Ordering Raw Data and System Time for Read Only devices

Author: Petre IORDANESCU, Date: December 2019
Categories: Distributed Systems, Software Design & Architectures

MOTIVATIE

Acest material este rezultatul unei cercetari și analize în vederea realizării sistemului DISCARD-COIN[1] (*Distributed Scalable Raw Data Collector from Interface , RENware P/N 0000-0101*). Acest sistem urmărește colectarea, secventierea și stocarea persistenta a datelor de la diverse dispozitive din teritoriu. Ca sa înțelegeți mai bine aceasta componenta de sincronizare și scopul ei, un simplu exemplu este suficient: sa presupunem o locatie publica cu mai multe camere de luat vederi. Fiecare camera transmite propriile informații iar un sistem uzual de secventiere și ordonare (multicast clasic pe care il poate asigura chiar routerul). Acest lucru nu inseamna decit ca „filmul" fiecărei camere individuale are „un streaming" corect și este consistent la urmărirea acestuia.

Totul bine și frumos, dar daca dorim sa urmărim „filmul complet" al unui individ care s-a ridicat de la o masa și s-a dus în alta parte, înregistrarea unei singure camere video nu mai este suficienta. Individul s-a ridicat și mergând a „iesit" din raza de acțiune a unei camere video, a intrat în raza de acțiune a alteia, etc (și inca este bine cit timp camerele video sunt așezate de asa natura astfel incit sa nu existe unghiuri moarte).

Trebuie sa CORELAM înregistrările mai multor camere video, însă secventialitatea și ordonarea INTRE CAMERE u a asigurat-o nimeni... Ne mai ramine timpul în speranta ca toate camerele au ceasurile perfect sincronizate intre ele ceea ce este o cerinta 99% aproape imposibila[2]...

Acum imaginați-va ca informația vine și de altfel de dispozitive, combinat (GPS telefon de exemplu)... Fiecare dispozitiv individul luat este OK – daca vrem sa corelam informațiile de a mai multe dispozitive însă, ne va trebui răbdare multa și o minte odihnita. De aceea am facut acest studiu și a rezultat acest material.

IPOTEZE / PRECHIZITE

Materialul prezent presupune că următoarele ipoteze și principii sunt cunoscute și le va folosi fără a oferi explicații suplimentare referitoare la conținutul acestora în sensul de algoritmi, definiții, vocabular sau alte elemente. Aceste lucruri, materiale, publicații, etc sunt citate și referențiate și folosite presupunând că sunt FOARTE CONSCUTE cititorului.

IPOTEZE

- Se cunosc principiile expuse de Lamport[3] referitoare la ceasuri logice și ordonare totală[4]
- Se va lucra cu modelul de ceas logic Lamport organizat ca implementare sub formă de vector (referințe multiple pe Internet)[5]
- Toate ceasurile logice aferente nodurilor rezidă într-un vector în care indexul este nodul; de exemplu vectorul A = 5, 2, 9] denotă un sistem format din 3 noduri: nodul 0 care are ceasul logic la valoarea 5, nodul 1 care are ceasul logic la valoarea 2 și nodul 2 care are ceasul logic la valoarea 9
- Toate nodurile conțin / întrețin un vector identic, acesta fiind ceea ce Lamport referențiază drept timestamp-ul derivat din ceasul logic al unui sistem (a nu se face confuzie cu un timestamp fizic, real). În ceasul logic vectorial aferent fiecărui nod, pozițiile indexului sunt neschimbate și consistente (referă aceleași noduri) astfel încât să existe comparabilitate între vectorii oricăror două noduri diferite
- Restul lucrurilor sunt exact așa cum au fost descrise de Lamport în materialul de baza referentiat în acest articol

OBIECTIVE

Prezentul material atinge doua aspecte importante, și anume tratarea mesajelor întârziate și rezolvarea acestora adica exact un „multicast" cu observația ca se aplica unor „emițători" eterogeni, ‚not reliable' și care în esenta **NUMAI TRANSMIT DATE**, capabilitățile acestora de a recepționa date fiind foarte limitate și de cele mai multe ori inexistente.

OBIECTIVUL 1 – MESAJE ÎNTÂRZIATE

Găsirea / identificarea unei soluții acceptabile de soluționare a mesajelor întârziate (delayed). În figura 1 spre exemplu, mesajul ***a1*** ajunge după sosirea mesajului ***a2*** cu toate că a fost transmis înaintea acestuia.

În figură, pe lângă „anomalia" observată, trebuie remarcat (chiar dacă pare evident, sublinierea este totuși bine venită) ca NODUL central, la momentul sosirii mesajului a2 nu are de unde să știe:

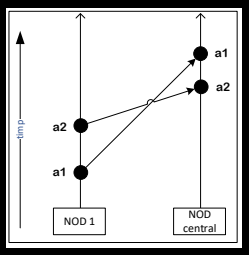

Figura 1

- Că mesajul ***a2*** este al doilea eveniment și nu primul întâmplat
- Că mașina NOD 1 a mai trimis un mesaj anterior acestuia
- Că va mai primi un mesaj care trebuia primit înaintea acestuia

Pentru rezolvarea acestui obiectiv trebuie minim ca mașina NOD central să poată „detecta anomalia" la momentul sosirii lui ***a2*** și aceasta fără a avea nici un fel de cunoștințe speciale despre NOD 1 – trebuie să poată face acest lucru indiferent cine sau de ce tip este (cameră, GPS, etc) este NOD 1.

OBIECTIVUL 2 – TIMP FIZIC REAL ÎN SISTEM

Ne interesează ca sistemul ca întreg să posede un ceas real COMUN ȘI UNIC bazat pe timp fizic și care să permită DEDUCȚII LOGICE umane și cu sens (pertinente ca și comparabilități între timpi) prin analiza combinată a informațiilor de la mai multe surse de date. Prin analiză combinată se înțelege analiză temporala între surse de date diferite.

Alte ipoteze

La acest moment la care au fost stabilite obiectivele, noi ipoteze trebuiesc statuate. Astfel:

Toate sistemele sunt egale (au același tratament iar din punct de vedere al principiilor de funcționare pot fi înlocuite unul cu celălalt fără a altera fluxul normal de operații și logica de funcționare a sistemului).

Excepție de la această regulă fac două sisteme, NODUL central și NODUL WD (ce va fi introdus odată cu prezentarea modalității de soluționare a obiectivului 2 și va fi explicat la acel moment).

NODUL central diferă print faptul este doar unul (din punct de vedre logic, altfel fizic acesta poate fi distribuit ACTIV – ACTIV pe mai multe mașini fizice) și doar recepționează mesaje (cu excepția faptului că poate și transmite comenzi catre nodurile dispozitive de culegere date din mediul fizic-real, dar acest lucru nu are nici o legătură – nici nu ajută nici nu încurcă – cu procesele și fenomenele descrise în acest document)

NODUL WD ale cărui mesaje au un conținut special în rest fiind complet identic și egal cu orice alt NOD emitent de date brute din mediul fizic.

Mai trebuie remarcat faptul că DOAR SISTEMUL CENTRAL este interesat de menținerea unei ordini, de reordonare și de timp fizic. Restul nodurilor nu au nici un fel de preocupare în acest sens și nici nu sunt sau trebuie să fie conștiente (EN: aware) de acest lucru. Acest lucru este important deoarece permite utilizarea de sisteme / dispozitive colectoare de date din mediul fizic așa cum sunt ele și fără a impune cerințe speciale de funcționare la nivelul acestora (adică se pot utiliza camere de luat vederi obișnuite, dispozitive GPS obișnuite – doar să poată fi conectate la rețeaua de comunicații digitale – LAN, Internet, etc).

OBIECTIVUL 1 – REZOLVAREA MESAJELOR ÎNTÂRZIATE

CONSIDERENTE INIȚIALE

Un element important este în abordare este să fim interesați NUMAI de relația „izolată" a fiecărui nod cu nodul central. Această abordare permite

[1] http://www.renware.eu/product_desc/MIDDLE%20TIER%20DISTRIBUTED%20COMPUTING%20FRAMEWORKS#ed424944-855d-4558-ab28-18db237c7828

[2] Se spune in popor ca cine are un ceas știe ce ora este, însă cine are mai multe ceasuri, nu va sti ce ora este...

[3] Leslie Lamport: https://en.wikipedia.org/wiki/Leslie_Lamport

[4] Lamport, L. (1978). "Time, clocks, and the ordering of events în a distributed system" *(Communications of the ACM . 21 (7): 558–565)*

[5] https://en.wikipedia.org/wiki/Vector_clock

studierea și abordarea procesului la nivel micro iar rezolvarea „dispozitiv cu dispozitiv" a aspectelor de acest gen va duce inerent la o funcționare consistentă a întregului sistem.

Analiza *combinată – temporală* inter-dispozitive va fi rezolvată odată cu rezolvarea obiectivului 2; la acest moment interesează doar identificarea unei soluții de „recuperare" a mesajelor pierdute de la un dispozitiv situație similară cu cea prezentată în figura 1.

Notă: în identificarea unei soluții de rezolvare a acestui obiectiv se vor folosi tehnicile de ordonare a ceasurilor logice așa cum au fost ele prezentate de Lamport și extinse la varianta vectorială.

IDENTIFICAREA ANOMALIILOR CARE INTRĂ SUB INCIDENȚA OBIECTIVULUI 1

Așa cum s-a arătat la descrierea acestui obiectiv, în primul rând NODUL central **trebuie să poată identifica anomaliile de genul celei arătate în figura 1 fără a fi necesară comunicarea cu alte noduri și fără menținerea de stări suplimentare AFERENTE FIECĂRUI NOD din sistem**, ceea ce ar mări ne acceptabil spațiul stărilor, complexitatea software-ului nodului central și ar diminua considerabil capabilitățile acestuia de a lucra consistent și eficace ca și nod distribuit, asincron, activ-activ, diminuarea neputând fi compensată decât de o creștere exponențială și nejustificată a software-ului aferent (neluând în calcul impactul asupra întreținerii acestui software).

Considerăm că la acest moment este necesară prezentarea unui exemplu mai amplu (cu mai multe mesaje) între un ***NOD dispozitiv*** și ***NODUL central***.

Astfel SC este singurul sistem care întreține o coadă a tuturor mesajelor de la toate camerele – în exemplu doar una – coada este *SHARED* (partajată) și *UNICĂ* pentru toate SC-urile din sistem în cazul acestea sunt distribuite balansat / activ-activ din rațiuni de performanță / redundanță (evident, nimic nu trebuie să oprească SC să poată fi distribuit balansat / activ-activ). În figura 2 se văd ceasurile logice ale C1 și SC, iar la SC ajunge și LCLK C1 conform Lamport.

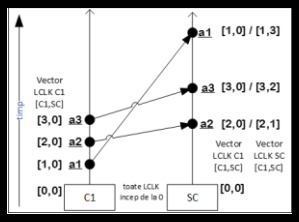

Figura 2

Acum putem vedea cum arată coada **UNICĂ – UNIFICATĂ** (MAINEQ – main events queue) după producerea / întâmplarea evenimentelor arătate în figura 2 :

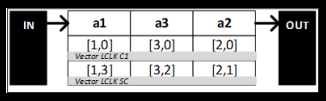

Figura 3

- Din figura 3 reiese ca ***a2*** este primul eveniment sosit (coada este FIFO) iar ***a1*** este ultimul eveniment sosit.
- În coadă sunt păstrați și vectorii pentru a fi folosiți la ordonarea mesajelor.
- La momentul intrării în coadă a mesajului ***a2***, SC nu avea „de unde să-și cunoască viitorul", deci nu avea de unde să știa ca va mai sosi și ***a1*** întârziat.
- Un lucru important care trebuie remarcat este faptul că între nodurile SC și C1 (în general C*i*, pentru ∀*i*) există o **completă** și **totală separare** din punct de vedere al stărilor – cele două noduri (oricare două noduri) pot cel mult să comunice prin transmitere de mesaje sau să păstreze (să memoreze informații aferente ultimului mesaj)

Dacă ne plasăm la nivelul SC și analizăm ceasurile logice Lamport (în figura 3) se observă că evenimentul / mesajul ***a1*** prezintă o situație altfel decât a celorlalte evenimente – ceasul logic aferent nu este într-o relație strict crescătoare față de ultimul ceas Lamport ([1,3] < [3,2]). Acest lucru DENOTĂ O ANOMALIE (DE TIP „EVENIMENT ÎNTÂRZIAT) AFERENTĂ LUI ***a1***.

METODA ȘI STRATEGIA DE REZOLVARE PROPUSĂ

Ce se poate face cu ***a1*** în această situație? Vom aplica două strategii în aceste situații:

- S1 – prin prima strategie se încearcă întâi re-plasarea lui ***a1*** în locul natural în care trebuia să fie în ordinea normală în care au fost transmise evenimentele (vom arăta mai încolo care este locul natural, cu toate că este evident acest lucru);
  în contextul extins și al unui ceas fizic real al sistemului – prezentat la obiectivul 2 – se poate obține „mai multă informație" după re-plasarea lui ***a1***)
- S2 – dacă locul natural în care trebuia să fie ***a1*** nu mai există în coada MAINEQ[6], atunci se va folosi o altă coadă dedicate acestor tip de mesaje, coada LADDQ (late and delayed queue) pentru colectarea mesajelor întârziate, coadă ce va fi prelucrată / preluată de către procesul (nivelul) care se ocupă de persistarea evenimentelor[7] (salvarea acestora într-o zonă cu persistență mare) și care va plasa evenimentul în locul său natural.

ALTE CONSIDERENTE

Referitor la coada LADDQ, aceasta poate stoca mesajele / evenimentele întârziate cu termen de valabilitate. Dacă se specifică un termen rezonabil de de exemplu "mai mult 10 zile" dar nu extrem de larg „de exemplu < 30 zile" această facilitate poate fi „de bun simț" și poate fi folosită pentru cazuri de excepție, de exemplu layerul de colectare date pentru persistența acestora are un bug și ștergerea mesajelor citi și prelucrate / comise a „fost uitată".

OBIECTIVUL 2 TIMP FIZIC REAL ÎN SISTEM

PRINCIPII GENERALE

În realizarea acestui obiectiv s-au luat în considerare următoarele principii:

- P1: Timpul fizic / real va fi unul al sistemului luat ca și „tot logic"; acesta va fi timpul real considerat de sistem în asociere cu informațiile / mesajele / evenimentele primite de la toate dispozitivele de culegere date brute; orice ce alți timpi / ceasuri nu vor fi considerate în conjunctură cu dispozitivele menționate anterior;
- P2: Timpul fizic / real al sistemului (ce va fi numit în continuare SYT) este independent de timpul surselor de date, de timpul oricărui dispozitiv de culegere date și de timpul oricărui sistem SC;
- P3: SYT nu are și nu trebuie să aibă vreo legătură cu timpul sursei sau cu timpul destinației; SYT trebuie să fie un timp generat independent, asincron, complet imparțial și agnostic referitor la alți timpi /ceasuri care pot fi identificați / identificate în sisteme; SYT nu are și nu trebuie să aibă nici o legătură cu nivelul de distribuire și de scalare a componentelor sau cu modul de funcționare al acestora;
- P4: Generatorul de SYT va funcționa complet imparțial și va genera cuante de timp real fără a aștepta sau a avea nevoie de vreo confirmare sau comandă nici înainte și nici după generarea oricărei cuantă de timp; generatorul SYT va funcționa indiferent de starea oricărei alte componente din sistem;
- P5: SYT trebuie să fie un timp „cât se poate de real", survenit în urma unor sincronizări periodice cu un server de timp / Time Server folosind protocol standard NTP;

MODUL DE FUNCȚIONARE

La acest moment va fi introdus un nou tip de nod în sistem, nod care va fi numit WD și care este „egal[8]" cu toate celelalte noduri cu următoarele excepții:

- Nodul WD doar emite evenimente către SC (identic cu oricare nod de culegere date) și nu recepționează niciodată nimic (pe principiul este „surd" care va asigura respectarea principiului P4);
- Nodul WD respectă toate principiile Lamport în sensul în care evenimentele emise vor fi însoțite de propriul ceas logic (LCLK) care este exact un ceas logic Lamport și nu are nici o legătură cu timpul real;
- Nodul WD își sincronizează periodic și constant propriul ceas fizic cu un server de timp standard; acest lucru va asigura micșorarea erorilor de timp (la 0 nu are cum să le reducă nimeni);
- Nodul WD nu va fi niciodată scalat la nivel activ-activ (altfel nu mai avem un ceas ci mai multe!) ci numai la nivel activ-pasiv în scop de redundanță – este important ca nodul SD să fie unic și singur altfel erorile de drift / skew între mai multe ceasuri nu vor putea fi corectate sau știute cu precizie maximă (prin „balansarea mașinii" se elimină din start orice șansă de a le cunoaște sau a le corecta);
- Nodul WD va acționa ca un sistem care oferă timp real cu observația importantă că prin comportamentul acestuia „el nu este un ceas la care te uiți dacă dorești să faci asta" ci este un sistem care te anunță cit este ceasul periodic și constant indiferent de dorința ta și te obligă să îți notezi acest timp chiar dacă nu vei face nimic cu el.

[6] Pentru ca cel mai probabil a fost deja consumat / folosit de un sistem extern; a nu se pierde din vedere ca ORDSYT este o componenta utila in conjunctura folosirii ei in alte sisteme care consuma informația, altfel de sine stator nu are decit valoare pur didactica

[7] Neprezentat in acest articol

[8] Adica nu este ceva mai special sau favorizat de catre alte sistem

Astfel se asigură principiile inițiale de la care s-a plecat inclusiv imparțialitatea și independența acestui nod.

Revenind, această mașină este complet egală cu celelalte. Vom nota evenimentele acestei masini cu ***wd*** (pentru a le diferenția vizual de evenimentele a, b, ... de la dispozitivele de colectare date brute).

Este de remarcat faptul ca acestui nod i „se poate întâmpla" orice astigmatism sau hazard care poate apare în comunicația dintre Cx și SC, inclusiv întârzierea de evenimente.

Acest nod (WD) trimite mesaje **wd** periodic, constant și cu o frecvență pre-stabilită către SC.

Să facem o noua diagramă temporală în care pe lângă un nod Cx vom adăuga și nodul WD; vectorul de ceasuri logice Lamport va avea componența [C1, SD, WD]:

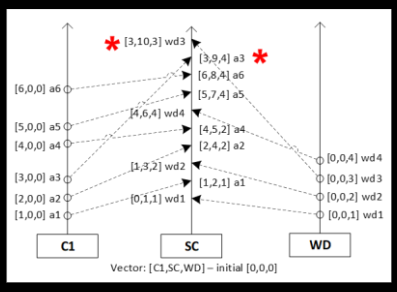

Figura 4

Anomaliile (evenimente sosite întârziat) au fost deja marcate pe figură pentru o mai bună vizibilitate, fără a avea vreo legătură cu modul de funcționare al sistemului.

Să vedem cum arată o coadă de mesaje în urma evenimentelor prezentate / intimplate în figura 4. Dar înainte se face o remarcă asupra evenimentelor întârziate de tip ***wd***. Acestora nu li se aplică nici una din strategiile prezentate pentru mesajele „normale" sosite de la noduri Cx. Pur și simplu mesajele ***wd*** întârziate „se *aruncă*" și se renunță la ele (discard). Deci să vedem cum arată coada după aceste evenimente.

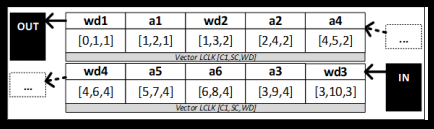

Figura 5

Acest lucru este observa în figura 5 unde este arătata coada după întâmplarea evenimentelor din figura 4 – coada este secționată doar pentru a încăpea mai bine pe ecran și nu are nici o legătură cu modul de funcționare al sistemului.

Se observă imediat „anomaliile" ***a3*** și ***wd3*** care nu respectă regula de „crescător a vectorului Lamport". Așa cum s-a spus anterior, regula pentru ***wd3*** întârziate este simplă – acesta „se aruncă". Aplicarea regulii doar pentru ***wd3*** (presupunând că ***a3*** nu este întârziat) se poate vedea în figura 6.

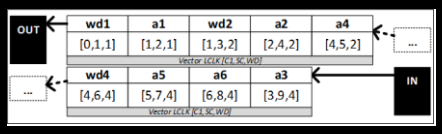

Figura 6

În ceea ce privește ***a3***, locul acestuia este între ***a2*** și ***a4*** (se pot analiza vectorii și se găsește CEL MAI BUN loc la o parcurgere „înapoi– spre OUT a vectorului Lamport în care vectorul aferent respectă regula de „mai mare". Dacă „acest loc a fost consumat (i.e. se ajunge la TAIL OF FIFO QUEUE) fără a identifica acest loc – adică am mai putea căuta – atunci pentru ***a3*** se aplică strategia S2 prezentată; este de remarcat faptul că ***a3*** se va muta în LADDQ însă pentru orice eventualitate se vor copia și ***wd***-urile mărginitoare cele mai apropiate (atenție toate aceste lucruri se fac la o singură parcurgere a cozii – ceea ce se vede în figura 6 este doar o prezentare simplificatoare a unei situații – în cazul mutării lui ***a3*** va trebui copiat și ***w3*** pentru a avea o minimă referință de timp, apoi va fi șters).

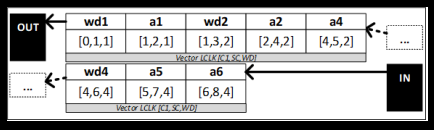

Figura 7

Presupunând că acest loc „a fost deja consumat", deci ***a3*** va ajunge în coada LADDQ împreună cu ***wd3*** iar coada MAINEQ va arăta ca in figura 7 în final.

Acum trebuie spus și elementul esențial al întregii prezentări vis-a-vis de obiectivul 2. Toate pachetele / mesajele ***wd*** au un conținut de interes pentru SC spre diferență de celelalte mesaje de la orice Cx care **nu prezintă interes ca și conținut pentru SC**. Pachetele / mesajele ***wd*** au ca și conținut timpul mașinii WD la momentul emiterii fiecărui wd.

Dacă la acest moment refacem coada MAINEQ cu pachete wd desfăcute vom obține informația din figura 8. În figură timpul a fost marcat ca și hh1, hh2, etc, în realitate fiind un timestamp cât se poate de real.

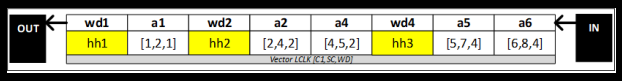

Figura 8

Jaloanele de timp au fost marcate cu fundal galben pentru o mai bună vizibilitate. Analizând informația obținută vom obține:

- Evenimentul a1 a avut loc în intervalul de timp (hh1,hh2) care este obligatoriu deschis altfel pe de o parte nu respectă în totalitate regulile Lamport iar pe de altă parte sincronizarea perfectă în timp până la ceva care să fie identic total (ceea ce este o imposibilitate în realitate ci este posibilă doar matematic pe hârtie)
- Evenimentele ***a2*** și ***a4*** au avut loc în intervalul (hh2,hh3) exact în ordinea ***a2*** și apoi ***a4***
- Evenimentele ***a5*** și ***a6*** au avut loc în intervalul (hh3,prezent) exact în ordinea ***a5*** și apoi ***a6***

Astfel datele colectate de dispozitive (conținutul mesajelor ax) sunt plasate și pe o scală temporală reală și suportă analiză combinată și temporală între ele. Așa cum sunt prezentate ax exact în același fel apar pe această scală și eventualele bx, cx, etc, analiza temporală între ele neavând nici o legătura cu sursa de la care provin, conținutul lor sau ruta pe care au parcurs-o de la sursă până la destinație.

CONCLUZII FINALE

GRANULARITATEA – Frecvența de transmisie / generare a lui WD către SC dă granularitate minimă a intervalelor de timp în care un eveniment de la un nod Cx poate fi încadrat.

UNIVERSALITATEA TIMPULUI – timpul trebuie să fie *UTC* în toate sistemele de stocare fie ele interne, externe, permanente, volatile etc. toate operațiile matematice de adunare, scădere, etc asupra timpilor trebuiesc făcute în *UTC* și apoi convertite în timpi locali (dacă sunt disponibile doar în timpi locali trebuiesc întâi convertite în *UTC* înainte de a efectua orice operații).

Conversiile *UTC* – timpi locali și invers trebuiesc făcute după metode cât se poate de standardizate sau identice de-a lungul întregului sistem; se face remarca că diverse sisteme ***pot*** prezenta flavour-uri / nuanțe diferite pentru anumiți timpi potențial mai exotici (în special pentru zonele unde se aplică schimbări de fus ora și la jumătăți de ora !).

Se atrage atenția că intre delta-urile acelorași timpi calculate în UTC și în timpi locali pot apare diferențe de minim +/- 1 oră dacă extremele temporale luate in calcul traversează fusuri orare diferite.

VIZIBILITATEA VARIAȚIILOR TIMPULUI – Afișarea în ‚user interface' a scalelor în timpi locali este recomandat să aibă jaloane vizibile în locurile în care au avut loc sau se știe că vor avea loc schimbari de fus orar.

În final se face observația că mașina WD trebuie să fie de o calitate impecabilă în ceea ce privește modulele RTC proprii și interfețele de rețea, aceasta fiind necesar pentru a minimiza erorile de timp inerente (drift-uri, skew-uri, etc). Sistemul va funcționa fără probleme și cu mașini WD de slaba calitate însă fenomenul GIGO (garbage in generates garbage out) își va pune amprenta pe calitatea datelor din sistem.

REFERINȚE

- ***Leslie Lamport*** (Lamport worked as a computer scientist at Massachusetts Computer Associates from 1970 to 1977, SRI International from 1977 to 1985, and Digital Equipment Corporation and Compaq from 1985 to 2001. În 2001 he joined Microsoft Research în Mountain View, California, which closed în 2014)
  *Time, Clocks, and the Ordering of Events ina Distributed Syste*m
  Communications of the ACM, July 1978, Volume 21, Number 7
- ***Andrew S. Tanenbaum*** (Professor of Computer Science at the Vrije Universiteit in Amsterdam, Netherlands, as the head of the Computer Systems Department)
  Distributed Operating Systems
  Pearson Education, 1994, ISBN-10: 0132199084, ISBN-13: 978-0132199087

RENware team și Petre IORDANESCU, 2020

(mai multe articole gasiti pe: http://www.renware.eu/articles)