Oninit

Introduction to Informix Enterprise Replication

By Paul Watson, Oninit LLC

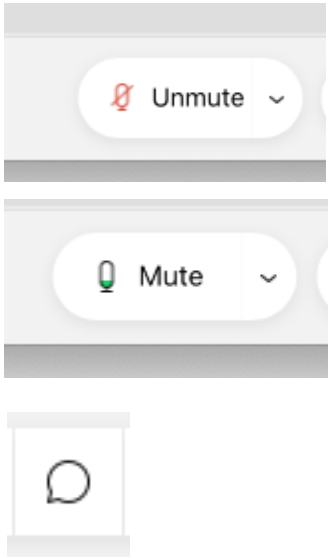
Date: Thursday, September 2, 2021, 2:00pm EDT

Informix Tech Talks by the IIUG

We have launched a new channel on YouTube for Informix Users! Please subscribe to our channel on YouTube to stay informed.

Webcast Guidelines

- **The Webcast is pre-recorded**. The replay and slides will be available on the IIUG Website
- **Please Mute your line.** Background sounds will distract everyone
- **Use the Chat Button** to ask questions



Paul Watson

Cell: +1 913 387 7529
Email: paul@oninit.com

Agenda

Basic Overview of ER
Configuration
Simple examples
Some Neat Tricks

What is ER

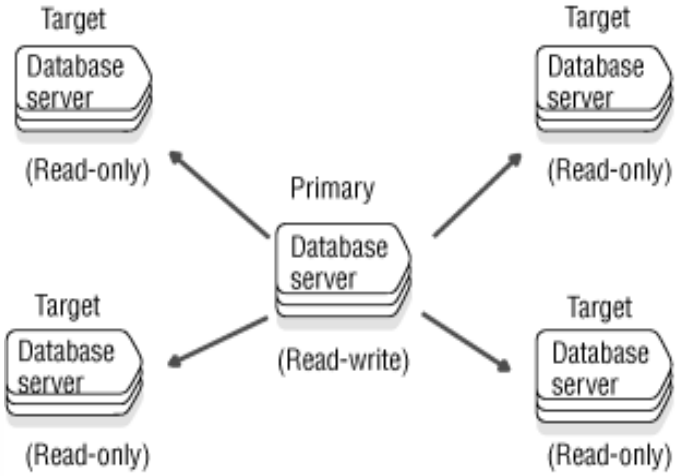
Asynchronous data replication

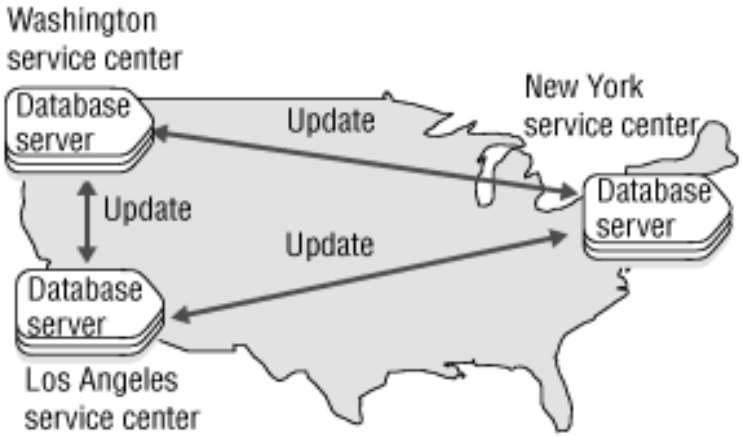
Log-based data capture

Flexible architecture

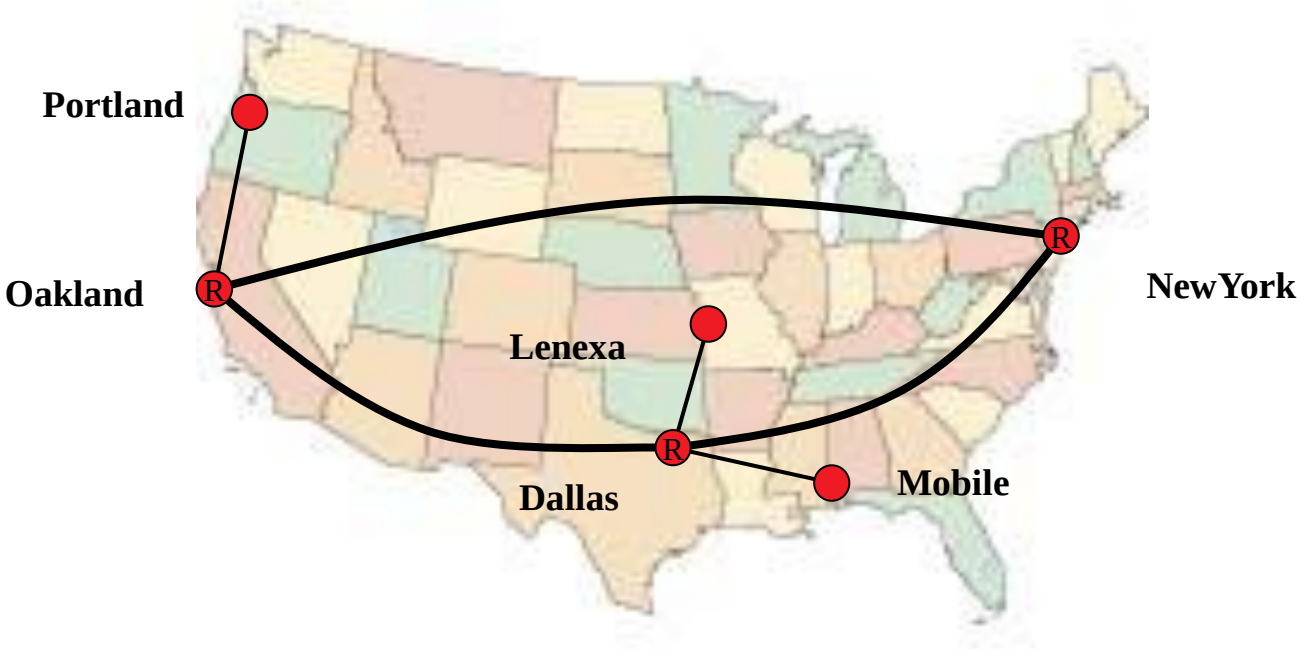
Topologies

- Supports Primary – Target and Update anywhere
- Supports various topologies
  - fully connected, hub/spoke, hierarchy, forest of hierarchy
- Supports Selective Replication
  - Row selectivity / Column selectivity





Topologies



Pre-requisites

The servers must trust each other

Configure the CDR dbspaces

Configure sqlhosts

Table requirements

- Primary Key
- Unique Index

ERKEY, CRCOLS, REPLCHECK

ONCONFIG

CDR_QUEUEMEM	Queue memory size. Default value=4MB. Recommended value= 30% of logical log size
CDR_QDATA_SBSPACE	List of queue smartblob space names separated with comma.Recommended to configure one logging and one non-logging sbspace.
CDR_DBSPACE	Syscdr database location. Defaults to rootdbs.
CDR_SERIAL	Needed only if serial column being used as primary key.

SQLHOSTS

Normal Traffic

```
theoden_net    onsoctcp        theoden              theodennet
gimli_net      onsoctcp        gimli                gimlinet
```

Replication Traffic

```
#Replication
g_theoden       group       -                -       i=1,e=theoden_repl
theoden_repl    onsoctcp    theoden-repl             oninitrepl g=g_theoden

g_gimli         group       -                -       i=2,e=gimli_repl
gimli_repl      onsoctcp    gimli-repl               oninitrepl g=g_gimli
```

Getting Started

Define server

```
cdr define server -c gimli_repl -A /data/ats -R /data/ris -I g_gimli
cdr define server -c theoden_repl -A /data/ats -R /data/ris -S g_gimli -I g_theoden
```

```
cdr list server
SERVER                  ID STATE    STATUS      QUEUE  CONNECTION CHANGED
------------------------------------------------------------------------
g_gimli              1 Active   Connected       0 Aug 11 20:08:30
g_theoden            2 Active   Connected       0 Aug 11 20:08:30
```

Getting Started

Define replicate

```
cdr define replicate -C ignore -S -M g_theoden theoden_oninit_nagios
      "oninit@g_theoden:informix.nagios" "select * from nagios"
      "oninit@g_gimli:informix.nagios" "select * from nagios"


define replicate -c  g_theoden -erkey --conflict=ignore
theoden_dbaudit_customer
       "P dbaudit@g_theoden:informix.customer"
       "select * from customer where active = 1 "
       "R dbaudit@g_gimili:informix.customer"
       "select * from customer where active = 1
```

Getting Started

-conflict or –C Specifies the rule that is used for conflict resolution.

Always: As it says

Ignore: you do not want Enterprise Replication to resolve conflicts.

Timestamp: the row or transaction with the most recent time stamp take precedence in a conflict.

Deletewins: Prevents upsert operatons

Getting Started

-u Create tables on the target if they don't exist

-k Specify the columns that are used as the replication key

-T fire the triggers on the target

Getting Started

```
cdr start repl theoden_dbaudit_customer

cdr start list theoden_dbaudit_customer

DEFINED REPLICATES ATTRIBUTES
------------------------------
REPLICATE:          theoden_dbaudit_customer
STATE:              Active ON:g_theoden
CONFLICT:           Ignore
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        dbaudit:informix.eric
OPTIONS:            transaction,fullrow
REPLID:             132276 / 0x204b4
REPLMODE:           PRIMARY  ON:g_theoden
APPLY-AS:           INFORMIX ON:g_theoden
REPLTYPE:           Master
```

Getting Started

```
cdr check repl -c g_theoden -m g_theoden -r theoden_dbaudit_customer_stats -n
theoden_dbaudit_customer_stats g_gimli


Aug 12 2021 12:04:02 --   Table scan for theoden_dbaudit_customer_stats start  ----

Node                    Rows     Extra   Missing  Mismatch Processed
----------------- --------- --------- --------- --------- ---------
g_theoden          72323964     0         0         0         0
g_gimli            72322241     0      1453         0         0

Aug 12 2021 12:04:02 --   Table scan for theoden_dbaudit_customer_stats end   ----
```

Getting Started

```
cdr check repl -c g_theoden -R -m g_theoden -r theoden_dbaudit_customer_stats -n
theoden_dbaudit_customer_stats g_gimli



Aug 12 2021 12:04:02 --   Table scan for theoden_dbaudit_customer_stats start  ----

Node                    Rows     Extra   Missing  Mismatch Processed
--------------- --------- --------- --------- --------- ---------
g_theoden          72323964     0         0         0       1453
g_gimli            72322241     0      1453         0          0

Aug 12 2021 12:04:02 --   Table scan for theoden_dbaudit_customer_stats end   ----
```

Getting Started

```
cdr stats check   theoden_dbaudit_customer_stats -r 2

-------------------------------------------------
Job tst
repl1             Started Jan 17 16:10:59
**********************************----+----+----+ Remaining 0:10:0

-------------------------------------------------
Job tst
repl1             Started Jan 17 16:10:59
*********************************************+ Remaining 0:01:01

-------------------------------------------------
Job tst
repl1             Completed
                  Started Jan 17 16:10:59, Elapsed Time 0:00:07
```

Getting Started

ontape/onbar

cdr check sec2er and cdr start sec2er

cdr migrate

Getting Started

That's about all there is to it ……

Replicatesets

Table requirements

- Primary Key
- Unique Index
- ERKEY

- REPLCHECK

Gotchas/Tips

Don’t just drop the syscdr database
Performance issues – blame the target
Export MyTrace=1 to show the SQL
Designate a ‘master’
Have lots and lots of logspace
Avoid very large transactions
Onstat is your friend
Upgrades

ERKEY and REPLCHECK are slow alters

Neat Tricks

Adding the additional columns are a slow alter

- Can take forever
- Table Locks
- Can Cause LTX

So we need to do the change as an IPA, but how?

ERKEY

Add the following columns

```
ALTER TABLE mytab ADD
    (
            ifx_erkey_1     integer default 0,
            ifx_erkey_2     integer default 0,
            ifx_erkey_3     smallint default 0
    );
```

And populate

ERKEY – populate

```
FOREACH c_mytab WITH HOLD FOR
            SELECT erkey_1, erkey_2, erkey_3
            INTO key1, key2, key3
            FROM mytab

            IF txnsize = 0 THEN
                        BEGIN WORK;
            END IF
            IF offset = (-1 * blksize) THEN
                        LET offset = 0;
            END IF

            LET key1 = 1;
            LET key2 = (tnum/blksize)::INT - offset;
            LET key3 = 1;
            UPDATE mytab
            SET (ifx_erkey_1, ifx_erkey_2, ifx_erkey_3) = (key1, key2, key3)
            WHERE CURRENT OF c_mytab;

            LET offset = offset + 1;
            LET txnsize = txnsize + 1;
            if  txnsize > 1000000 then
                        COMMIT WORK;
                        LET txnsize = 0;
            end if;

            LET tnum = tnum - 1;
END FOREACH;
```

ERKEY – the magic

```
begin work;
lock table mytab in exclusive mode;

create unique index er_mytab on mytab (ifx_erkey_1,
        ifx_erkey_2, ifx_erkey_3)
fragment by expression
(mod(ifx_erkey_2 , 7 ) in (-1, 0 )) in eridx01_dbs, …;

alter table mytab
add constraint primary key
(
        ifx_erkey_1,
        ifx_erkey_2,
        ifx_erkey_3
) constraint pk_${TABNAME};

….
```

ERKEY – the magic

```
update syscolumns set colattr = 17
       where tabid = (select tabid from systables
              where tabname = "mytab")
       and colname = "ifx_erkey_1";

update syscolumns set colattr = 33
       where tabid = (select tabid from systables
              where tabname = "mytab")
       and colname = "ifx_erkey_2";

update syscolumns set colattr = 65
       where tabid = (select tabid from systables
              where tabname = "mytab")
       and colname = "ifx_erkey_3";

commit work;
```

REPLCHECK

Add the following columns

```
ALTER TABLE mytab ADD
(
        ifx_replcheck  bigint
);


UPDATE syscolumns
SET colattr = 9
WHERE tabid = tid and colname = 'ifx_replcheck';
```

REPLCHECK – populate

Internally CDR uses ifx_checksum() function (ish)

```
ifx_checksum(col1,
   ifx_checksum(col2,
     ifx_checksum(col3,
       ifx_checksum(coln, 0)))))…)
```

You might see casting errors on basic datatype but

```
ifx_checkum(colx::lvarchar)
```

Note: export MyTrace = 1 and running cdr check command will (should) show the SQL being run

Questions

paul@oninit.com
+1 913 387 7529

Thank You

Informix Tech Talks by the IIUG on YouTube

Please subscribe to our channel on YouTube to stay informed. This will be a place for Informix how-to videos.

Subscribe at:
https://www.youtube.com/c/InformixTechTalksbytheIIUG