harbour - Podrška #24716

san o semaforima: fmk/harbour semafor dbf/postgresql

06.10.2011 15:07 - Ernad Husremović

Status:	Zatvoreno	**Početak:**	06.10.2011
Prioritet:	Normalan	**Završetak:**	
Odgovorna osoba:	Ernad Husremović	**% završeno:**	0%
Kategorija:		**Procjena vremena:**	0.00 sat
Ciljna verzija:			

Opis

Povezani tiketi:

korelira sa harbour - Podrška #24728: fmk semafori, prototip	**Zatvoreno**	**07.10.2011**

Historija

#1 - 06.10.2011 15:14 - Ernad Husremović

sinoć nisam dobro spavao. tamo u pola jedan počnu mi misli ... i ne staje ...
do pola četiri nisam zasp'o

A san je vezan za fmk. hrkljuš !?X?=?F ... ovaj moj mozak je fakat prokliz'o :)

Iako polukošmar, nešto mi se interesantno učinilo za zapisati.

Koncept fmk sa dbf/PostgreSQL

lokacija dbf-ova - privatni keš ~/.fmk/bringout/

sve tabele jednog programskog modula staviti u ~/.fmk/<oznaka_firme>, npr ~/.fmk/bringout

sve te tabele trebaju biti keš podataka sa servera (šifrarnici i tabele kumulativa).

#2 - 06.10.2011 15:30 - Ernad Husremović

semafor sistem

na postgresql serveru napravimo semafor tabelu sa sljedećim poljima:

1. table
2. algorithm
3. version
4. date_last_transaction
5. user_last_transaction

primjer takve tabele PostgreSQL database: fmk_bringout: table: public.fmk_semaphor

table	algorithm	version	date_last_transaction	user_last_transaction
suban	date	89	15.09.2011	vzeljka
konta	all	77	17.09.2011	vsasa
partn	all	872	17.09.2011	semir

~/.fmk/bring.out/my_fmk_semaphor.dbf

moja tabela može da se razlikuje:

table	algorithm	version	date_last_transaction	user_last_transaction
suban	date	85	15.09.2011	vzeljka
konta	all	77	17.09.2011	vsasa
partn	all	870	17.09.2011	vsasa

Kada god trebamo šifrarnik ili kumulativ aplikacija provjerava da li je naš primjerak tabela ažuran.

Ako nije - radimo refresh lokalnih dbf tabela postgresql=>dbf

Polje Algoritam je bitno da se za svaku budaleštinu ne desi "vozanje" svih zapisa.

Za naše najveće tabele, možemo utvrditi datumska polja koja se smatraju identifikatorima opsega podataka koje treba refreshirati.

Tako recimo u primjeru suban tabela koristi kao ključno polje datum dokumenta.

U gornjem primjeru na serveru imamo verziju 89 na serveru a naša lokalna kopija je 85. Algoritam je date (i koristimo datum_dok kao ključno polje za određivanje opsega za sinhronizaciju).

Sinhro opcija će uraditi fetch postgresql where date>= minimum od datuma koji ima na serveru i lokalnom kešu u ovom slučaju to je 15.09.2011

Znači svaki novi dokument u toku dana će mijenjati verziju. Sinhro opcija će "vozati" između servera i lokalnog keša SAMO podatke od tog dana.

#3 - 06.10.2011 15:31 - Ernad Husremović

algoritam "all"

šifrarnici su manje dinamični u smislu promjene. kod njih će algoritam "all" - mjenjaj sve zapise ako verzija na serveru > verzija lokalnog keša.

tu nema puno pameti.

Naravno to se može usavršavati novim algoritmima, ali realno nije bitno.

#4 - 06.10.2011 15:32 - Ernad Husremović

tako u gornjem primjeru trebamo sinhronizirati samo "partn" tabelu.

#5 - 06.10.2011 15:34 - Ernad Husremović

možeš sve pobrisati iz keša ... ako bilo šta zapne !

šta ako se pobriše ~/.fmk/bringout - nestaju podaci iz pripreme i parametri (ovo bi svakako bilo dobro prebaciti na server i primjeniti istu taktiku kao kumulativne datoteke ... i problem riješen)

pored toga reindeks nije problematičan. dešava se lokalno za jednog korisnika.

#6 - 06.10.2011 15:36 - Ernad Husremović

par giga na računaru svakog korisnika u lokalnom kešu !

Kod vindije bi ovo bio slučaj. Ali im imamo problem da sada iskorištavamo 1% kapaciteta lokalnog računara i da se dešavaju zapušenja u vozanju fajlova između servera i klijenta ...

Ovo definitivno nije super stvar, međutim ja mislim da bi lokalna harbour aplikacija, posebno za potrebe **unosa podataka** !!!! (to je nama "rak-rana" sa clipper-om to) žvakala ko od šale.

#7 - 06.10.2011 15:42 - Ernad Husremović

big picture

postojeća aplikacija kod svakog korisnika bi radila sa lokalnim kešom. Znači imali bi jednokorisnički rad

Na postgresql serveru bi imali podatke. Otvara nam se odmah Analysis/Reporting sistem korisničkih podataka sa standardnom SQL bazom. Korisnici kao što je vindija bi imali mogućnost da odbc-om ili na bilo koji standardan način pokupe podatke koje žele.

Primjer api view-ova koji se vide u xtuple bazama su sjajan primjer mogućnosti izolacije konkretnih tabela od pristupa korisnika. Korisnik bi imao pristup api.fin_transactions koji bi bio u stvari view koji mergira suban i doks i anal .. i šta mu već može zatrebati u njegovim pregledima. Tu se mogu raditi i join-i sa tabelama partnera i konta ... tako da korisnik za reporting potrebe ima sve na tapetu ...

#8 - 06.10.2011 15:44 - Ernad Husremović

jednostavnost implementacije

Za razliku od mnogih "kunta-kinte" tehnika koje su mi ranije padale na pamet, imam dojam da se ova razlikuje **JEDNOSTAVNOŠĆU**.

Naime, sinhronizacija je poprilično dummy proces, a sama aplikacija, kada se podaci sinhroniziraju, ponaša se kao jednokorisnička dbf aplikacija !

znači imamo use case koji provjereno radi i to radi sasvim dobro u većini primjena.

#9 - 06.10.2011 15:48 - Ernad Husremović

moć PostgreSQL-a

postgresql 9.1 nudi sinhronu replikaciju out-of-box. Šta to znači. To znači da možemo imati dva servera koji će replicirati podatke i na taj način raditi horizontalno skaliranje baze.

Također, PostgreSQL je baza knowhow, tako da ovaj put nemamo puno rasipanja.

#10 - 06.10.2011 15:49 - Ernad Husremović

portiranje fmk bez vratolomija

Ovo bi mogla biti formula portiranja fmk koja nas ne uvodi u nekakve sulude vratolomije koje se na kraju dodatno stvari zakomplikovati, koje ne realno ne možemo dovesti do kraja niti isporučiti klijentima.

#11 - 06.10.2011 15:55 - Ernad Husremović

univerzalnost koncepta sinhronizacije

ovaj model sinhronizacije je generalno potreban.

javascritpt - HTML5/Localstorage sinhronizacija su mehanizmi online/offline rada. Koncept koji ovdje pominjem bi se analogno mogao primjeniti na ta scenarija u M3 ili POS-u ...

FMK POS i semafori

Ovdje dolazim do jedne možda i najinteresantnije priče.

delphy gaeway i postojeći koncept sinhronizacije, to svi znamo - nije dobar. Pun je iskakanja ... I kada mi radimo servisne aktivnosti ... kažemo: "pripremite log za period"

e taj log za period nije ništa drugo nego gore pomenut "date" algoritam, s tim što korisnik a ne sistem određuje datum od koga se treba sinhronizirati ...

Sa postojećim VPN-ovima i internet konekcijama prema prodavnicama, semafori u stvari mogu biti i rješenje sinhronizacije prodavnica !

#12 - 06.10.2011 16:06 - Ernad Husremović

povrati dokumenata i slične budaleštine

povrati su, to barem svi znamo, jedna od krajnje spornih opcija naših aplikacija.

Međutim, korisnici ih vole imati. Što kaže Nedra: mora biti povrat !

Ako u gornjem primjeru imamo suban:

```
10.09. ...

12.09. ...

....
18.09 ...
```

i mi vratimo dokument od 12.09 - algoritam će opet biti date, a datum 12.09. Znači mi jednostavno setujemo donji limit za koji znamo da je nedirnut i sinhroniziramo podatke.

Ono što se ovim mora nadograditi jeste to da se na postgresql serveru mora držati lista statusa za svakog korisnika.

primjer takve tabele PostgreSQL database: fmk_bringout: table: public.fmk_semaphor verzija 2:

ako imamo tri usera tabele suban onda moraju postojati tri zapisa:

table	user	last_user_synchro	algorithm	version	date_last_transaction	user_last_transaction
suban	hernad	80	date	82	12.09.2011	hernad
suban	vsasa	85	date	89	17.09.2011	vzeljka
suban	vzeljka	89	date	89	17.09.2011	vzeljka

uvođenjem zapisa za svakog usera (user, last_user_synchro na serveru, nisam siguran da nam uopšte my_fmk_semaphor.dbf treba.

#13 - 06.10.2011 16:12 - Ernad Husremović

- Fajl dbf2pg.prg dodano

- Fajl cache.prg dodano

kako ovo implementirati

u harbour/contrib/pgsql

dbf2pg ... znamo šta radi ...

interesantan je cache.prg on upravo radi vozanja o kojima pričam. E sada, nad ovim je potrebno napraviti inteligenciju u sinhronizaciji putem ovih semafora.

#14 - 06.10.2011 16:13 - Saša Vranić

uhhhhhhh, hernade hernade.... :)

#15 - 06.10.2011 16:24 - Ernad Husremović

kriterij Id prodajnog mjesta + dat_dok

sada su sve prodavnice posebne baze. Pa imamo 20 POS-ova. To je loše urađeno.

Na serverskoj strani treba postojati POS u koji bi sav prometi išao. Ako bi uradili konsolidaciju, onda bi se morao promijeniti kriterij sinhronizacije:

IdProdajnogMjesta + DatDok

Hm ali to kod nas onda spaja i šifrarnike što bi u slučajevima da trebamo različite cijene artikala izazvalo probleme, ili da prodavnice imaju različite asortimane ...

Hm to je kod nas loše riješeno pa zato. Cijena bi morala uvije da se veže uz prodajno mjesto. kod xtuple-a to je site. Svaki site (warehouse, store) može imati svoj cjenovnik ...

Da to je jedna od anomalija FMK - šifrarnik robe ovakav kakav jest ...

#16 - 06.10.2011 16:27 - Ernad Husremović

modifikacija struktura - server only

da li bi bilo potrebno raditi modifikaciju struktura na svakom kljentu ? ne bi.

slično kao "full" algoritam možemo dodati "full_zap" koji će tabelu kreirati iznova. Ako je na serveru promjenjena struktura on će tu strukturu pokupiti.

hmmm ... znači aplikacija bi uvijek gledala sa stanovišta struktura tabela kakvo je stanje servera. To je dobra stvar. Skroz dobra stvar.

#17 - 06.10.2011 16:34 - Ernad Husremović

pseudo kod

```
fuction use_with_sync(table)

if !exists(table) or is_semafor(table, my_identity, full_with_zap)
  kreiraj_sa_servera_iznova
endif

try probaj_otvoriti(table)
   say "ok otvorio sam table"
recover
   kreiraj_sa_servera_iznova
end

aSemafor[table] = get_semafor(table, my_identity, full_with_zap)

if aSemafor[table] treba sihnronizirati datum

   update_from_server( cTable, cIdFirma, cFromDate)

endif

say "huh ... napokon je moj cache frišak ..."
```

```
end function
```

#18 - 06.10.2011 16:37 - Saša Vranić

Iskreno rečeno sa ovog stajališta, zapravo trenutnog stajališta, mislim da bi rewrite fmk sa harburom prije implementirati nego što ćemo se tako brzo ufurati u xtuple pakete itd... Ali xtuple je jako dobra osnova za "krađu" ideja.

Ali, i meni se sve pomutilo više u glavi. Unazad dva mjeseca toliko sam stvari pretamburao... da ne znam ni gdje sam pošao ni gdje sam stigao.

Ovo što pišeš o psql i keširanju je najvjerovatnije super ideja, samo ja ne bih plaho pametovao oko postojećeg stanja, treba ići na rewrite svega toga... upravo radi tih stvari, šifrarnika, proširenja opcija, mogućnosti itd... oni naši id-ovi sa karakterima umjesto inkrementalnih polja itd...

#19 - 06.10.2011 16:38 - Ernad Husremović

izbaciti create dbf iz priče ?

Odgovarajuće imenovanje tabela je naravno neophodno.

PostgreSQL ima sjajan feature - schema sistem.

fmk_bringout database:

privatne tabele stavimo u shemu koje počinje aliasem korisnika:

- hernad.fin_params
- hernad.fin_psuban

u fmk shemu stavljamo kumulative

- fmk.fin_suban
- fmk.fin_doks
- fmk.kalk_doks
- fmk.partn
- fmk.konto

#20 - 06.10.2011 16:41 - Ernad Husremović

i onda klijent jednostavno sve sa baze počupa ... svaka modifikacija koja se desi na strani servera se jednostavno signalizira sa algoritmom keširanja "full_with_zap" (briši lokalnu dbf tabelu potpuno pa je nanovo kreiraj na osnovu podataka sa servera). i klijent kupi tabele koje su spremljene na serveru.

#21 - 06.10.2011 16:59 - Saša Vranić

a što se tiče ovoga treba ovo na papir fino staviti... teško je ovako tvoje misli pratiti

#22 - 06.10.2011 17:02 - Saša Vranić

na onu tvoju tablu u kancelariji... :)

Hoćeš da sutra dođemo pa da vidimo konkretno šta i kako ?

#23 - 06.10.2011 17:13 - Ernad Husremović

Saša Vranić je napisao/la:

> Iskreno rečeno sa ovog stajališta, zapravo trenutnog stajališta, mislim da bi rewrite fmk sa harburom prije implementirati nego što ćemo se tako brzo ufurati u xtuple pakete itd... Ali xtuple je jako dobra osnova za ukrast ideja.
>
> Ali, i meni se sve pomutilo više u glavi. Unazad dva mjeseca toliko sam stvari pretamburao... da ne znam ni gdje sam pošao ni gdje sam stigao.
>
> Ovo što pišeš o psql i keširanju je najvjerovatnije super ideja, samo ja ne bih plaho pametovao oko postojećeg stanja, treba ići na rewrite svega toga... upravo radi tih stvari, šifrarnika, proširenja opcija, mogućnosti itd... oni naši id-ovi sa karakterima umjesto inkrementalnih polja itd...

Ono što radimo na M3 i A4 je najperspektivnije. Ono što nudi xTuple je prije svega jedna čestita i već razrađena osnova.

xTuple je takođe "old school" aplikacija po mnogim stvarima.

Međutim, ključne stvari rade, i rade skroz dobro:

1. klasični desktop gui koji je za korisnika maksimalno komforan po pitanju unosa i pregleda podataka
2. reporting sistem za naše potrebe odmah upotrebljiv

Sistem ekstenzija je odlična ideja i pomjera xTuple od onoga što bi bila u potpunosti "old shcool" aplikacija.

Dinamički pristup kroz javascript ekstenzije koje se instaliraju na bazi daje takvom sistemu veliki stepen fleksibilnosti i kada se developer istrenira može isporučivati feature iznimno brzo ...
Tu je super stvar što je to sa stanovišta aplikaciongo developera sve opet eventdriven programiranje i javascript. Znači još jedna javascript platforma:

- appcelerator
- node.js
- xtuple_core

E sada naravno da bi se sve to koristilo trebamo se ufurati u taj core, ali ne moramo raditi totalni switch što bi bilo da imamo recimo potrebu da interveniše unutar core-a na C++ nivou.

od svega je ipak najbitnija baza !

Međutim, za mene je najvažniji xtuple-ov dio baza koja je vrlo "čista", čitljiva, sa dobro osmišljenom organizacijom sa shemama (api, public, svaki modul ima svoju schemu)

OpenERP brand new POS

Juče sam pisao članak o openerp-u i dođem do informacije da je openerp izdao POS modul (ali ne kao OSS) http://www.openerp.com/products/pos

To su aplikacije koje se moraju početi praviti. Mi se sa M3/A4 definitivno "vrtimo" oko ovoga jer nas javascript neminovon ubacuje u web track.

Kada pišeš appcelerator aplikaciju, switch na čistu web aplikaciju nije nikakav šok.

Međutim kada pišeš recimo čistu QT desktop C++ aplikaciju ... onda se dešava skok. Dva svijeta.

Ovaj koncept koji su kod xtuple-a napravili sa javascript package-ima je jedan značajan kompromis. Developer se udaljava od "web track" koncepta programiranja ali ne tako ekstremno kao u slučaju QT desktop C++ aplikacije, a da ne pominjemo pisanje FMK/harbour aplikacije :)

Ako bi rangirali switchanje developera od 1..10, pri čemu se klasična web javascript HTML5 aplikacija uzima kao polaznu osnovu (web track). Onda je swithc sljedeći:

- nova HTML5 web aplikacija = 1 (switch se ne dešava)
- appcelerator rich client mobilna aplikacija = 4 (neminovno, s obzirom da se mora intenzivno koristiti API mobilnih uređaja)
- appcelerator rich client desktop aplikacija = 2 (veoma mali switch - to i jeste browser aplikacija koja može imati više native funkcija - manje web browser security ograničenja)
- qtscript xtuple_core package javascript = 6-7
- QT C/C++ = 8-9
- harbour app = 10
- Clipper DOS app ... izlazi iz skale :)

Pišem ovo zato što mislim da ovi skokovi stvaraju troškove, odnosno smanjuju naš kapacitet i efikasnost ako sebi za konačni cilj postavljamo pravljenje "web track" ERP aplikacija. Cool aplikacija, ono što je trend. Ono što se vrti na mobilnim uređajima tabletima IPAD-ovima.

#24 - 06.10.2011 17:34 - Ernad Husremović

Saša Vranić je napisao/la:

> na onu tvoju tablu u kancelariji... :)
>
> Hoćeš da sutra dođemo pa da vidimo konkretno šta i kako ?

Ne boj se. Nema nove promjene plana. Mi trebamo pripremiti P1. A P1 je isti ko juče :) Ticket kao što vidiš nije na knowhow području ...

Što se tiče priče iz naslova ticketa, glavno pitanja na koje ne mogu dati odgovor je pitanje koliko bi trebalo prebaciti aplikacije sa clipper-a na fmk aplikaciju u slučaju kada bi imali ovo rješenje semafora. Tu aplikaciju bi onda mogli portirati kao da je jednokorisnička dbf aplikacija.

Kod svih ranijih neuspješnih zamisli o portiranju FMK na harbour glavni problem je bilo handliranje sql tabela. Logika dbf-ova je drugačija i tačka. A Naš fmk korisnički interfejs usko je integrisan sa tom DBF logikom (tabele, unosi podataka ...) ...

Ova ideja bi napravila "tamper" u funkcijama ažuriranja dokumenata te elementarnim add, replace, delete operacijama ... a te smo hook-ove već 100 puta implementirali ... to se može implementirati.

Ukratko kada bih znao da nema nekih "bombi" koje bi naše napore rasturili u tom dijelu, mislim da bi se upuštanje u realizaciju postgresql_signal funkcija isplatilo i da bi to mogli realizovati.

Da skratim ja se bojim da bi zadatak: **portirajmo fmk na harbour kao da ćemo aplikacije koristiti isključivo u dbf jednokorisničkom okruženju** mogao biti problematičan.

Puno više nego postgresql_signal library-ja.

Ali šta god mislio, svrha ovog ticketa je ideje iz mojih sinoćnjih halucinacija zapišem jer bi se neke definitivno mogle iskoristititi.

Ti se svakako NEMOJ previše iscrpljivati i dumati nad ovim.

Jedino što bi bilo korisno, a računam da te neće puno iscrpiti :) je da daš komentar na ovaj dio:

> a skratim ja se bojim da bi zadatak: **portirajmo fmk na harbour kao da ćemo aplikacije koristiti isključivo u dbf jednokorisničkom okruženju** mogao biti problematičan.

Danas si portirao harbour/ld source code. Kolika je tu razlika ? Sjećaš li se koliki je to bio posao da se uradi port ? Jesmo li mi upošte probali ostale funkcije LD-a pa vidjeli da to ne šteka. Sjećaš li se šta ?

#25 - 06.10.2011 17:34 - Ernad Husremović

- Odgovorna osoba promijenjeno iz Ernad Husremović u Saša Vranić

#26 - 06.10.2011 18:00 - Ernad Husremović

xtuple package za hendliranje fmk podataka

ima ovdje još jedna interesantna stvar. Svi fmk podaci zahvaljujući postgresql schemama mogu ići u istu database kao xtuple.

ako je uistoj bazi, može se napraviiti xtuple package koji će kreirati sve fmk tabele ...

Može se napraviti xtuple package koji će iz tih tabela praviti reporte. ili praviti novu formu za unos ...

Ukratko podaci iz fmk klijenata bivaju dostupni i xtuple/knowhow klijentima ...

#27 - 06.10.2011 18:06 - Ernad Husremović

šta to znači ?

to znači da možemo postupnim migracijama izbacivati određene fmk tabele koje ne valjaju (npr gore pomenuta roba, kontni plan, partneri) i "podmetati" ih fmk klijentu.

Ti mostovi se mogu raditi kao postgresql trigeri koji će na serverskoj strani raditi ažuriranja tabela.

Zatim, ako sada imamo čago podatke koji su razrađeni u fmk na jednom dovoljnom nivou, možemo te podatke i rad sa fmk ostaviti ...

Sami znamo da je puno začkoljia rađeno oko fakturisanja za čagu oni njihovi rabati na cigare ...

Ovi mostovi bi omogućili da se **razbije pritisak** koji imamo **moramo preći** sa zastarjelog fmk na novi sistem

Ništa manji problem nije ni činjenica da se troši značajna energija na implementaciju **novih funkcija u fmk** što je definitivno pogrešan smjer ...

Ovdje bi mogli sve nove funkcije implementirati u knowhow/tuple sa fmk podacima !

#28 - 06.10.2011 18:22 - Ernad Husremović

joj al me ovaj unos podataka u fmk zeza

- pravim taj feature u xtuple ...

znači zajednička tačka je baza.

zbrka fmk i xtuple/knowhow podataka

u toj bazi mogu egzistirati redundantni podaci - xtuple clients, xtuple vendors i partneri ...

trigeri mogu obezbijediti automatsku sinhronizaciju tih podataka, tako da nije potrebno raditi pošto-poto mergiranje tih podataka.

trigeri će obezbijediti integritet tako da to ne mora uticati na nivo podataka (npr. ove šifre ima u fmk u knowhow nema ...)

Naravno cilj je tu redundatnost postupno eliminisati, ali je bitno da se to može na nivo baze kvalitetno riješiti bez rušenja jedne ili druge strane ...

ukratko u ovom scenariju moguća je koegzistencija i **postupan "prenos ovlasti"** sa fmk na knowhow.

#29 - 06.10.2011 18:28 - Ernad Husremović

hajmo sada "ukrasti" iz xtuple/knowhow

idemo dalje ...

xtuple ima sistem usera/role-ova riješen taman kako treba. Svaki feature se može odobriti ili zabraniti za grupu usera.

Ta stvar je dostupna i sa nivoa paketa (naravno).

Ono što je interesantna opcija pored mergiranja jeste i korištenje ovog security sistema u fmk !

sve što treba je u security event logu (i evelnt log je riješen u xtuple skroz pristojno) umjesto poziva fmk security funkcija ubaciti knowhow ...

Šta dobijamo: fmk useri se administriraju u knowhow/xtuple ... to su stvari koje sigurno nije teško implementirati niti za njih treba puno vremena.

Ukratko, možemo oplemenjivati fmk ali ne tako što ćemo razvoj usmjeriti u fmk nego što ćemo koristiti ono što je već urađeno u xtuple-u.

#30 - 06.10.2011 20:41 - Saša Vranić

Pa i nije bio neki velik posao... a što se tiče ostalih opcija pa ono radi sve u principu, sjećam se samo da su neke GET operacije štekale... ali ko zna sada šta je to.

#31 - 07.10.2011 10:54 - Ernad Husremović

- Odgovorna osoba promijenjeno iz Saša Vranić u Ernad Husremović

bijeg od fmk korisničkog interfejsa

fmk korisnički interfejs i generalno korisnički workflow su za određenu grupu korisnika super. Modalni pristup unosu podataka omogućava im da bez puno razmišljanja uvijek na isti ili sičan način obave posao.

Pri tome naravno ne želim reći da to čini korisnički interfejs fmk dobrim. Čini ga dobrim u određenom kontekstu za određenu grupu korisnika.

Koja je to grupa ? Prije svega naši dugogodišnji korisnici. Oni koji kucanjem na "fmk način" žele doživjeti penziju.

Da li pored te grupe postoje još neke primjene fmk korisničkog interfejsa ? Mišljenja sam da postoje.

<ENTER>, <ENTER> 123 <ENTER> će kod jednostavnih unosa još dugo biti "just enough" a uz to za development i održavanje najjednostavniji način.

Interfejs konzolnih aplikacija je u tim use-caseovima zato najbolji.

Zato je za idealno rješenje imati mogućeost da se određene funkcije unosa rada sa bilo kojim od korisničkih interfejsa: web HTML5, mobile - touch, konzolni "fmk" like, standardni desktop GUI.

Sve ovisno od uslova-parametara primjene.

Uzmimo poslove prodaje za primjer: veliki ili mali asortiman artikala, veličina prostora na kome se instalira računar za prodaju, da li je potrebna autonomija - rad bez napajanja, trebamo li mobilnost proizvođača - recimo da se tabletom zajedno hoda kroz prodajni salon za kupcem ... da li prodajno mjesto prljavo (ulje, vlaga ...), da li se kod izdavanja računa mogu zadavati posebni uslovi - popusti (i da to prodavac zadaje) ili je cijena uvijek ista ...

Ova masa parametara vodi tome da je nekada konzolni-dummy način unosa podataka optimalan.

Zato, bez obzira što je terminal/konzolna aplikacija za unos podataka u većini slučajeva stvar prošlosti, ona u određenim područjima može zlata vrijediti.

#32 - 07.10.2011 10:59 - Ernad Husremović

zaboravih odakle sam krenuo, aha .. "bijeg od fmk korisničkog interfejsa". Ovaj koncept nam može omogućiti da na mjestima gdje je fmk UI živa muka napravimo xtuple paket(e) koji će korisnika riješiti te muke. Ako tome pridodamo (vidi gore hajmo sada "ukrasti" iz xtuple/knowhow") integraciju xtuple user/role/security sistema u fmk dobijamo mogućnost da ti interfejsi upravo budu ono što sam maloprije pričao - mogućnost različitog izbora za korisnike

#33 - 07.10.2011 10:59 - Ernad Husremović

RNAL ramaglas kao xtuple package !?

#34 - 07.10.2011 11:01 - Ernad Husremović

vindija kalk unos kao xtuple package ?!

Sve ovo su primjeri gdje imamo da je fmk koža postala za korisnika "tijesna". A tijesna je prije svega radi količine podataka (koji PostgreSQL definitivno rješava) i konzolni način unosa podataka koji xtuple GUI također definitivno rješava.

#35 - 07.10.2011 12:55 - Ernad Husremović

- Naslov promijenjeno iz harbour semafor dbf / postgresql u san o semaforima harbour / semafor dbf / postgresql

#36 - 07.10.2011 12:56 - Ernad Husremović

- Naslov promijenjeno iz san o semaforima harbour / semafor dbf / postgresql u san o semaforima: fmk/harbour semafor dbf/postgresql

- Status promijenjeno iz Dodijeljeno u Zatvoreno

Fajlovi

dbf2pg.prg	8,64 KB	06.10.2011	Ernad Husremović
cache.prg	11,8 KB	06.10.2011	Ernad Husremović