A Time-Based Dynamic Synchronization Policy for Consolidated Database Systems

Xinxue (Shawn) Qu
Mendoza College of Business, University of Notre Dame,
Notre Dame, IN 46556 U.S.A. {xqu2@nd.edu}

Zhengrui Jiang
School of Business, Nanjing University,
Nanjing 210093 CHINA {zjiang@nju.edu.cn}

Appendix A

Notation Table

T	Finite maintenance horizon time length
$g = 1, 2, \ldots, G$	G types of data errors
$h = 1, 2, \ldots, H$	H types of information queries
$\lambda_{\Gamma,g}$	Poisson arrival rate for type g data error
$\lambda_{Q,h}$	Poisson arrival rate for type h information query
$\Gamma^{g}_{(t,t+1)}$	Amount of type g data errors from epoch t to epoch t+1
$Q^{h}_{(t,t+1)}$	Amount of type h query from epoch t to epoch t+1
Γ^{g}	Accumulated amount of type g data error
S	CDB system state
$\beta_{g\to h}$	Unit staleness cost of type g data error to type h query
a_k	Action take at the kth decision epoch
I	Check interval length
C_U	Fixed synchronization cost
C_d	Business disruption cost
S	System state space
$S_{(t_1,t_2)}$	The change of system state from time t_1 till time t_2
$d_t(S)$	Decision rule at epoch t given system state S
π	Maintenance policy
$T[S,S']$	Transition probability from state S to state S′
d^*	Optimal decision rule

ζ_k	Threshold for System Staleness Cost at decision epoch k
T_p	State transition probability matrix

Appendix B

Important Derivations

Derivation of Expected Interval Staleness Cost $\mathbf{E}\left[\mathbf{C_{(k,k+1)}}\right]$

Denoting the number of type h queries arriving during the time interval by n_h, which follows a Poisson distribution, and their arrival times by t_{n_h}, t_{n_h-1}, …, t_1, respectively, we have

$$E\left[C_{(k,k+1)}\right] = \sum_{h=1}^{H}\sum_{n_h=1}^{\infty}\left\{\int_{kI}^{kI+I} d\left(t_{n_h}\right)\int_{kI}^{t_{n_h}} d\left(t_{n_h-1}\right)\ldots\int_{kI}^{t_2} d(t_1) * \left(\lambda_{Q,h}\right)^{n_h} * e^{-\lambda_{Q,h}*I} * \left[\sum_{i=1}^{n_h}\sum_{g=1}^{G} E_{\Delta\Gamma^g,(kI,t_i)}[f_h(\Delta\Gamma^g)]\right]\right\} \tag{B-0}$$

With a check interval of length I, the time window between the kth decision epoch and the k+1th epoch is $(kI, kI+I)$, k= 0, 1, 2, ….

In time interval $(kI, kI+I)$, we assume that there are n_h information queries for type h query (h=1, 2…, H). Because the arrivals of queries follow independent Poisson distribution, the time between two consecutive arrivals $t_i - t_{i-1}$ follow i.i.d exponential distribution. Thus, based on Assumption 1, the probability that type h query occurs n_h times at time $t_1, t_2, \ldots, t_{n_h} \in (kI, kI+I)$ is:

$$P\left(n_h, t_1, t_2, \ldots, t_{n_h}\right) = \left(\lambda_{Q,h}\right)e^{-\lambda_{Q,h}*t_1} * \left(\lambda_{Q,h}\right)e^{-\lambda_{Q,h}*(t_2-t_1)} * \ldots * \left(\lambda_{Q,h}\right)e^{-\lambda_{Q,h}*\left(t_{n_h}-t_{n_h-1}\right)} * e^{-\lambda_{Q,h}*\left(I-t_{n_h}\right)} = \left(\lambda_{Q,h}\right)^{n_h} * e^{-\lambda_{Q,h}*I} \tag{B-1}$$

During the interval, the set of new data errors is denoted by $S_{(k,k+1)}$, $S_{(k,k+1)} = (\Gamma^1_{(k,k+1)}, \Gamma^2_{(k,k+1)}, \ldots, \Gamma^g_{(k,k+1)}, \ldots, \Gamma^G_{(k,k+1)})$.

The expected interval staleness cost is

$$E\left[C_{(k,k+1)}\right] = \sum_{h=1}^{H}\sum_{n_h=1}^{\infty}\{\int_{kI}^{kI+I} d\left(t_{n_h}\right)\int_{kI}^{t_{n_h}} d\left(t_{n_h-1}\right)\ldots\int_{kI}^{t_2} d(t_1) * (\lambda_{Q,h})^{n_h} * e^{-\lambda_{Q,h}*I} * \left[\sum_{i=1}^{n_h}\sum_{\Delta\Gamma^1=0}^{\infty}\sum_{\Delta\Gamma^2=0}^{\infty}\ldots\sum_{\Delta\Gamma^G=0}^{\infty} f_h(\Delta\Gamma^1, \Delta\Gamma^2, \ldots, \Delta\Gamma^G)\frac{e^{-\lambda_{\Gamma,1}(t_i-kI)}[\lambda_{\Gamma,1}(t_i-kI)]^{\Delta\Gamma^1}}{(\Gamma^1!)} * \ldots * \frac{e^{-\lambda_{\Gamma,G}t_i}[\lambda_{\Gamma,G}(t_i-kI)]^{\Delta\Gamma^G}}{(\Delta\Gamma^G!)}\right]\} \tag{B-2}$$

where t_i is the time when the i^{th} type h query arrives.

Based on Assumption 1, the G types of data errors occur independently, hence the joint probability distribution of $(\Delta\Gamma^1, \Delta\Gamma^2, \ldots, \Delta\Gamma^G)$ is the product of the occurring probabilities of all G types of data error.

Regarding the cost function $f_h(\Delta\Gamma^1, \Delta\Gamma^2, \ldots, \Delta\Gamma^G)$, according to Assumption 2, each data error leads to business losses independently. Then,

$$f_h(\Delta\Gamma^1, \Delta\Gamma^2, \ldots, \Delta\Gamma^G) = f_h(\Delta\Gamma^1) + f_h(\Delta\Gamma^2) + \cdots + f_h(\Delta\Gamma^G) \tag{B-3}$$

Let $t_i' = t_i - kI$, then $\sum_{\Delta\Gamma^1=0}^{\infty}\sum_{\Delta\Gamma^2=0}^{\infty}\ldots\sum_{\Delta\Gamma^G=0}^{\infty} f_h(\Delta\Gamma^1, \Delta\Gamma^2, \ldots, \Delta\Gamma^G)\frac{e^{-\lambda_{\Gamma,1}t_i'}(\lambda_{\Gamma,1}t_i')^{\Delta\Gamma^1}}{(\Delta\Gamma^1!)} * \ldots * \frac{e^{-\lambda_{\Gamma,G}t_i'}(\lambda_{\Gamma,G}t_i')^{\Delta\Gamma^G}}{(\Delta\Gamma^G!)}$ can be decomposed to (under the independent assumption)

$$\sum_{\Delta\Gamma^1=0}^{\infty} f_h(\Delta\Gamma^1)\frac{e^{-\lambda_{\Gamma,1}t_i'}(\lambda_{\Gamma,1}t_i')^{\Delta\Gamma^1}}{(\Delta\Gamma^1!)} + \cdots + \sum_{\Delta\Gamma^G=0}^{\infty} f_h(\Delta\Gamma^G)\frac{e^{-\lambda_{\Gamma,G}t_i'}(\lambda_{\Gamma,G}t_i')^{\Delta\Gamma^G}}{(\Delta\Gamma^G!)} \tag{B-4}$$

Each element in Eq. (B-4) is an expected value derived based on the distribution of $\Delta\Gamma^g$.

The above result can be rewritten as

$$E_{\Delta\Gamma^1}[f_h(\Delta\Gamma^1)] + E_{\Delta\Gamma^2}[f_h(\Delta\Gamma^2)] + \cdots + E_{\Delta\Gamma^G}[f_h(\Delta\Gamma^G)] = \sum_{g=1}^{G} E_{\Delta\Gamma^g,(kI,t_i)}[f_h(\Delta\Gamma^g)] \tag{B-5}$$

This expected cost function depends on the length of the considered time interval. Specifically, during a time interval $(kI,\ t_i)$,

$$\Delta\Gamma^g_{(kI,t_i)} \sim \text{Poisson}\left(\lambda_{\Gamma,g} * (t_i - kI)\right) \tag{B-6}$$

Based on this result,

$$E\left[C_{(k,k+1)}\right] = \sum_{h=1}^{H}\sum_{n_h=1}^{\infty}\{\int_{kI}^{kI+I} d(t_{n_h})\int_{kI}^{t_{n_h}} d(t_{n_h-1}) \ldots \int_{kI}^{t_2} d(t_1) * (\lambda_{Q,h})^{n_h} * e^{-\lambda_{Q,h}*I} * \left[\sum_{i=1}^{n_h}\sum_{g=1}^{G} E_{\Delta\Gamma^g,(t,t_i)}[f_h(\Delta\Gamma^g)]\right]\} \tag{B-7}$$

Previous studies have assumed a linear form for the cost function (Dey 2006), so a special case here is to assume a linear form for $f_h(\Delta\Gamma^g) = \beta_{h,g} * \Delta\Gamma^g$. Then

$$E[f_h(\Delta\Gamma^g)] = \beta_{h,g} * \lambda_{\Gamma,g} * (t_i - kI) \tag{B-8}$$

The linear expression for $E\left[C_{(k,k+1)}\right]$ thus becomes

$$E\left[C_{(k,k+1)}\right] = \sum_{h=1}^{H}\sum_{n_h=1}^{\infty}\{\int_{kI}^{kI+I} d(t_{n_h})\int_{kI}^{t_{n_h}} d(t_{n_h-1}) \ldots \int_{kI}^{t_2} d(t_1) * (\lambda_{Q,h})^{n_h} * e^{-\lambda_{Q,h}*I} * [(\sum_{g=1}^{G}\beta_{h,g} * \lambda_{\Gamma,g})\sum_{i=1}^{n_h}(t_i - kI)]\} \tag{B-9}$$

Since $(\sum_{g=1}^{G}\beta_{h,g} * \lambda_{\Gamma,g})$ is not affected by the outside summation on h and n_h, we have

$$E\left[C_{(k,k+1)}\right] = \sum_{h=1}^{H}(\sum_{g=1}^{G}\beta_{h,g} * \lambda_{\Gamma,g})\sum_{n_h=1}^{\infty}(\lambda_{Q,h})^{n_h} * e^{-\lambda_{Q,h}*I} * \left\{\int_{kI}^{kI+I} d(t_{n_h})\int_{kI}^{t_{n_h}} d(t_{n_h-1}) \ldots \int_{kI}^{t_2} d(t_1) * \sum_{i=1}^{n_h}(t_i - kI)\right\} \tag{B-10}$$

Given that kI is a constant denoting the starting time of the interval, we can denote $t_i' = t_i - kI$.

$$\begin{aligned} E\left[C_{(k,k+1)}\right] &= \sum_{h=1}^{H}(\sum_{g=1}^{G}\beta_{h,g} * \lambda_{\Gamma,g})\sum_{n_h=1}^{\infty}(\lambda_{Q,h})^{n_h} * e^{-\lambda_{Q,h}*I} * \left\{\int_0^{I} d(t_{n_h}' + kI)\int_0^{t_{n_h}'} d(t_{n_h-1}' + kI) \ldots \int_0^{t_2'} d(t_1' + kI) * \sum_{i=1}^{n_h}(t_i')\right\} \\ &= \sum_{h=1}^{H}(\sum_{g=1}^{G}\beta_{h,g} * \lambda_{\Gamma,g})\sum_{n_h=1}^{\infty}(\lambda_{Q,h})^{n_h} * e^{-\lambda_{Q,h}*I} * \left\{\int_0^{I} dt_{n_h}'\int_0^{t_{n_h}'} dt_{n_h-1}' \ldots \int_0^{t_3'} dt_2'\int_0^{t_2'} dt_1' * \sum_{i=1}^{n_h}(t_i')\right\} \end{aligned} \tag{B-11}$$

According to Dey et al. (2006)

$$\int_0^I dt_n \int_0^{t_n} dt_{n-1} \dots \int_0^{t_3} dt_2 \int_0^{t_2} dt_1 * \left(\sum_{i=1}^{n} t_i^m \right) = \frac{I^{m+n}}{(m+1)(n-1)!} \tag{B-12}$$

Using the induction above,

$$\int_0^I dt'_{n_h} \int_0^{t'_{n_h}} dt'_{n_h-1} \dots \int_0^{t'_3} dt'_2 \int_0^{t'_2} dt'_1 * \sum_{i=1}^{n_h} (t'_i) = \frac{I^{n_h+1}}{2(n_h-1)!} \tag{B-13}$$

$$E\left[C_{(t,t+I)}\right] = \sum_{h=1}^{H} \left(\sum_{g=1}^{G} \beta_{h,g} * \lambda_{\Gamma,g} \right) \sum_{n_h=1}^{\infty} \left(\lambda_{\mathrm{Q},h}\right)^{n_h} * e^{-\lambda_{\mathrm{Q},h} * I} * \frac{I^{n_h+1}}{2(n_h-1)!} \tag{B-14}$$

Let $j = n_h - 1$,

$$E\left[C_{(k,k+1)}\right] = \sum_{h=1}^{H} \left(\sum_{g=1}^{G} \beta_{h,g} * \lambda_{\Gamma,g} \right) \sum_{n_h=1}^{\infty} \left(\lambda_{\mathrm{Q},h}\right)^{j+1} * e^{-\lambda_{\mathrm{Q},h} * I} * \frac{I^{j+2}}{2j!} = \frac{1}{2} \sum_{h=1}^{H} \left(\sum_{g=1}^{G} \beta_{g} * \lambda_{\Gamma,g} \right) * \lambda_{\mathrm{Q},h} I^2 \sum_{j=0}^{\infty} \frac{e^{-\lambda_{\mathrm{Q},h} * I} * \left(\lambda_{\mathrm{Q},h} * I\right)^j}{j!} \tag{B-15}$$

Here, $\sum_{j=0}^{\infty} \frac{e^{-\lambda_{\mathrm{Q},h} * I} * \left(\lambda_{\mathrm{Q},h} * I\right)^j}{j!}$ is a summation of the probability of a Poisson $(\lambda_{\mathrm{Q},h} * I)$ distribution, which equals to 1. Therefore,

$$E\left[C_{(k,k+1)}\right] = \frac{1}{2} \sum_{h=1}^{H} \left(\sum_{g=1}^{G} \beta_{h,g} * \lambda_{\Gamma,g} \right) \lambda_{\mathrm{Q},h} I^2 = \frac{I^2}{2} \sum_{h=1}^{H} \left(\sum_{g=1}^{G} \beta_{h,g} * \lambda_{\Gamma,g} \right) \lambda_{\mathrm{Q},h} \tag{B-16}$$

The above expectation of $C_{(k,k+1)}$ is a special case when the cost function is linear with the data errors.

Appendix C

Proofs of Selected Lemmas and Propositions

Proof of Lemma 1

To prove that $V_k(S_k)$ is a non-decreasing function of S_k, we only need to show that given $S_k^1 > S_k^2$, $V_k(S_k^1) \geq V_k(S_k^2)$ will hold.

According to Eq. (5): $V_k(S_k) = \min_{a_k}\{a_k C_U + (1-a_k)E[C(S_k)] + E[V_{k+1}(S_{k+1})]\} + E[C_{(k,k+1)}]$, the total expected interval staleness costs remain the same regardless of adopted maintenance policies. Hence, we consider the following two scenarios:

Scenario 1: If $a_k(S_k^1) = a_k(S_k^2) = 1$,

$$V_k(S_k^1) = V_k(S_k^2) = C_U + E[C_{(k,k+1)}] + \sum_{S_{(k,k+1)} \in \mathbb{S}} p(S_{(k,k+1)}) V_{k+1}[S_{(k,k+1)}] \tag{C-1}$$

Scenario 2: Otherwise, assume π_1 is the optimal policy when the system is in state S_k^1 and π_2 is the optimal policy for a system state S_k^2. Suppose k^U is the first decision epoch with a synchronization for the CDB in state S_k^1 under the optimal policy π_1. If the system in state S_k^2 also follows policy π_1, which also runs the first synchronization operation at k^U. Then from time $k^U + 1$, the expected system costs will be the same in the two scenarios, i.e. $E[V_{k^U+1}^{\pi_1}(S_{k^U+1})] = \sum_{S_{(k,k+1)} \in \mathbb{S}} p(S_{(k,k+1)}) V_{k^U+1}[S_{(k,k+1)}]$. Therefore,

$$V_k(S_k^1) = V_k^{\pi_1}(S_k^1) = E[C(S_k^1)] + E[C(S_{k+1}^1)] + \cdots + E[C(S_{k^U-1}^1)] + C_U + \sum_{\vec{\Delta} \in \mathbb{S}} p(S_{(k,k+1)}) V_{k^U+1}[S_{(k,k+1)}] \tag{C-2}$$

$$V_k^{\pi_1}(S_k^2) = E[C(S_k^2)] + E[C(S_{k+1}^2)] + \cdots + E[C(S_{k^U-1}^2)] + C_U + \sum_{S_{(k,k+1)} \in \mathbb{S}} p(S_{(k,k+1)}) V_{k^U+1}[S_{(k,k+1)}] \tag{C-3}$$

Since $S_k^1 > S_k^2$, we will have $E[C(S_k^1)] > E[C(S_k^2)]$. Without any synchronization from decision epoch k to epoch $k^U - 1$, the data errors and queries in the system follow the same traffic pattern. This means $S_t^1 > S_t^2$ will always hold for $t \in \{k, k+1, \ldots, k^U - 1\}$. Therefore, we have

$$E[C(S_k^1)] + E[C(S_{k+1}^1)] + \cdots + E[C(S_{k^U-1}^1)] > E[C(S_k^2)] + E[C(S_{k+1}^2)] + \cdots + E[C(S_{k^U-1}^2)] \tag{C-4}$$

Hence, $V_k(S_k^1) = V_k^{\pi_1}(S_k^1) > V_k^{\pi_1}(S_k^2)$.

Further, because π_2 is the optimal policy to apply given system in state S_k^2, the following inequality must hold:

$$V_k^{\pi_1}(S_k^2) \geq V_k^{\pi_2}(S_k^2) = V_k(S_k^2) \tag{C-5}$$

Therefore, we conclude: $V_k(S_k^1) > V_k(S_k^2)$.

From the discussions under both Scenarios 1 and 2, we conclude that given $S_k^1 > S_k^2$, $V_k(S_k^1) \geq V_k(S_k^2)$ will always hold. Therefore, $V_k(S_k)$ is a non-decreasing function with S_k.

Proof of Lemma 2

The optimal decision should always lead to a smaller expected system costs at any decision epoch. When the CDB is synchronized, based on the optimal system costs function in Eq. (5), we denote the incurred future system cost as

$$J_k(S_k, a_k = 1) = C_U + E\left[C_{(k,k+1)}\right] + \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right) V_{k+1}(S_{(k,k+1)}) \tag{C-6}$$

In the absence of a synchronization, we denote the incurred future system cost as

$$J_k(S_k, a_k = 0) = E[C(S_k)] + E\left[C_{(k,k+1)}\right] + \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right) V_{k+1}(S_{(k,k+1)} + S_k) \tag{C-7}$$

According to Lemma 1, $V_{k+1}(S_{(k,k+1)} + S_k) \geq V_{k+1}(S_{(k,k+1)})$. Therefore, if $E[C(S_k)] > C_U$, the incurred future system cost without an synchronization$J_k(S_k, a_k = 0)$ will be larger than the incurred future system cost after synchronization: $J_k(S_k, a_k = 1)$. Therefore, it is always optimal to synchronize the CDB when $E[C(S_k)] > C_U$.

Proof of Proposition 1

In Eq. (11), $W(S_k) = E[C(S_k)] + \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right)[V_{k+1}\left(S_k + S_{(k,k+1)}\right) - V_{k+1}\left(S_{(k,k+1)}\right)]$. According to Definition 1, $E[C(S_k)]$ is increasing with S_k. Also by Lemma 1, $V_{k+1}\left(S_k + S_{(k,k+1)}\right)$ is a non-decreasing function with S_k. Hence we can conclude that $W(S_k)$ is monotonically increasing with S_k.

According to the Bellman Eq. (10): $a_k = \arg min\{a_k C_U + (1 - a_k)E[C(S_k)] + E[V_{k+1}(S_{k+1})]\}$, and from Eq. (6) $E[V_{k+1}(S_{k+1})] = \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right) V_{k+1}[S_{(k,k+1)} + (1 - a_k)S_k]$, we have the optimal action a_k at epoch k as

$$a_k = \begin{cases} 1, & E[C(S_k)] + \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right) V_{k+1}[S_{(k,k+1)} + S_k] \geq C_U + \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right) V_{k+1}[S_{(k,k+1)}] \\ 0, & E[C(S_k)] + \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right) V_{k+1}[S_{(k,k+1)} + S_k] < C_U + \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right) V_{k+1}[S_{(k,k+1)}] \end{cases} \tag{C-8}$$

Moving $E_{S_{(k,k+1)}}\left[V_{k+1}\left(S_{(k,k+1)}\right)\right]$ to the left-hand-side of the inequality, the condition becomes the comparison between $W(S_k) = E[C(S_k)] + \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right)[V_{k+1}\left(S_k + S_{(k,k+1)}\right) - V_{k+1}\left(S_{(k,k+1)}\right)]$ and C_U. The optimal action at decision epoch k is not to synchronize the CDB if $W(S_k) < C_U$, and to synchronize the CDB otherwise.

Proof of Lemma 3

According to the control limit policy in Eq. (12) and Proposition 1, as S_k increases, $W(S_k)$ increases from smaller than C_U to larger than C_U. When the value of $W(S_k)$ crosses C_U, the optimal action will change from 0 to 1. Therefore, $a_k(S_k)$ is a non-decreasing function of S_k.

Proof of Lemma 4

Based on the control limit policy (12), at the last decision epoch K, the optimal action is to synchronize whenever $W(S_K) \geq C_U$. In addition, $W(S_K) = E[C(S_K)] + \sum_{\bar{\Delta}} p\left(S_{(k,k+1)}\right)[V_{K+1}\left(S_K + S_{(k,k+1)}\right) - V_{K+1}\left(S_{(k,k+1)}\right)]$.

At the last epoch, $V_{K+1}(\cdot) = 0$, so we have $W(S_K) = E[C(S_K)]$. The optimal action to take should be determined by

$$a_k = \begin{cases} 1, & W(S_K) = E[C(S_K)] \geq C_U \\ 0, & W(S_K) = E[C(S_K)] < C_U \end{cases} \tag{C-9}$$

Following the decision rule in Eq. (13), we have the threshold $\zeta_K = C_U$. Because the threshold ζ_k is the boundary for $E[C(S_k)]$, which makes $W(S_k)$ larger than C_U when $E[C(S_k)]$ crosses ζ_k. Therefore, we have the value of threshold at kth epoch:

$$\zeta_k = C_U - \sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right)[V_{k+1}\left(S_K + S_{(k,k+1)}\right) - V_{k+1}\left(S_{(k,k+1)}\right)] \tag{C-10}$$

Based on Lemma 1, $\sum_{S_{(k,k+1)}} p\left(S_{(k,k+1)}\right)[V_{k+1}\left(S_K + S_{(k,k+1)}\right) - V_{k+1}\left(S_{(k,k+1)}\right)]$ should always be positive, therefore,

$$\zeta_k < C_U = \zeta_K, \text{ for } k\text{=1, 2, …, } K\text{-1} \tag{C-11}$$

Proof of Lemma 5

The expected accumulated data errors between two check points is: $(\lambda_{\Gamma,1}I, \lambda_{\Gamma,2}I, \ldots, \lambda_{\Gamma,G}I)$. In expectation, the expected number of new coming information queries in the next interval is $(\lambda_{\mathrm{Q},1}I, \lambda_{\mathrm{Q},2}I, \ldots, \lambda_{\mathrm{Q},H}I)$.

The expected loss to all those expected arriving queries by the accumulated data errors in one check interval equals

$$E[C(S_I)] = I^2\left[\sum_{h=1}^{H}\lambda_{\mathrm{Q},h}\left(\sum_{g=1}^{G}\beta_{g\to h}\lambda_{\Gamma,g}\right)\right] \tag{C-12}$$

If the data errors accumulated in one check interval will lead to an expected data staleness cost in the next coming interval that is larger than the synchronization cost C_U, in expectation it would be optimal to synchronize the CDB system at every single decision epoch, then there will be no need to discuss the policy in detail.

Therefore, for the upper bound, we have the constraint $E[C(S_I)] \le C_U$, i.e.,

$$I^2\left[\sum_{h=1}^{H}\lambda_{\mathrm{Q},h}\left(\sum_{g=1}^{G}\beta_{g\to h}\lambda_{\Gamma,g}\right)\right] \le C_U \tag{C-13}$$

which is equivalent to

$$I \le \sqrt{C_U\left[\sum_{h=1}^{H}\lambda_{\mathrm{Q},h}\left(\sum_{g=1}^{G}\beta_{g\to h}\lambda_{\Gamma,g}\right)\right]^{-1}} \tag{C-14}$$

Hence, we have the interval upper bound $I^{ub} = \sqrt{C_U\left[\sum_{h=1}^{H}\lambda_{\mathrm{Q},h}\left(\sum_{g=1}^{G}\beta_{g\to h}\lambda_{\Gamma,g}\right)\right]^{-1}}$.

Appendix D

Comparison between TDS Policy and Hybrid Policy

To demonstrate the superiority of the time-based dynamic synchronization (TDS) policy, we also compare the TDS policy against a hybrid policy proposed by Dey et al. (2015) with the following characteristics: It involves two thresholds: a fixed time threshold, I^*, and a staleness cost threshold, S^*, both optimized to be used in combination. Under this hybrid policy, a system refresh is initiated if either the time threshold or the staleness cost threshold is triggered, whichever happens earlier.

Since the TDS policy and the hybrid policy (Dey et al. 2015) use different model parameters and adopt different assumptions, to make the comparison possible, we have to (1) obtain the key model parameters for Dey et al.'s hybrid policy by mapping its model parameters with ours, and (2) downgrade our policy to fit Dey et al.'s model assumptions. More specifically, the key adaptions made to facilitate a meaningful comparison are summarized in Table D1 below.

Table D1. Key Adaptations Made to the TDS Policy

Original Assumptions	Adapted Settings
Multiple types of data errors	Single type of data error
Multiple types of queries	Single type of query
Unit staleness cost for each error-query pair	One generalized random unit staleness cost
Different costs for synchronizations running in business hours and off-business hours	No disruption cost when planned ahead of time or scheduled during off-business hours

Experimental Results

In our simulated experiments, we implemented the hybrid policy developed by Dey et al. (2015) and the downgraded TDS policy based on the revised assumptions described above. The key parameter values are set as followings: patching (data error) arriving rate $\lambda = 3$, mean severity level $\bar{C}_e = 2$, simulated time window $T = 5{,}000$ units of time, and system check interval for TDS policy $I = 1$ (daily). Regarding the normalized patching setup cost (c_s) and normalized business disruption cost (c_d) as defined in Dey et al. (2015), we tried multiple values: $c_s, c_d \in \{5, 10, 15, 20, 25, 30, 35, 40\}$.

In each experiment run, we first select a combination of c_s and c_d values, then compute the optimal TDS policy and the optimal hybrid policy, and subsequently simulate the synchronization operations using the two policies and record their costs.

Table D2 summarizes the percentage of cost saving achieved by the TDS policy compared to the hybrid policy. The results show that the cost saving is consistently positive under all scenarios, ranging from 3.84% to 10.22%. This clearly shows that the TDS policy is a superior method when compared with the hybrid method.

Table D2. Performance Comparison: (Cost of Hybrid – cost of TDS)/Cost of Hybrid

	c_d=5	c_d=10	c_d=15	c_d=20	c_d=25	c_d=30	c_d=35	c_d=40
c_s=5	6.84%	6.72%	6.84%	6.87%	6.90%	6.69%	6.84%	6.59%
c_s=10	7.88%	9.24%	10.20%	10.15%	10.15%	10.18%	10.22%	9.89%
c_s=15	6.95%	9.89%	9.24%	9.58%	9.56%	9.59%	9.55%	9.55%
c_s=20	5.91%	9.19%	9.45%	9.21%	9.22%	9.23%	9.33%	9.24%
c_s=25	4.86%	8.81%	9.33%	9.09%	9.03%	9.36%	9.09%	9.09%
c_s=30	5.03%	8.42%	8.81%	9.25%	8.95%	8.76%	8.93%	8.93%
c_s=35	4.57%	7.97%	8.76%	9.12%	9.10%	9.09%	9.10%	9.04%
c_s=40	3.84%	7.52%	8.35%	8.96%	8.96%	8.82%	8.64%	8.55%

To understand how the hybrid policy works, we also record the numbers of times that synchronizations are trigged by the time-based threshold and total control threshold, as summarized in Table D3. The results are as expected —when the normalized business disruption cost (c_d) is low and dominated by the normalized patching setup cost (c_s), the synchronizations are mostly triggered by the total control threshold; when the normalized business disruption cost (c_d) dominates the normalized patching setup cost (c_s), synchronizations are mostly triggered by the time-based threshold. When $c_d \geq 20$, among all the scenarios, only one synchronization is triggered by the total control-based threshold and all other synchronizations are initiated by the time-based threshold.

Table D3. Numbers of Synchronizations Triggered by Time-based and Total Control Thresholds

	c_d =5	c_d =10	c_d =15	c_d =20	c_d =25	c_d =30	c_d =35	c_d =40
c_s =5	(4336, 282)	(4742, 1)	(4743, 0)	(4742, 0)	(4743, 0)	(4743, 0)	(4743, 0)	(4743, 0)
c_s =10	(2184, 949)	(3340, 11)	(3353, 0)	(3353, 0)	(3353, 0)	(3353, 0)	(3353, 0)	(3353, 0)
c_s =15	(1233, 1293)	(2662, 59)	(2737, 1)	(2738, 0)	(2738, 0)	(2738, 0)	(2738, 0)	(2738, 0)
c_s =20	(709, 1483)	(2174, 167)	(2369, 2)	(2371, 0)	(2371, 0)	(2371, 0)	(2371, 0)	(2371, 0)
c_s =25	(454, 1531)	(1794, 285)	(2110, 9)	(2121, 0)	(2121, 0)	(2121, 0)	(2121, 0)	(2121, 0)
c_s =30	(319, 1502)	(1488, 398)	(1905, 28)	(1936, 0)	(1936, 0)	(1936, 0)	(1936, 0)	(1936, 0)
c_s =35	(232, 1459)	(1270, 472)	(1748, 37)	(1792, 0)	(1792, 0)	(1792, 0)	(1792, 0)	(1792, 0)
c_s =40	(166, 1429)	(1065, 553)	(1591, 72)	(1675, 1)	(1676, 0)	(1676, 0)	(1676, 0)	(1676, 0)

Why the TDS Policy Outperforms the Hybrid Policy?

To understand why the TDS policy consistently outperforms the hybrid policy, we first examine how the hybrid policy works. The hybrid policy tries to take advantage of the strengths of both the time-based policy and the total control policy. Which policy plays a more dominant role largely depends on the relative size of the business disruption cost:

(1) When the disruption cost is low in relation to the setup cost, synchronization operations under the hybrid policy will be primarily triggered by its total control policy component. This result comes with an extra cost — most synchronization operations under the hybrid policy will incur business disruption cost, while the disruption cost is never incurred under the TDS policy. As validated by our experimental results, this difference itself can erode any small theoretical advantage that the total control policy may enjoy over the TDS policy.

(2) When the disruption cost is high compared to the setup cost, synchronizations under the hybrid policy will be primarily triggered by the time-based policy. As we have shown in the paper, when the time-based policy (with an optimized synchronization interval) is a standalone policy, it is dominated by the TDS policy. This result is theoretically intuitive because synchronization at every decision epoch (by setting the synchronization threshold extremely low) is a possible scenario under the TDS policy. When the time-based policy is a component of the hybrid policy, as shown in Theorem 3 of Dey et al. (2015), its optimal time interval threshold (x_H^*) increases from that of the standalone policy (x^*). When comparing the time-based policy against the TDS policy, increasing the time interval of the former generally makes it worse when compared with the later. Although the lower synchronization frequency issue can be compensated by the inclusion of the total control policy, any synchronization triggered under the total control policy comes at an extra cost — the business disruption cost. With all factors considered, the TDS policy should outperform the hybrid policy under this scenario.

In sum, the dynamic nature of the TDS policy (running synchronization only when the benefit is greater than the cost) and the fact that it can avoid business disruption cost bring too much of an advantage for the hybrid policy to overcome, hence the TDS policy can outperform the hybrid policy.

Reference

Dey, D., Lahiri, A., and Zhang, G. 2015. "Optimal Policies for Security Patch Management," *INFORMS Journal on Computing* (27:3), pp. 462-477.

Silvers, F. 2011. *Data Warehouse Designs: Achieving ROI with Market Basket Analysis and Time Variance,* Boca Raton, FL: CRC Press.