# Using the Co:Z Launcher to Enable zEnterprise Distributed Workloads

June 21, 2011

Steve Goetze
Kirk Wolf

DovetailedTechnologies

http://dovetail.com
info@dovetail.com

# Dovetailed Technologies

We provide z/OS customers world wide with innovative solutions that enhance and transform traditional mainframe workloads:

- Co:Z Co-Processing Toolkit for z/OS
- OpenSSH Accelerator for z/OS
- T:Z Quickstart for Tomcat and z/OS
- JZOS - acquired by IBM in 2005 and now part of the z/OS Java SDK

# Co:Z Components

- Reusable record streaming library
- Co:Z SFTP
  - OpenSSH SFTP with z/OS exploitation
- Co:Z Batch
  - full featured BPXBATCH replacement
- Co:Z Dataset Pipes
  - convert datasets to streams / streams to datasets
  - other utilities
- Co:Z Launcher
  - z/OS batch cooperative processing (distributed apps+data)
- Co:Z Target System Toolkit
  - used with Co:Z Launcher and Dataset Pipes
  - currently distribute binaries for Windows, Linux [ Intel, System z ]
  - open source - customers can build on other POSIX platforms

# Synergy Between Components

- Reusable record streaming library
  - SFTP, Dataset Pipes, Launcher
- Co:Z Batch
  - Dataset Pipes (Local)
  - SFTP client batch job scripting
- Co:Z Target System Toolkit
  - Dataset Pipes (Remote initiated)
- Co:Z Launcher
  - Dataset Pipes (Job step initiated)
- SAF/RACF certificate support
  - Launcher, SFTP

# CO:Z Batch

- Similar to z/OS BPXBATCH Utility
  - run a Unix command or shell in a batch job step
  - return code adopted from exit code
  - STDOUT and STDERR DDs to either datasets or USS files
- Unlike BPXBATCH
  - //STDIN input can come from dataset or DD *
  - Can launch a login shell in the same address space
    - profile scripts are automatically run by a login shell

# Co:Z Dataset Pipes

- z/OS Unix shell commands for converting record-oriented datasets to Unix pipes
  - fromdsn – read a dataset and write to stdout
  - todsn – read from stdin and write to a dataset
- Can be invoked in three modes:
  - locally (batch or USS)
  - remote (via SSH client)
  - remotely from within Co:Z launched process
- Extensive set of data conversion options

# Co:Z Batch and Dataset Pipes

```
// EXEC PGM=COZBATCH  run a Unix login shell
//STDOUT  DD SYSOUT=*
//STDERR  DD SYSOUT=*
//STDIN   DD *
 #runs locally under USS
 fromdsn '//sys1.maclib(acb)' |
  sed '/\.\*/d' | # strip comment lines
  todsn //DD:OUTPUT
//OUTPUT  DD DSN=&&TEMP,DISP=(NEW,PASS),...
```

# fromdsn / todsn examples

```
todsn //MVS1.OUTPUT.DATASET < /home/user/file
```

```
ps -ef | todsn -o 'recfm=fb,lrecl=80' //MY.DSN
```

```
echo 'add line to end' | todsn -a //LOG.DATA
```

```
fromdsn //DD:EBCDIC |
  todsn -t ISO8859-1 //DD:ASCII
```

```
fromdsn -l rdw //BINARY.DATA > rdw.bin
```

# fromdsn / todsn examples

```
fromdsn -x shr //mvs1.input.dataset >
  /home/user/mydata


fromdsn //input.data |
  grep '^A1' |
  todsn -x 'new like(input.data)' //filter.data


netstat -n |
  todsn -x 'sysout(6) writer(CDKK0201)
  dest(ARCHT) spin(unalloc)'   //SYSOUT
```

# Dataset Pipes (Remote Initiated)

```
fromdsn -ssh user@zos.myco.com \
  //mvs1.input.dataset > c:\mydata\data1.txt
```

```
todsn -ssh user@zos.myco.com //my.dsn \
   < /var/logs/mylog
```

```
fromdsn -ssh user@zos.myco.com \
  -b -l rdw //mvs1.binary.dat > /tmp/rdw.bin
```

```
todsn -ssh user@zos.myco.com \
  -x 'sysout(a) writer(intrdr) recfm(f)
  lrecl(80)' //SYSOUT < job.jcl
```

# Co:Z Launcher

- Launch a process on a remote system from a batch job step using an SSH connection
- Input is redirected from DD STDIN, output is redirected to DD STDOUT and STDERR
- Remote shell exit code is adopted as job step condition code
- todsn and fromdsn commands in remote process can access job's DDs and datasets
- z/OS console commands can monitor, control, and send input  to remote process

# Co:Z Launcher and Dataset Pipes

```
// EXEC PGM=COZPROC,ARGS='u@linux.myco.com'
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDIN    DD *
#Runs on u@linux.myco.com
fromdsn '//sys1.maclib(acb)' |
  sed '/\.\*/d' | # strip comment lines
  todsn //DD:OUTPUT
//OUTPUT   DD DSN=&&TEMP,DISP=(NEW,PASS),...
```

# Co:Z Launcher example

```
//STEP1   EXEC PROC=COZPROC,
//        ARGS='myuid@linux.myco.com'
//STDIN   DD *
#Runs remotely on linux.myco.com
echo "We are running on: " `uname -sr`

cat /var/log/syslog.0 |
  todsn -w trunc \
        -x 'like(sys2.logs.model)'  \
        '//sys2.logs.linux1(+1)'
//
```

# wget | awk | DB2 load

```
//STEP1   EXEC PROC=COZPROC,ARGS='u@dmz1.myco.com'
//STDIN   DD *
wget -O-ftp://ftp.visi.com/Congress.db.txt |
awk -F "\t" -v sq="'" '{
  if (NR == 1)  #skip header/empty table
    print "DELETE FROM CONGRESS;"
  else {
    print "INSERT INTO CONGRESS VALUES("
    print sq $1 sq ", "
    print sq $2 sq ", "
    print sq $4 sq ", "
    print sq $3 sq
    print ");"
  }'
}
```

# wget | awk | DB2 load

```
//STDOUT DD DSN=&&SPUFIN,DISP=(NEW,PASS),
//       SPACE=(CYL,(2,1)),
//       DCB=(RECFM=FB,LRECL=80)
//STEP2    EXEC PGM=IKJEFT01,
//              DYNAMNBR=20,COND=(0,NE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD *
DSN SYSTEM(DBS1)
RUN PROGRAM(DSNTEP2) PLAN(DSNTEP71)
  LIB('DB2V810.RUNLIB.LOAD')
END
//SYSIN    DD DSN=&&SPUFIN,DISP=(OLD,DELETE)
//
```

# Co:Z – File transfer appliance example

```
//STEP1    EXEC PROC=COZPROC,
//         ARGS='myuid@linux1.myco.com'
//TRANS1   DD DSN=TRANS1.DATA,DISP=SHR
//STDIN    DD *
fromdsn -b //DD:TRANS1 |
  gzip -c - |
  gpg -r key-1 --batch --output=- --encrypt=- |
  curl -T- http://rhost.com/upload?partner=023
//
```

- Compress, encrypt and upload a dataset via HTTP
- Leverage open source tools on a secure file transfer gateway
- Data is never stored on gateway disk
- Pipelines are concurrent

# Co:Z Launcher “R” Example

```
// EXEC PROC=COZPROC,ARGS='u@linux.myco.com'
//LIFEEXP DD DISP=SHR,DSN=KIRK.LIFEEXP.DATA
//STDIN DD *
#Run the R statistical system with inline commands
#This example is a 2-dimensional multiple regression

R --no-save  <<EOB
mypipe = pipe("fromdsn DD:LIFEEXP", open="r")
life = read.table(mypipe, header=TRUE)
close(mypipe)
multilinearFit =
  lm(LifeExp~PeoplePerTV+PeoplePerDoctor,data=life)
summary(multilinearFit)
EOB
//
```

# Co:Z Launcher data tunneling

- By default, data connections for remote todsn/fromdsn commands are tunneled in the SSH connection

- On secure networks, data connections can be configured to use clear channel sockets. Ideal for exploiting:
  - zBX intra-ensemble data network with minimum overhead / maximum performance
  - z/OS to Linux for System z hipersockets

- On unsecure networks where encryption is required, consider using “OpenSSH Accelerator for z/OS”

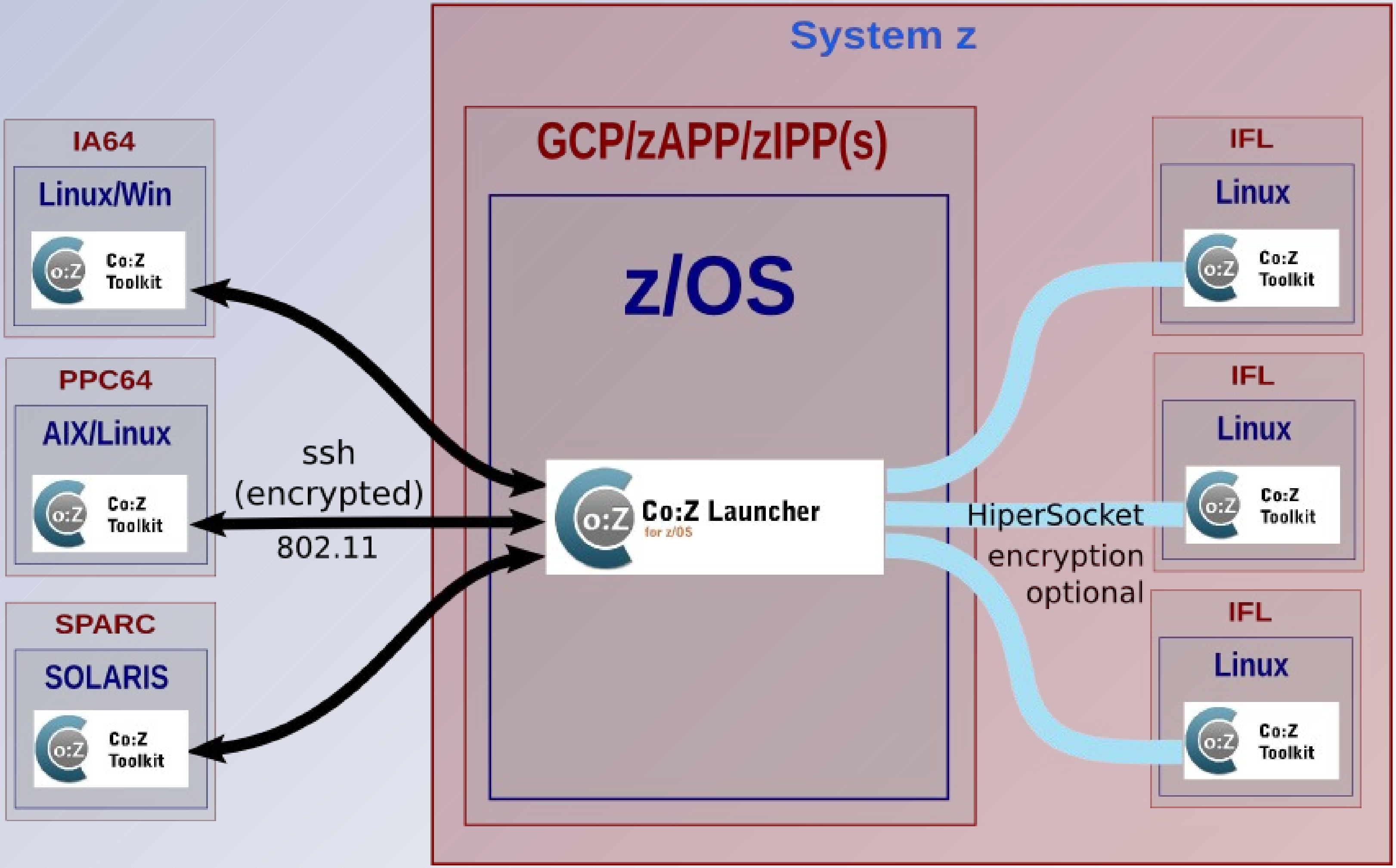
# Clear channel example

```
//COZSMF  JOB (),'COZ'
//DUMPSMF  EXEC PGM=IFASMFDP
//SYSPRINT DD   SYSOUT=*
//SMFDATA  DD   DISP=SHR,DSN=SYS1.SMF.DATA
//SMFOUT   DD   SYSOUT=*
//OUTDD    DD   DSN=&&SMFUNLD,DISP=(NEW,PASS),
//              UNIT=SYSDA,SPACE=(CYL,(20,20))
//SYSIN    DD   *
    INDD(SMFDATA,OPTIONS(DUMP))
    OUTDD(OUTDD,TYPE(000:255))
//RUNCOZ EXEC PROC=COZPROC,ARGS='u@linux.myco.com'
//SMFUNLD  DD   DSN=&&SMFUNLD,DISP=(OLD,DELETE)
//COZCFG   DD   *
ssh-tunnel=false
//STDIN    DD   *
fromdsn -b -l rdw //DD:SMFUNLD |
  smfp   # piped into an SMF processing program
```

# Co:Z Target System Toolkit

- Client commands, written in portable C++
  - todsn, fromdsn, cozagent, …
- Binary packages:
  - Windows
  - Linux LSB RPMs for Intel, System z
  - AIX Power package (tested on zBX Model 002)
  - Solaris
- Open Source package:
  - uses standard configure/make/make install

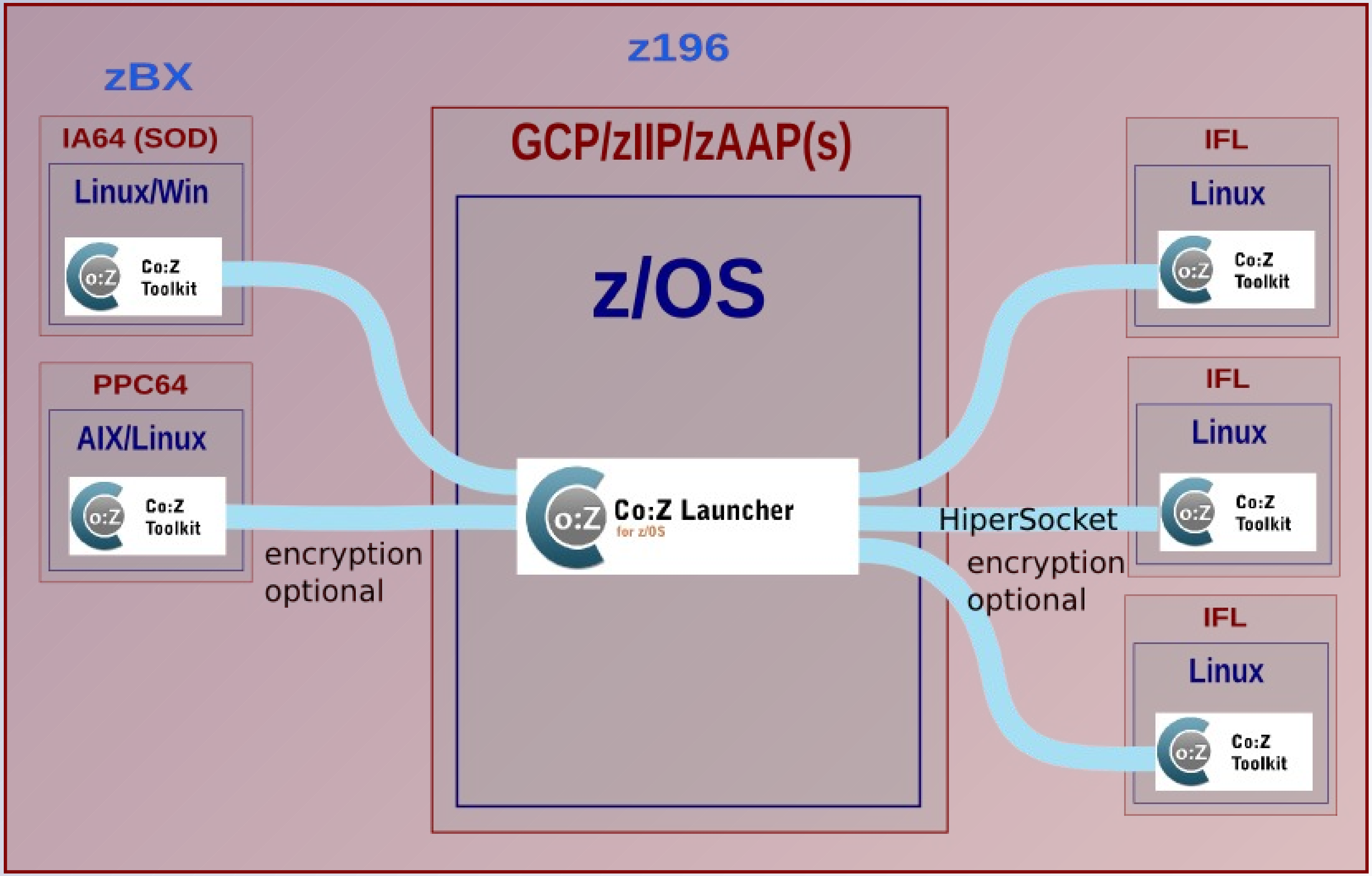
# z/OS cooperative processing with Co:Z



# Co:Z Launcher Success Story

- Generating PDFs from XML dataset in z/OS batch job
  - Java application using open source iText framework
  - 59 docs/minute on existing 2094-405 workload
- Added (1) zIIP (zAAP mode)
  - 80 docs/minute
- Co:Z Launcher
  - simple JCL change; no program changes
  - via HiperSocket to VM/Linux IFLs
  - exploiting **ssh-tunnel=false** feature
  - >900 docs/minute
- z/OS operator training
  - don't cancel job if it doesn't use CPU time
  - Java program changed to add progress messages; viewable in sysout/SDSF.

# Co:Z z196 + zBX Exploitation



# Co:Z Launcher Benefits

- Manage remote processes from z/OS batch
- Leverage open source technologies from z/OS
- Expanded batch application workloads
  - z/OS step-to-step flow control and dataset passing
  - Unix process flow and pipes
  - Fully integrated cross-platform applications
  - Easy to use
  - High performance
  - Secure

# OpenSSH Accelerator for z/OS

- Allows customers to create accelerated versions of IBM Ported Tools OpenSSH “ssh” and “sshd” modules.
- Uses CPACF instructions if available for OpenSSH algorithms with fall back to original software routines.
- Exploits the highly modular OpenSSL architecture to install hooks into the internal “EVP” engine interfaces using the z/OS program binder.
- May reduce ssh CPU time by 67%; overall time for sftp by > 60% (YMMV)
- Not supported by IBM.
  - Instructions included for selecting the original or accelerated versions in your jobs.  Original versions should be used when obtaining IBM service for Ported Tools OpenSSH.