.1 Motivation

NEEDS TO BE WRITTEN

0.2 FBLT

Algorithm 1: FBLT

Data: s_i^k: interfered transaction;
s_j^l: interfering transactions;
$\delta_i^k(L)$: the maximum number of times s_i^k can be aborted during an interval L;
η_i^k: number of times s_i^k has already been aborted during an interval L;
m_set: contains at most m non-preemptive transactions;
m_prio: priority of any transaction in m_set. m_prio is higher than any priority of any real-time task;
$r(s_i^k)$: time point at which s_i^k joined m_set;
Result: atomic sections that will abort

1 **if** $p_j^l > p_i^k$ **then**
2   **if** $\eta_i^k(L) \leq \delta_i^k(L)$ **then**
3   Increment η_i^k by 1;
4   Abort s_i^k;
5   **else**
6   **if** $s_i^k \notin m\_set$ **then**
7   Add s_i^k to m_set;
8   Record $r(s_i^k)$;
9   Increase priority of s_i^k to m_prio;
10   **end**
11   **if** $s_j^l \in m\_set\ AND\ r(s_j^l) < r(s_i^k)$ **then**
12   Abort s_i^k;
13   **else**
14   Abort s_j^l;
15   **end**
16   **end**
17 **else**
18   Swap s_i^k and s_j^l;
19   Go to Step 4
20 **end**

0.2.1 Illustrative Example

NEEDS TO BE WRITTEN

0.3 Properties

0.4 Retry Cost Under FBLT

We now derive an upper bound on the retry cost of any job τ_i^x under FBLT during an interval $L \leq T_i$. Since all tasks are sporadic (i.e., each task τ_i has a minimum period T_i), T_i is the maximum study interval for each task τ_i.

Claim 1 *The total retry cost for any job τ_i^x due to: 1) conflicts between its transactions and transactions of other jobs under FBLT during an interval $L \leq T_i$. 2) release of higher priority jobs, is upper bounded by:*

$$RC_{to}(L) \leq \sum_{\forall s_i^k \in s_i} \left(\delta_i^k(L) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k) \right) + RC_{re}(L) \tag{1}$$

where χ_i^k is the set of at most $m-1$ maximum length transactions sharing objects with s_i^k. Each transaction $s_{iz}^k \in \chi_i^k$ belongs to a distinct task $\tau_j, j \neq i$. $RC_{re}(L)$ is the retry cost resulting from release of higher piority jobs which preempt τ_i^x. $RC_{re}(L)$ is calculated by (6.8) in [2] for G-EDF, and (6.10) in [2] for G-RMA.

Proof 1 *By definition of FBLT, $s_i^k \in \tau_i^x$ can be aborted at maximum $\delta_i^k(L)$ times during interval L before s_i^k joins m_set. Before joining m_set, s_i^k can be aborted due to higher priority transactions, or transactions in the m_set. Original priority (before step) of transactions in m_set can be of higher or lower priority than p_i^x. Thus, the maximum time s_i^k is aborted before joining m_set occurs if s_i^k is aborted for δ_i^k. The worst case scenario for s_i^k after joining m_set occurs if s_i^k is preceded by $m-1$ maximum lenght conflicting transactions. Hence, s_i^k has to wait for the previous $m-1$ transactions to commit first. Priority of s_i^k after joining m_set is higher than any real-time task. So, s_i^k is not aborted by any task. If s_i^k has not joined m_set yet, and a higher piority job τ_j^y is released while s_i^k is running, then s_i^k may be aborted if τ_j^y has conflicting transactions with s_i^k. τ_j^y causes only one abort in τ_i^x because τ_j^y preempts τ_i^x only once. If s_i^k has already joined m_set, then s_i^k cannot be aborted by release of higher priority jobs. So, the maximum number of abort times to transactions in τ_i^x due to release of higher prirority jobs is less or equal to number of interfering higher priority jobs to τ_i^x. Claim follows.*

Claim 2 *The blocking time for a job τ_i^x due to lower priority jobs during an interval $L \leq T_i$ is upper bounded by:*

$$D(\tau_i^x) = min\left(max_1^m(s_{j_{max}, \forall \tau_j^l, p_j^l < p_i^x})\right) \tag{2}$$

where $s_{j_{max}}$ is the maximum length transaction in any job τ_j^l with original priority lower than p_i^x. The right hand side of (2) is the minimum of the m maximum transactional lengths in all jobs with lower priority than τ_i^x.

Proof 2 *τ_i^x is blocked when it is initially released and all processors are busy with lower priority jobs with non-preemptive transactions. Although τ_i^x can be preempted by higher priority jobs, τ_i^x cannot be blocked after it is released. If τ_i^x is preempted by a higher priority job τ_j^y, then τ_j^y finishes execution, the underlying scheduler will not choose a lower priority job than τ_i^x before τ_i^x. So, after τ_i^x is released, there is no chance for any transaction s_u^v belonging to a lower priority job than τ_i^x to run before τ_i^x. Thus, s_u^v cannot join m_set before τ_i^x finishes. Consequently, the worst case blocking time for τ_i^x occurs when the maximum length m transactions in lower priority jobs than τ_i^x are executing non-preemptively. After the minimum length transaction in the m_set finishes, the underlying scheduler will choose τ_i^x or a higher priority job to run. Claim follows.*

Claim 3 *Response time of any job τ_i^x during an interval $L \leq T_i$ under FBLT is upper bounded by*

$$R_i^{up} = c_i + RC_{to}(L) + D(\tau_i^x) + \left\lfloor \frac{1}{m} \sum_{\forall j \neq i} W_{ij}(R_i^{up}) \right\rfloor \tag{3}$$

where $RC_{to}(L)$ is calculated by (1), $D(\tau_i^x)$ is calculated by by (2), and $W_{ij}(R_i^{up})$ is calculated by (11) in [3] for G-EDF, and (17) in [3] for G-RMA. (11) and (17) in [3] inflates c_j of any job of $\tau_j \neq \tau_i$, $p_j > p_i$ by retry cost of transactions in τ_j.

Proof 3 *Response time of any job τ_i^x is a direct result of FBLT bahviour. Response time of any job τ_i^x is the sum of its worst case execution time c_i, plus retry cost of transactions in τ_i^x ($RC(L)$), plus blocking time of τ_i^x ($D(\tau_i^x)$), and the workload interference of higher priority jobs. Workload interfernence of higher priority jobs scheduled by G-EDF is calculated by (11) in [3], and by (17) in [3] for G-RMA.*

0.5 FBLT vs. Competitors

Let $RC_A(T_i)$ denote the retry cost of any τ_i^x using the synchronization method A during T_i. Let $RC_B(T_i)$ denote the retry cost of any τ_i^x using synchronization method B during T_i.

Then, schedulability of A is comparable to B if

$$\sum_{\forall \tau_i} \frac{c_i + RC_A(T_i)}{T_i} \leq \sum_{\forall \tau_i} \frac{c_i + RC_B(T_i)}{T_i}$$
$$\sum_{\forall \tau_i} \frac{RC_A(T_i)}{T_i} \leq \sum_{\forall \tau_i} \frac{RC_B(T_i)}{T_i} \quad (4)$$

0.5.1 FBLT vs. ECM

Claim 4 *Schedulability of FBLT is equal or better to ECM's when maximum number of abort times of any transaction s_i^k in any job of τ_i is less or equal to number of conflicting transactions to s_i^k in all other jobs with higher priority than priority of current job of τ_i.*

Proof 4 *By substituing $RC_A(T_i)$ and $RC_B(T_i)$ in (4) with (1) and (6.7) in [2] respectively.*

$$\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k)\right) + RC_{re}(T_i)}{T_i} \quad (5)$$
$$\leq \sum_{\forall \tau_i} \frac{\left(\sum_{\forall \tau_j \in \gamma_i^{ex}} \sum_{\theta \in \theta_i^{ex}} \left(\left\lceil \frac{T_i}{T_j} \right\rceil \sum_{\forall \bar{s_j^h}(\theta)} len\left(\bar{s_j^h}(\theta) + s_{max}^j(\theta)\right)\right)\right) + RC_{re}(T_i)}{T_i}$$

Let $\theta_i^{ex} = \theta_i + \theta_i^*$ where θ_i^* is the set of objects not accessed directly by τ_i but can enforce transactions in τ_i to retry due to transitive retry. Let $\gamma_i^{ex} = \gamma_i + \gamma_i^*$where γ_i^* is the set of tasks that access objects in θ_i^*. $\bar{s_j^h}(\theta)$ can access multiple objects, so $s_{max}^j(\theta)$ is the maximum length transaction conflicting with $\bar{s_j^h}(\theta)$. FBLT needs to bound transitive retry. So, θ_i^{ex} will be avoided in (5). Thus, transactions under ECM behave as if there were no transitive retry. Consequently, (5) will be

$$\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k)\right)}{T_i} \quad (6)$$
$$\leq \sum_{\forall \tau_i} \frac{\left(\sum_{\forall \tau_j \in \gamma_i} \sum_{\theta \in \theta_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil \sum_{\forall \bar{s_j^h}(\theta)} len\left(\bar{s_j^h}(\theta) + s_{max}^j(\theta)\right)\right)\right)}{T_i}$$

As $\bar{s_j^h}(\theta)$ is included only once for all objects accessed by it. $s_{max}^j(\theta)$ is also included once for each $\bar{s_j^h}(\theta)$. Consequently, (6) becomes

$$\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k)\right)}{T_i} \quad (7)$$
$$\leq \sum_{\forall \tau_i} \frac{\left(\sum_{\forall \tau_j \in \gamma_i} \sum_{\bar{s_j^h}(\theta),\, \theta \in \theta_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil len\left(\bar{s_j^h}(\theta) + s_{max}^j(\theta)\right)\right)\right)}{T_i}$$

Although different s_i^k *can have common conflicting transactions* $\bar{s_j^h}$*, but no more than one* s_i^k *can be preceeded by the same* $\bar{s_j^h}$ *in the m_set. This happens because transactions in the m_set are non-preemptive. Original priority of transactions preeceding* s_i^k *in the m_set can be of lower or higher priority than original priority of* s_i^k*. Under G-EDF,* $\tau_j \neq \tau_i$ *can have at least one job of higher priority than* τ_i^x*, then* $\left\lceil \frac{T_i}{T_j} \right\rceil \geq 1$*. So, each one of the* s_{iz}^k *in the left hand side of (7) is included in one of the* $\bar{s_j^h}(\theta)$ *in the right hand side of (7).* $\therefore$ *(7) holds if*

$$\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i} \delta_i^k(T_i) len(s_i^k)}{T_i} \leq \sum_{\forall \tau_i} \frac{\left(\sum_{\forall \tau_j \in \gamma_i} \sum_{\bar{s_j^h}(\theta),\, \theta \in \theta_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil len\left(s_{max}^j(\theta) \right) \right) \right)}{T_i} \tag{8}$$

For each $s_i^k \in s_i$*, there are a set of zero or more* $\bar{s_j^h}(\theta) \in \tau_j, \forall \tau_j \neq \tau_i$ *that are conflicting with* s_i^k*. Assuming this set of conflicting transactions with* s_i^k *is denoted as* $\eta_i^k = \left\{ \bar{s_j^h}(\theta) \in \tau_j : (\theta \in \theta_i) \wedge (\forall \tau_j \neq \tau_i) \wedge \left(\bar{s_j^h}(\theta) \notin \eta_i^l,\, l \neq k \right) \right\}$*. The last condition* $\bar{s_j^h}(\theta) \notin \eta_i^l,\, l \neq k$ *in defintion of* η_i^k *ensures that common transactions* $\bar{s_j^h}$ *that can conflict with more than one transaction* $s_i^k \in \tau_i$ *are split among different* η_i^k*,* $k = 1, .., |s_i|$*. This condition is necessary because in ECM, no two or more transactions of* τ_i^x *can be aborted by the same transaction of* τ_j^h *where* $p_j^h > p_i^x$*. By substitution of* η_i^k *in (8), we get*

$$\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i} \delta_i^k(T_i) len(s_i^k)}{T_i} \leq \sum_{\forall \tau_i} \frac{\left(\sum_{\forall k=1}^{|s_i|} \sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil len\left(s_{max}^j(\theta) \right) \right) \right)}{T_i} \tag{9}$$

$\because$ *len* $(s_{max}^j(\theta)) \geq len(s_i^k)$*,* $\therefore$ *(9) holds if for each* $s_i^k \in \tau_i$

$$\delta_i^k \leq \frac{\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil len\left(s_i^k\right) \right)}{len(s_i^k)} = \sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left\lceil \frac{T_i}{T_j} \right\rceil$$

So, $\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left\lceil \frac{T_i}{T_j} \right\rceil$ *is the maximum number of conflicting transactions with* s_i^k *in all jobs with higher priority than priority of current job of* τ_i*. Claim follows.*

0.5.2 FBLT vs. RCM

Claim 5 *Schedulability of FBLT is equal or better to RCM's when maximum number of abort times of each transaction* s_i^k *in any job of* τ_i *is less or equal to number of conflicting transactions with* s_i^k *in all jobs with higher priority than* τ_i *minus sum of maximum length* $m-1$ *transactions conflicting with* s_i^k*.*

Proof 5 *By substituing $RC_A(T_i)$ and $RC_B(T_i)$ in (4) with (1) and (6.9) in [2] respectively.*

$$\begin{aligned} &\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i} \left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k) \right) + RC_{re}(T_i)}{T_i} \\ \leq \quad &\sum_{\forall \tau_i} \frac{\left(\sum_{\forall \tau_j^* \in \gamma_i^{ex}} \sum_{\forall \theta \in \theta_i^{ex}} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) \sum_{\forall \bar{s_j^h}(\theta)} len\left(\bar{s_j^h}(\theta) + s_{max}^j(\theta) \right) \right) + RC_{re}(T_i)}{T_i} \end{aligned} \tag{10}$$

where $\tau_j^* = \{\tau_j : (\tau_j \neq \tau_i) \wedge (p_j > p_i)\}$. Let $\theta_i^{ex} = \theta_i + \theta_i^*$ where θ_i^* is the set of objects not accessed directly by τ_i but can enforce transactions in τ_i to retry due to transitive retry. Let $\gamma_i^{ex} = \gamma_i + \gamma_i^*$where γ_i^* is the set of tasks that access objects in θ_i^*. $\bar{s_j^h}(\theta)$ can access multiple objects, so $s_{max}^j(\theta)$ is the maximum length transaction conflicting with $\bar{s_j^h}(\theta)$. FBLT needs to bound transitive retry. So, θ_i^{ex} will be avoided in (10). Thus, transactions under RCM behave as if there were no transitive retry. Consequently, (10) will be

$$\begin{aligned} &\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i} \left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k) \right)}{T_i} \\ \leq \quad &\sum_{\forall \tau_i} \frac{\sum_{\forall \tau_j^* \in \gamma_i} \sum_{\forall \theta \in \theta_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) \sum_{\forall \bar{s_j^h}(\theta)} len\left(\bar{s_j^h}(\theta) + s_{max}^j(\theta) \right)}{T_i} \end{aligned} \tag{11}$$

As $\bar{s_j^h}(\theta)$ is included only once for all objects accessed by it, $s_{max}^j(\theta)$ is also included once for each $\bar{s_j^h}(\theta)$. Consequently, (11) becomes

$$\begin{aligned} &\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i} \left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k) \right)}{T_i} \\ \leq \quad &\sum_{\forall \tau_i} \frac{\left(\sum_{\forall \tau_j^* \in \gamma_i} \sum_{\bar{s_j^h}(\theta),\, \theta \in \theta_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) len\left(\bar{s_j^h}(\theta) + s_{max}^j(\theta) \right) \right)}{T_i} \end{aligned} \tag{12}$$

Although different s_i^k can have common conflicting transactions $\bar{s_j^h}$, but no more than one s_i^k can be preceeded by the same $\bar{s_j^h}$ in the m_set. This happens because transactions in the m_set are non-preemptive. Original priority of transactions preeceding s_i^k in the m_set can be of lower or higher priority than original priority of s_i^k. Under G-RMA, $p_j > p_i$ means that $T_j \leq T_i$, then $\left\lceil \frac{T_i}{T_j} \right\rceil \geq 1$. For each $s_i^k \in s_i$, there are a set of zero or more $\bar{s_j^h}(\theta) \in \tau_j^*$ that are conflicting with s_i^k. Assuming this set of conflicting transactions with s_i^k is denoted as $\eta_i^k = \left\{ \bar{s_j^h}(\theta) \in \tau_j^* : (\theta \in \theta_i) \wedge \left(\bar{s_j^h}(\theta) \notin \eta_i^l,\, l \neq k \right) \right\}$. The last condition $\bar{s_j^h}(\theta) \notin \eta_i^l$, $l \neq k$ in defintion of η_i^k ensures that common transactions $\bar{s_j^h}$ that can conflict with more than one transaction $s_i^k \in \tau_i$ are split among different η_i^k, $k = 1,..,|s_i|$. This condition is necessary because in RCM, no two or more transactions of τ_i^x can be aborted by the same transaction *of τ_j^h where $p_j^h > p_i^x$. By substitution of η_i^k in (12), we get*

$$\begin{aligned} & \sum\nolimits_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i} \left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k) \right)}{T_i} \\ \leq & \sum\nolimits_{\forall \tau_i} \frac{\left(\sum_{\forall k=1}^{|s_i|} \sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) len\left(\bar{s_j^h}(\theta) + s_{max}^j(\theta) \right) \right) \right)}{T_i} \end{aligned} \tag{13}$$

$\bar{s_j^h}$ belongs to higher priority jobs than τ_i and s_{max}^j belongs to higher priority jobs than τ_i or τ_i itself. Transactions in m_set can belong to jobs with original priority higher or lower than τ_i. So, (12) holds if for each $s_i^k \in \tau_i$

$$\delta_i^k(T_i) len(s_i^k) \leq \left(\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) len\left(\bar{s_j^h}(\theta) + s_{max}^j(\theta) \right) \right) \right) - \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k) \tag{14}$$

$$\therefore\ \delta_i^k(T_i) \leq \left(\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) len\left(\frac{\bar{s_j^h}(\theta) + s_{max}^j(\theta)}{s_i^k} \right) \right) \right) - \sum_{s_{iz}^k \in \chi_i^k} len\left(\frac{s_{iz}^k}{s_i^k} \right) \tag{15}$$

$\because\ len\left(\frac{s_{iz}^k}{s_i^k} \right) \leq len(s_{iz}^k)$, and $len(s_{max}^j(\theta)) > len(s_i^k)$ $\therefore$ (15) holds if

$$\therefore\ \delta_i^k(T_i) \leq \left(\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) \right) - \sum_{s_{iz}^k \in \chi_i^k} len\left(s_{iz}^k \right)$$

$\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right)$ represents number of conflicting transactions with s_i^k in all jobs with higher priority than τ_i. $\sum_{s_{iz}^k \in \chi_i^k} len\left(s_{iz}^k \right)$ is sum of maximum $m-1$ transactional length transactions conflicting with s_i^k. Claim follows.

0.5.3 FBLT vs. G-EDF/LCM

Claim 6 *Schedulability of FBLT is equal or better to G-EDF/LCM's when maximum number of abort times of each transaction s_i^k in any job of τ_i is less or equal to sum of total number each transaction s_j^h can conflict with s_i^k multiplied by maximum α with which s_j^h can conflict with maximum length transaction sharing objects with s_i^k and s_j^h.*

Proof 6 *By substituting $RC_A(T_i)$ and $RC_B(T_i)$ in (4) with (1) and (6.7) in [2] respectively.*

$$\begin{aligned}
&\sum_{\forall\tau_i}\frac{\sum_{\forall s_i^k\in s_i}\left(\delta_i^k(T_i)len(s_i^k)+\sum_{\forall s_{iz}^k\in\chi_i^k}len(s_{iz}^k)\right)+RC_{re}(T_i)}{T_i} \\
\leq\ &\sum_{\forall\tau_i}\frac{\left(\sum_{\forall\tau_j\in\gamma_i^{ex}}\sum_{\theta\in\theta_i^{ex}}\left(\left\lceil\frac{T_i}{T_j}\right\rceil\sum_{\forall \bar{s_j^h}(\theta)}len\left(\bar{s_j^h}(\theta)+\alpha_{max}^{j\bar{h}}s_{max}^j(\theta)\right)\right)\right)}{T_i} \\
+\ &\sum_{\forall\tau_i}\frac{\left(\sum_{\forall s_i^k}\left(1-\alpha_{max}^{ik}\right)len\left(s_{max}^i\right)\right)+RC_{re}(T_i)}{T_i}
\end{aligned}\tag{16}$$

Let $\theta_i^{ex}=\theta_i+\theta_i^*$ where θ_i^* is the set of objects not accessed directly by τ_i but can enforce transactions in τ_i to retry due to transitive retry. Let $\gamma_i^{ex}=\gamma_i+\gamma_i^*$where γ_i^* is the set of tasks that access objects in θ_i^*. $\bar{s_j^h}(\theta)$ can access multiple objects, so $s_{max}^j(\theta)$ is the maximum length transaction conflicting with $\bar{s_j^h}(\theta)$. FBLT needs to bound transitive retry. So, θ_i^{ex} will be avoided in (16). Thus, transactions under G-EDF/LCM behave as if there were no transitive retry. Consequently, (16) will be

$$\begin{aligned}
&\sum_{\forall\tau_i}\frac{\sum_{\forall s_i^k\in s_i}\left(\delta_i^k(T_i)len(s_i^k)+\sum_{\forall s_{iz}^k\in\chi_i^k}len(s_{iz}^k)\right)}{T_i} \\
\leq\ &\sum_{\forall\tau_i}\frac{\left(\sum_{\forall\tau_j\in\gamma_i}\sum_{\theta\in\theta_i}\left(\left\lceil\frac{T_i}{T_j}\right\rceil\sum_{\forall \bar{s_j^h}(\theta)}len\left(\bar{s_j^h}(\theta)+\alpha_{max}^{j\bar{h}}s_{max}^j(\theta)\right)\right)\right)}{T_i} \\
+\ &\sum_{\forall\tau_i}\frac{\left(\sum_{\forall s_i^k}\left(1-\alpha_{max}^{ik}\right)len\left(s_{max}^i\right)\right)}{T_i}
\end{aligned}\tag{17}$$

As $\bar{s_j^h}(\theta)$ is included only once for all objects accessed by it. $s_{max}^j(\theta)$ is also included once for each $\bar{s_j^h}(\theta)$. Consequently, (17) becomes

$$\begin{aligned}
&\sum_{\forall\tau_i}\frac{\sum_{\forall s_i^k\in s_i}\left(\delta_i^k(T_i)len(s_i^k)+\sum_{\forall s_{iz}^k\in\chi_i^k}len(s_{iz}^k)\right)}{T_i} \\
\leq\ &\sum_{\forall\tau_i}\frac{\left(\sum_{\forall\tau_j\in\gamma_i}\sum_{\forall \bar{s_j^h}(\theta),\theta\in\theta_i}\left(\left\lceil\frac{T_i}{T_j}\right\rceil len\left(\bar{s_j^h}(\theta)+\alpha_{max}^{j\bar{h}}s_{max}^j(\theta)\right)\right)\right)}{T_i} \\
+\ &\sum_{\forall\tau_i}\frac{\left(\sum_{\forall s_i^k}\left(1-\alpha_{max}^{ik}\right)len\left(s_{max}^i\right)\right)}{T_i}
\end{aligned}\tag{18}$$

Although different s_i^k can have common conflicting transactions $\bar{s_j^h}$, but no more than one s_i^k can be preceded by the same $\bar{s_j^h}$ in the m_set. This happens because transactions in the m_set are non-preemptive. Original priority of transactions preceding s_i^k in the m_set can be of lower or higher priority than original priority of s_i^k. Under G-EDF/LCM, $\tau_j\neq\tau_i$ can have at least one job of higher priority than current job of τ_i, then $\left\lceil\frac{T_i}{T_j}\right\rceil\geq 1$. So, each one of the s_{iz}^k in the left hand side of (18) *is included in one of the $\bar{s_j^h}(\theta)$ in the right hand side* *of* (18). $\therefore$ (18) *holds if*

$$\begin{aligned}
&\sum_{\forall\tau_i} \frac{\sum_{\forall s_i^k \in s_i} \delta_i^k(T_i) len(s_i^k)}{T_i} \\
\leq\ & \sum_{\forall\tau_i} \frac{\left(\sum_{\forall\tau_j \in \gamma_i} \sum_{\forall \bar{s_j^h}(\theta), \theta \in \theta_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil len\left(\alpha_{max}^{j\bar{h}} s_{max}^j(\theta)\right)\right)\right)}{T_i} \\
+\ & \sum_{\forall\tau_i} \frac{\sum_{\forall s_i^k} \left(1-\alpha_{max}^{ik}\right) len\left(s_{max}^i\right)}{T_i}
\end{aligned} \tag{19}$$

For each $s_i^k \in s_i$*, there are a set of zero or more* $\bar{s_j^h}(\theta) \in \tau_j, \forall\tau_j \neq \tau_i$ *that are conflicting with* s_i^k*. Assuming this set of conflicting transactions with* s_i^k *is denoted as* $\eta_i^k = \left\{\bar{s_j^h}(\theta) \in \tau_j : (\theta \in \theta_i) \wedge (\forall\tau_j \neq \tau_i) \wedge \left(\bar{s_j^h}(\theta) \notin \eta_i^l,\, l \neq k\right)\right\}$*. The last condition* $\bar{s_j^h}(\theta) \notin \eta_i^l$, $l \neq k$ *in definition of* η_i^k *ensures that common transactions* $\bar{s_j^h}$ *that can conflict with more than one transaction* $s_i^k \in \tau_i$ *are split among different* η_i^k, $k = 1, .., |s_i|$*. This condition is necessary because in G-EDF/LCM, no two or more transactions of* τ_i^x *can be aborted by the same transaction of* τ_j^h *where* $p_j^h > p_i^x$*. By substitution of* η_i^k *in (19), we get*

$$\begin{aligned}
&\sum_{\forall\tau_i} \frac{\sum_{\forall s_i^k \in s_i} \delta_i^k(T_i) len(s_i^k)}{T_i} \\
\leq\ & \sum_{\forall\tau_i} \frac{\sum_{k=1}^{|s_i|} \sum_{\forall \bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil len\left(\alpha_{max}^{j\bar{h}} s_{max}^j(\theta)\right)\right)}{T_i} \\
+\ & \sum_{\forall\tau_i} \frac{\left(\sum_{\forall s_i^k} \left(1-\alpha_{max}^{ik}\right) len\left(s_{max}^i\right)\right)}{T_i}
\end{aligned} \tag{20}$$

$\bar{s_j^h}$ *belongs to higher priority jobs than* τ_i *and* s_{max}^j *belongs to higher priority jobs than* τ_i *or* τ_i *itself. Transactions in m_set can belong to jobs with original priority higher or lower than* τ_i*. So, (18) holds if for each* $s_i^k \in \tau_i$

$$\delta_i^k(T_i) len(s_i^k) \leq \left(\sum_{\forall \bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil\right) len\left(\alpha_{max}^{j\bar{h}} s_{max}^j(\theta)\right)\right) + \left(1 - \alpha_{max}^{ik}\right) len\left(s_{max}^i\right)$$

$$\therefore\ \delta_i^k(T_i) \leq \left(\sum_{\forall \bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil\right) len\left(\frac{\alpha_{max}^{j\bar{h}} s_{max}^j(\theta)}{s_i^k}\right)\right) + \left(1 - \alpha_{max}^{ik}\right) len\left(\frac{s_{max}^i}{s_i^k}\right) \tag{21}$$

$\because\ len\left(\frac{s_{max}^j(\theta)}{s_i^k}\right) \geq 1$, $\therefore$ *(21) holds if*

$$\therefore\ \delta_i^k(T_i) \leq \left(\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil \alpha_{max}^{j\bar{h}}\right)\right)$$

Claim follows.

0.5.4 FBLT vs. G-RMA/LCM

Claim 7 *Schedulability of FBLT is equal or better to G-RMA/LCM's when maximum number of abort times of each transaction s_i^k in any job of τ_i is less or equal to sum of maximum length $m-1$ transactions conflicting with s_i^k subtracted from sum of total number each transaction s_j^h can conflict with s_i^k multiplied by maximum α with which s_j^h can conflict with maximum length transaction sharing objects with s_i^k and s_j^h.*

Proof 7 *By substituing $RC_A(T_i)$ and $RC_B(T_i)$ in (4) with (1) and (6.9) in [2] respectively.*

$$
\begin{aligned}
&\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k)\right) + RC_{re}(T_i)}{T_i} \\
\leq \quad & \sum_{\forall \tau_i} \frac{\sum_{\forall \tau_j^* \in \gamma_i^{ex}} \sum_{\forall \theta \in \theta_i^{ex}} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1\right) \sum_{\forall \bar{s_j^h}(\theta)} len\left(\bar{s_j^h}(\theta) + \alpha_{max}^{j\bar{h}} s_{max}^j(\theta)\right)}{T_i} \\
+ \quad & \sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k} \left(1-\alpha_{max}^{ik}\right) len(s_{max}^i) + RC_{re}(T_i)}{T_i}
\end{aligned}
\tag{22}
$$

where $\tau_j^* = \{\tau_j : p_j > p_i\}$. Let $\theta_i^{ex} = \theta_i + \theta_i^*$ where θ_i^* is the set of objects not accessed directly by τ_i but can enforce transactions in τ_i to retry due to transitive retry. Let $\gamma_i^{ex} = \gamma_i + \gamma_i^*$ where γ_i^* is the set of tasks that access objects in θ_i^*. $\bar{s_j^h}(\theta)$ can access multiple objects, so $s_{max}^j(\theta)$ is the maximum length transaction conflicting with $\bar{s_j^h}(\theta)$. FBLT needs to bound transitive retry. So, θ_i^{ex} will be avoided in (22). Thus, transactions under G-RMA/LCM behave as if there were no transitive retry. Consequently, (22) will be

$$
\begin{aligned}
&\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k)\right)}{T_i} \\
\leq \quad & \sum_{\forall \tau_i} \frac{\sum_{\forall \tau_j^* \in \gamma_i} \sum_{\forall \theta \in \theta_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1\right) \sum_{\forall \bar{s_j^h}(\theta)} len\left(\bar{s_j^h}(\theta) + \alpha_{max}^{j\bar{h}} s_{max}^j(\theta)\right)}{T_i} \\
+ \quad & \sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k} \left(1-\alpha_{max}^{ik}\right) len(s_{max}^i)}{T_i}
\end{aligned}
\tag{23}
$$

As $\bar{s_j^h}(\theta)$ is included only once for all objects accessed by it. $s_{max}^j(\theta)$ is also included once for each $\bar{s_j^h}(\theta)$. Consequently, (23) becomes

$$
\begin{aligned}
&\sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k)\right)}{T_i} \\
\leq \quad & \sum_{\forall \tau_i} \frac{\left(\sum_{\forall \tau_j^* \in \gamma_i} \sum_{\bar{s_j^h}(\theta),\, \theta \in \theta_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1\right) len\left(\bar{s_j^h}(\theta) + \alpha_{max}^{j\bar{h}} s_{max}^j(\theta)\right)\right)}{T_i} \\
+ \quad & \sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k} \left(1-\alpha_{max}^{ik}\right) len(s_{max}^i)}{T_i}
\end{aligned}
\tag{24}
$$

Although different s_i^k can have common conflicting transactions $\bar{s_j^h}$, but no more than one s_i^k can be preceded by the same $\bar{s_j^h}$ in the m_set. This happens because transactions in the m_set are non-preemptive. Original priority of transactions preceding s_i^k in the m_set can be of lower or higher priority than original priority of s_i^k. Under G-RMA, $p_j > p_i$ means that $T_j \leq T_i$, then $\left\lceil \frac{T_i}{T_j} \right\rceil \geq 1$. For each $s_i^k \in s_i$, there are a set of zero or more $\bar{s_j^h}(\theta) \in \tau_j^*$ that are conflicting with s_i^k. Assuming this set of conflicting transactions with s_i^k is denoted as $\eta_i^k = \left\{ \bar{s_j^h}(\theta) \in \tau_j^* : (\theta \in \theta_i) \wedge \left(\bar{s_j^h}(\theta) \notin \eta_i^l,\, l \neq k \right) \right\}$. The last condition $\bar{s_j^h}(\theta) \notin \eta_i^l,\, l \neq k$ in definition of η_i^k ensures that common transactions $\bar{s_j^h}$ that can conflict with more than one transaction $s_i^k \in \tau_i$ are split among different η_i^k, $k = 1,..,|s_i|$. This condition is necessary because in G-RMA/LCM, no two or more transactions of τ_i^x can be aborted by the same transaction of τ_j^h where $p_j^h > p_i^x$. By substitution of η_i^k in (24), we get

$$\begin{aligned} & \sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k \in s_i} \left(\delta_i^k(T_i) len(s_i^k) + \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k) \right)}{T_i} \\ \leq \quad & \sum_{\forall \tau_i} \frac{\left(\sum_{\forall k=1}^{|s_i|} \sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) len\left(\bar{s_j^h}(\theta) + \alpha_{max}^{j\bar{h}} s_{max}^j(\theta) \right) \right)}{T_i} \\ + \quad & \sum_{\forall \tau_i} \frac{\sum_{\forall s_i^k} \left(1 - \alpha_{max}^{ik} \right) len(s_{max}^i)}{T_i} \end{aligned} \tag{25}$$

$\bar{s_j^h}$ belongs to higher priority jobs than τ_i and s_{max}^j belongs to higher priority jobs than τ_i or τ_i itself. Transactions in m_set can belong to jobs with original priority higher or lower than τ_i. So, (25) holds if for each $s_i^k \in \tau_i$

$$\begin{aligned} \delta_i^k(T_i) len(s_i^k) \quad \leq \quad & \left(\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) len\left(\bar{s_j^h}(\theta) + \alpha_{max}^{j\bar{h}} s_{max}^j(\theta) \right) \right) \right) \\ + \quad & \left(1 - \alpha_{max}^{ik} \right) len(s_{max}^i) - \sum_{s_{iz}^k \in \chi_i^k} len(s_{iz}^k) \end{aligned}$$

$$\begin{aligned} \therefore\ \delta_i^k(T_i) \quad \leq \quad & \left(\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) len\left(\frac{\bar{s_j^h}(\theta) + \alpha_{max}^{j\bar{h}} s_{max}^j(\theta)}{s_i^k} \right) \right) \right) \\ + \quad & \left(1 - \alpha_{max}^{ik} \right) len(\frac{s_{max}^i}{s_i^k}) - \sum_{s_{iz}^k \in \chi_i^k} len\left(\frac{s_{iz}^k}{s_i^k} \right) \end{aligned} \tag{26}$$

$\because len\left(\frac{s_{iz}^k}{s_i^k} \right) \leq len(s_{iz}^k)$, $\alpha_{max}^{j\bar{h}} \leq 1$ and s_{max}^i can be greater, less or equal to s_i^k $\therefore$ (26) holds if

$$\therefore\ \delta_i^k(T_i) \leq \left(\sum_{\bar{s_j^h}(\theta) \in \eta_i^k} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) \alpha_{max}^{j\bar{h}} len\left(\frac{\bar{s_j^h}(\theta) + s_{max}^j(\theta)}{s_i^k} \right) \right) - \sum_{s_{iz}^k \in \chi_i^k} len\left(s_{iz}^k \right) \tag{27}$$

$\because\ len(\frac{s^j_{max}(\theta)}{s^k_i}) \geq 1$, $\therefore$ *(27) holds if*

$$\therefore\ \delta^k_i(T_i) \leq \left(\sum_{s^{\bar{h}}_j(\theta)\in\eta^k_i} \left(\left\lceil \frac{T_i}{T_j} \right\rceil + 1 \right) \alpha^{j\bar{h}}_{max} \right) - \sum_{s^k_{iz}\in\chi^k_i} len\left(s^k_{iz}\right)$$

Claim follows.

0.5.5 FBLT vs. PNF

Claim 8 *Schedulability of FBLT is equal or better to PNF's with G-EDF and G-RMA if for any τ^x_i the following conditions are satisfied:*

1. *Total sum of transactional lengths of all transactions conflicting with the maximum length transaction in τ^x_i, $s_{i_{max}}$, is no less than maximum retry cost of τ^x_i due to release of higher priority jobs.*
2. *Maximum abort number of $s_{i_{max}}$ is less or equal to difference between total sum of transactional lengths of all transactions conflicting with $s_{i_{max}}$ and maximum retry cost of τ^x_i due to release of higher priority jobs divided by length of $s_{i_{max}}$.*
3. *Maximum abort number of any s^k_i, other than $s_{i_{max}}$, should be less or equal to total sum of transactional lengths of all conflicting transactions with s^k_i divided by length of s^k_i.*

Proof 8 *By substituting $RC_A(T_i)$ and $RC_B(T_i)$ in (4) with (1) and (6.1) in [2] respectively.*

$$\begin{aligned} & \sum\nolimits_{\forall\tau_i} \frac{\sum_{\forall s^k_i\in s_i}\left(\delta^k_i(T_i)len(s^k_i)+\sum_{s^k_{iz}\in\chi^k_i} len(s^k_{iz})\right)+RC_{re}(T_i)}{T_i} \\ \leq \quad & \sum\nolimits_{\forall\tau_i} \frac{\sum_{\forall\tau_j\in\gamma_i}\sum_{\theta\in\theta_i}\left(\left(\left\lceil\frac{T_i}{T_j}\right\rceil+1\right)\sum_{\forall s^{\bar{h}}_j(\theta)} len\left(s^{\bar{h}}_j(\theta)\right)\right)}{T_i} \end{aligned} \tag{28}$$

$s^{\bar{h}}_j(\theta)$ can access multiple objects, so $s^j_{max}(\theta)$ is the maximum length transaction conflicting with $s^{\bar{h}}_j(\theta)$. As $s^{\bar{h}}_j(\theta)$ is included only once for all objects accessed by it. $s^j_{max}(\theta)$ is also included once for each $s^{\bar{h}}_j(\theta)$. Consequently, 28 becomes

$$\begin{aligned} & \sum\nolimits_{\forall\tau_i} \frac{\sum_{\forall s^k_i\in s_i}\left(\delta^k_i(T_i)len(s^k_i)+\sum_{s^k_{iz}\in\chi^k_i} len(s^k_{iz})\right)+RC_{re}(T_i)}{T_i} \\ \leq \quad & \sum\nolimits_{\forall\tau_i} \frac{\sum_{\forall\tau_j\in\gamma_i}\sum_{s^{\bar{h}}_j(\theta),\,\theta\in\theta_i}\left(\left(\left\lceil\frac{T_i}{T_j}\right\rceil+1\right) len\left(s^{\bar{h}}_j(\theta)\right)\right)}{T_i} \end{aligned} \tag{29}$$

$RC_{re}(T_i)$ is given by (6.8) in [2] in case of G-EDF, and (6.10) in [2] in case of G-RMA. Substituting $RC_{re}(T_i) = \sum_{\forall\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil s_{i_{max}}$, covers $RC_{re}(T_i)$ given by (6.8) and (6.10) in [2] and maintains validity of 29. If τ_j has no shared objects with τ_i, then release of any higher priority job τ_j^y will not abort any transaction in any job of τ_i. $\therefore$ 29 holds if

$$\sum_{\forall\tau_i}\frac{\sum_{\forall s_i^k\in s_i}\left(\delta_i^k(T_i)len(s_i^k)+\sum_{s_{iz}^k\in\chi_i^k}len(s_{iz}^k)\right)+\left(\sum_{\forall\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil s_{i_{max}}\right)}{T_i} \tag{30}$$
$$\leq \sum_{\forall\tau_i}\frac{\left(\sum_{\forall\tau_j\in\gamma_i}\sum_{\bar{s_j^h}(\theta),\,\theta\in\theta_i}\left(\left(\left\lceil\frac{T_i}{T_j}\right\rceil+1\right)len\left(\bar{s_j^h}(\theta)\right)\right)\right)}{T_i}$$

Although different s_i^k can have common conflicting transactions $\bar{s_j^h}$, but no more than one s_i^k can be preceded by the same $\bar{s_j^h}$ in the m_set. This happens because transactions in the m_set are non-preemptive. Original priority of transactions preceding s_i^k in the m_set can be of lower or higher priority than original priority of s_i^k. Under PNF, τ_j^y can have a priority higher or lower priority than τ_i^x, still transactions in τ_j^y can abort transactions in τ_i^x. So, each one of the s_{iz}^k in the left hand side of 30 is included in one of the $\bar{s_j^h}(\theta)$ in the right hand side of 30. $\therefore$ 30 holds if

$$\sum_{\forall\tau_i}\frac{\sum_{\forall s_i^k\in s_i}\left(\delta_i^k(T_i)len(s_i^k)\right)+\left(\sum_{\forall\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil s_{i_{max}}\right)}{T_i} \tag{31}$$
$$\leq \sum_{\forall\tau_i}\frac{\left(\sum_{\forall\tau_j\in\gamma_i}\sum_{\bar{s_j^h}(\theta),\,\theta\in\theta_i}\left(\left\lceil\frac{T_i}{T_j}\right\rceil len\left(\bar{s_j^h}(\theta)\right)\right)\right)}{T_i}$$

One of the s_i^k is $s_{i_{max}}$, so 31 becomes

$$\sum_{\forall\tau_i}\frac{\sum_{\forall s_i^k\in s_i,\,s_i^k\neq s_{i_{max}}}\left(\delta_i^k(T_i)len(s_i^k)\right)+\left(\left(\delta_{i_{max}}+\sum_{\forall\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil\right)s_{i_{max}}\right)}{T_i} \tag{32}$$
$$\leq \sum_{\forall\tau_i}\frac{\left(\sum_{\forall\tau_j\in\gamma_i}\sum_{\bar{s_j^h}(\theta),\,\theta\in\theta_i}\left(\left\lceil\frac{T_i}{T_j}\right\rceil len\left(\bar{s_j^h}(\theta)\right)\right)\right)}{T_i}$$

For each $s_i^k\in s_i$ including $s_{i_{max}}$, there are a set of one or more $\bar{s_j^h}(\theta)\in\tau_j$, $\forall\tau_j\in\gamma_i$ that are conflicting with s_i^k. Assuming this set of conflicting transactions with s_i^k is denoted as $\eta_i^k = \left\{\bar{s_j^h}(\theta)\in\tau_j : (\theta\in\theta_i)\wedge(\forall\tau_j\in\gamma_i)\wedge\left(\bar{s_j^h}(\theta)\notin\eta_i^l,\, l\neq k\right)\right\}$. The last condition $\bar{s_j^h}(\theta)\notin\eta_i^l$, $l\neq k$ in definition of η_i^k ensures that common transactions $\bar{s_j^h}$ that can conflict with more than one transaction $s_i^k\in\tau_i$ are split among different η_i^k, $k=1,..,|s_i|$. This condition is necessary because in PNF, no two or more transactions of τ_i^x can be aborted by the same transaction of τ_j^h whether $p_j^h > p_i^x$ or not. By substitution of η_i^k in 31, we get

$$\sum_{\forall\tau_i}\frac{\left(\sum_{\forall s_i^k\in s_i,\,s_i^k\neq s_{i_{max}}}\left(\delta_i^k(T_i)len(s_i^k)\right)\right)+\left(\left(\delta_{i_{max}}(T_i)+\sum_{\forall\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil\right)s_{i_{max}}\right)}{T_i} \tag{33}$$
$$\leq \sum_{\forall\tau_i}\frac{\left(\sum_{\forall k=1}^{|s_i|}\sum_{\bar{s_j^h}(\theta)\in\eta_i^k}\left(\left\lceil\frac{T_i}{T_j}\right\rceil len\left(\bar{s_j^h}(\theta)\right)\right)\right)}{T_i}$$

Since $s_{i_{max}} \in s_i$, $\therefore$ (33) holds if the following two conditions hold for each τ_i :

1. $\delta_{i_{max}} \leq \frac{\left(\sum_{\bar{s_j^h}(\theta)\in\eta_{i_{max}}}\left\lceil\frac{T_i}{T_j}\right\rceil len\left(\bar{s_j^h}(\theta)\right)\right)-\left(\sum_{\forall\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil s_{i_{max}}\right)}{s_{i_{max}}}$, *where $\eta_{i_{max}}$ is one of the η_i^ks that corresponds to $s_{i_{max}}$.* $\because$ $\delta_{i_{max}} \geq 0$, $\therefore$ $\left(\sum_{\bar{s_j^h}(\theta)\in\eta_{i_{max}}}\left\lceil\frac{T_i}{T_j}\right\rceil len\left(\bar{s_j^h}(\theta)\right)\right) \geq \left(\sum_{\forall\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil s_{i_{max}}\right)$.

2. *For each s_i^k other than $s_{i_{max}}$,* $\delta_i^k \leq \sum_{\bar{s_j^h}(\theta)\in\eta_i^k}\left(\left\lceil\frac{T_i}{T_j}\right\rceil len\left(\frac{\bar{s_j^h}(\theta)}{s_i^k}\right)\right)$.

Claim follows.

0.5.6 FBLT vs. Lock-free

Claim 9 *Under G-EDF and G-RMA, schedulability of FBLT is equal or better than lock-free's if $s_{max} \leq r_{max}$. If transactions execute in FIFO order (i. e., $\delta_i^k = 0, \forall s_i^k$) and contention is high, s_{max} can be much larger than r_{max}.*

Proof 9 *Lock-free synchronization [1, 3] accesses only one object. Thus, the number of accessed objects per transaction in FBLT is limited to one. This allows us to compare the schedulability of FBLT with the lock-free algorithm.*

By substituting $RC_A(T_i)$ and $RC_B(T_i)$ in (4) with (1) and (6.17) in [2] respectively.

$$\sum_{\forall\tau_i}\frac{\sum_{\forall s_i^k\in s_i}\left(\delta_i^k(T_i)len(s_i^k)+\sum_{s_{iz}^k\in\chi_i^k}len(s_{iz}^k)\right)+RC_{re}(T_i)}{T_i} \leq \sum_{\forall\tau_i}\frac{\left(\sum_{\forall\tau_j\in\gamma_i}\left(\left(\left\lceil\frac{T_i}{T_j}\right\rceil+1\right)\beta_{i,j}r_{max}\right)\right)+RC_{re}(T_i)}{T_i} \tag{34}$$

where $\beta_{i,j}$ is the number of retry loops of τ_j that access the same objects as accessed by any retry loop of τ_i [1]. r_{max} is the maximum execution cost of a single iteration of any retry loop of any task [1]. For G-EDF(G-RMA), any job τ_i^x under FBLT has the same pattern of interference from higher priority jobs as ECM(RCM) respectively. $RC_{re}(T_i)$ for ECM, RCM and lock-free are given by Claims 25, 26 and 27 in [2] respectively. $\therefore$ $RC_{re}(T_i) = \left\lceil\frac{T_i}{T_j}\right\rceil s_{i_{max}}, \forall\tau_j \in \gamma_i$ *covers $RC_{re}(T_i)$ for G-EDF/FBLT and G-RMA/FBLT.* $RC_{re}(T_i) = \left\lceil\frac{T_i}{T_j}\right\rceil r_{i_{max}}, \forall\tau_j \in \gamma_i$ *covers retry cost for G-EDF/lock-free and G-RMA/lock-free.* $\therefore$ *(34) becomes*

$$\sum_{\forall\tau_i}\frac{\sum_{\forall s_i^k\in s_i}\left(\delta_i^k(T_i)len(s_i^k)+\sum_{s_{iz}^k\in\chi_i^k}len(s_{iz}^k)\right)+\sum_{\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil s_{i_{max}}}{T_i} \leq \sum_{\forall\tau_i}\frac{\left(\sum_{\forall\tau_j\in\gamma_i}\left(\left(\left\lceil\frac{T_i}{T_j}\right\rceil+1\right)\beta_{i,j}r_{max}\right)\right)+\sum_{\tau_j\in\gamma_i}\left\lceil\frac{T_i}{T_j}\right\rceil r_{i_{max}}}{T_i} \tag{35}$$

Since $s_{max} \geq s_{i_{max}}$, $len(s_i^k)$, $len(s_{iz}^k)$, $\forall i, z, k$ *and* $r_{max} \geq r_{i_{max}}$ $\therefore$ *(35) holds if*

$$\sum_{\forall \tau_i} \frac{\left(\left(\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i)+\sum_{s_{iz}^k \in \chi_i^k} 1\right)\right)+\sum_{\tau_j \in \gamma_i}\left\lceil \frac{T_i}{T_j} \right\rceil\right) s_{max}}{T_i} \leq \sum_{\forall \tau_i} \frac{\left(\left(\sum_{\forall \tau_j \in \gamma_i}\left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil+1\right)\beta_{i,j}\right)\right)+\sum_{\tau_j \in \gamma_i}\left\lceil \frac{T_i}{T_j} \right\rceil\right) r_{max}}{T_i} \tag{36}$$

$\therefore$ *(36) holds if for each* τ_i

$$\left(\left(\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i)+\sum_{s_{iz}^k \in \chi_i^k} 1\right)\right)+\sum_{\tau_j \in \gamma_i}\left\lceil \frac{T_i}{T_j} \right\rceil\right) s_{max} \leq \left(\left(\sum_{\forall \tau_j \in \gamma_i}\left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil+1\right)\beta_{i,j}\right)\right)+\sum_{\tau_j \in \gamma_i}\left\lceil \frac{T_i}{T_j} \right\rceil\right) r_{max} \tag{37}$$

$\therefore$

$$\frac{s_{max}}{r_{max}} \leq \frac{\left(\sum_{\forall \tau_j \in \gamma_i}\left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil+1\right)\beta_{i,j}\right)\right)+\sum_{\tau_j \in \gamma_i}\left\lceil \frac{T_i}{T_j} \right\rceil}{\left(\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i)+\sum_{s_{iz}^k \in \chi_i^k} 1\right)\right)+\sum_{\tau_j \in \gamma_i}\left\lceil \frac{T_i}{T_j} \right\rceil} \tag{38}$$

It appears from (38) that as δ_i^k, *as well as* $|\chi_i^k|$, *increases, then* s_{max}/r_{max} *decreases. So, to get the lower bound on* s_{max}/r_{max}, *let* $\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i)+\sum_{s_{iz}^k \in \chi_i^k} 1\right)$ *reaches its maximum value. This maximum value is the total number of interfering transactions belonging to any job* τ_j^l, $j \neq i$. *Priority of* τ_j^l *can be higher or lower than current instance of* τ_i. *Beyond this maximum value, higher values for any* δ_i^k *are ineffective as there will be no more transactions to conflict with* s_i^k. $\therefore$ $\sum_{\forall s_i^k \in s_i}\left(\delta_i^k(T_i)+\sum_{s_{iz}^k \in \chi_i^k} 1\right) \leq \sum_{\forall \tau_j \in \gamma_i}\left(\left\lceil \frac{T_i}{T_j} \right\rceil+1\right)$. *Consequently, (38) will be*

$$\frac{s_{max}}{r_{max}} \leq \frac{\left(\sum_{\forall \tau_j \in \gamma_i}\left(\left(\left\lceil \frac{T_i}{T_j} \right\rceil+1\right)\beta_{i,j}\right)\right)+\sum_{\tau_j \in \zeta_i}\left\lfloor \frac{T_i}{T_j} \right\rfloor}{\left(\sum_{\forall \tau_j \in \gamma_i}\left(\left\lceil \frac{T_i}{T_j} \right\rceil+1\right)\right)+\sum_{\tau_j \in \zeta_i}\left\lfloor \frac{T_i}{T_j} \right\rfloor} \tag{39}$$

As we look for the lower bound on $\frac{s_{max}}{r_{max}}$, *let* $\beta_{i,j}$ *assumes its minimum value. So,* $\beta_{i,j} = 1$. $\therefore$ *(39) holds if* $\frac{s_{max}}{r_{max}} \leq 1$.

Let $\delta_i^k(T_i) \to 0$ *in (38). This means transactions approximately execute in their arrival order. Let* $\beta_{i,j} \to \infty$, $\left\lceil \frac{T_i}{T_j} \right\rceil \to \infty$ *in (38) . This means contention is high. Consequently,* $\frac{s_{max}}{r_{max}} \to \infty$. *So, if transactions execute in FIFO order and contention is high,* s_{max} *can be much larger than* r_{max}. *Claim follows.*

Bibliography

[1] U.M.C. Devi, H. Leontyev, and J.H. Anderson. Efficient synchronization under global edf scheduling on multiprocessors. In *18th Euromicro Conference on Real-Time Systems*, pages 10 pp. –84, 0-0 2006.

[2] Mohammed El-Shambakey. *Real-Time Software Transactional Memory: Contention Managers, Time Bounds, and Implementations*. Phd proposal, Virginia Tech, 2012.

[3] M. Elshambakey and B. Ravindran. Stm concurrency control for multicore embedded real-time software: Time bounds and tradeoffs. In *SAC*, 2012.