2012

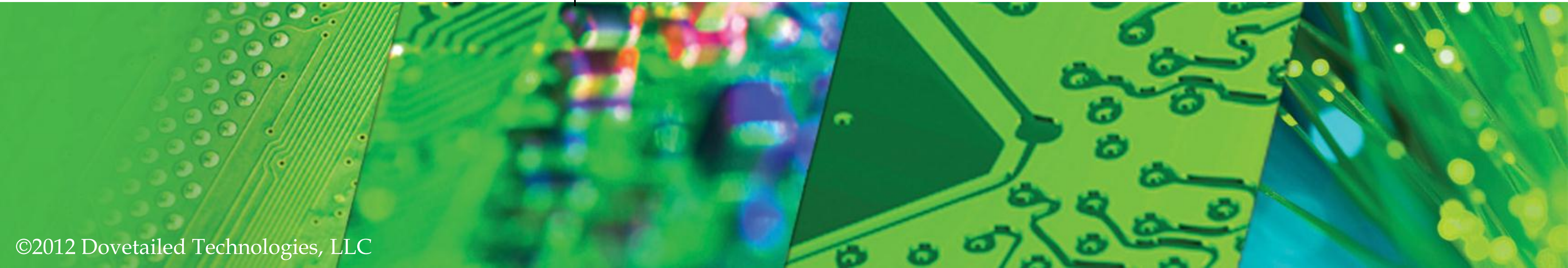
**IBM System z Technical University**

Enabling the infrastructure for smarter computing

# z/OS Hybrid Batch Processing on the zEnterprise

## zES30

Steve Goetze steve@dovetail.com
Kirk Wolf kirk@dovetail.com



# Trademarks

- Co:Z® is a registered trademark of Dovetailed Technologies, LLC
- z/OS®, zEnterprise®, and zBX® are registered trademarks of IBM Corporation
- SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.
- Oracle ® and Java ® are registered trademarks of Oracle and/or its affiliates
- iText ® is a registered trademark of 1T3XT BVBA

# Agenda

- Define Hybrid Batch Computing
- Hello World Example
- Applications
- Load balancing Hybrid Batch workloads
- Summary / Wrap Up

# zEnterprise Hybrid Computing

- A *System of Systems*
  - ✓ Combined technology platforms: zSeries, POWER, x86
  - ✓ Capable of hosting many workloads integrated together
  - ✓ Managed as a single entity

"IBM's new hybrid z/blade environment is really a new governance arrangement between the z world and the distributed systems world."

-- Jeff Frey, IBM Fellow

# What Are the Implications for z/OS?

"The sweet spot for z/OS is highly integrated applications for which transactional integrity, recoverability and data proximity are critical."

-- Jeff Frey, Jose Castano

# zEnterprise Hybrid Computing Models

Well Known:

- zBX/zLinux as user-facing edge, web and application servers
  - ✓ z/OS provides back-end databases and transaction processing
- zBX as special purpose appliances or optimizers
  - ✓ DB2 Analytics Accelerator
  - ✓ DataPower

Another Model: **z/OS Hybrid Batch**

- zBX/zLinux integrated with z/OS batch

# z/OS Hybrid Batch Processing

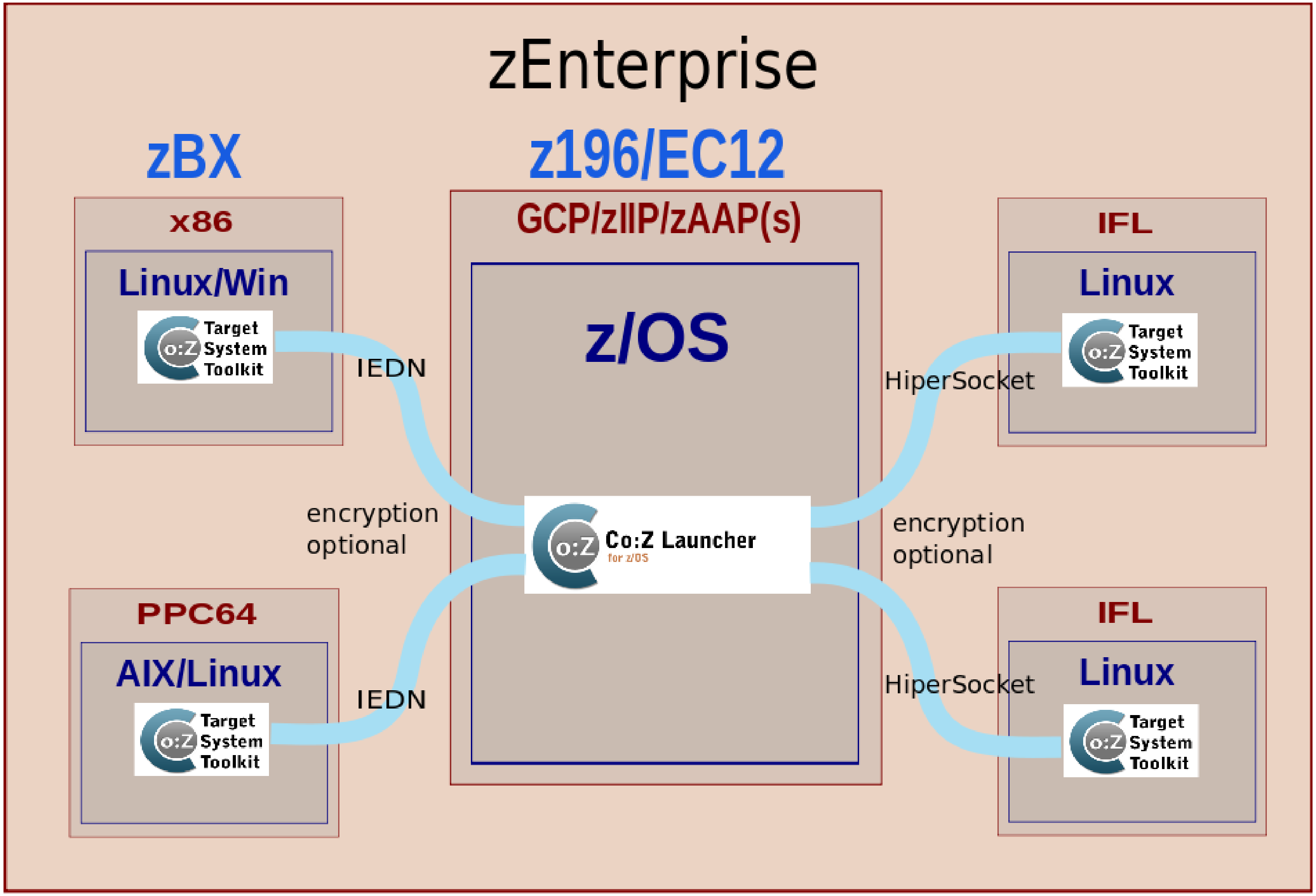
1. The ability to execute a program or script on a virtual server from a z/OS batch job step
2. The target program may already exist and should require little or no modification
3. The target program's input and output are redirected from/to z/OS spool files or datasets
4. The target program may easily access other z/OS resources: DDs, data sets, POSIX files and programs
5. The target program's exit code is adopted as the z/OS job step condition code

Requires new enablement software…

# Co:Z Co-Processing Toolkit

- Implements z/OS Hybrid Batch model
- Co:Z Launcher starts a program on a target server and automatically redirects the standard streams back to jobstep DDs
- The target program can use Co:Z DatasetPipes commands to reach back into the active jobstep and access z/OS resources:
  - ✓ **`fromdsn/todsn`** – read/write a z/OS DD or data set
  - ✓ **`fromfile/tofile`** – read/write a z/OS Unix file
  - ✓ **`cozclient`** – run z/OS Unix command
- Free (commercial support licenses are available)

# Co:Z Hybrid Batch Processing



# Hybrid Batch – Hello World

- Simple example illustrating the principles of Hybrid Batch Processing
- Launch a process on a remote Linux server
  - ✓ Write a message to stdout
  - ✓ In a pipeline:
    - ▫ Read the contents of a dataset from a jobstep DD
    - ▫ Compress the contents using the Linux gzip command
    - ▫ Write the compressed data to the z/OS Unix file system
  - ✓ Exit with a return code that sets the jobstep CC

## Linux on z / zBX

## z/OS

## Linux on z / zBX

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *
echo “Hello $(uname)!”
fromdsn -b DD:INPUT |
gzip  -c |
tofile -b /tmp/out.gz
exit 4
//
```

## Linux on z / zBX

```
echo “Hello $(uname)!”
fromdsn –b DD:INPUT | <------
gzip  –c |
tofile –b /tmp/out.gz
exit 4
```

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *




//
```

## Linux on z / zBX

```
echo “Hello $(uname)!”
fromdsn –b DD:INPUT |
gzip  –c |
tofile –b /tmp/out.gz
exit 4
```

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *





//
```

## Linux on z / zBX

```
echo “Hello Linux!”
fromdsn –b DD:INPUT |
gzip  –c |
tofile –b /tmp/out.gz
exit 4
```

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *




//
```

## Linux on z / zBX

```
echo “Hello Linux!”
fromdsn –b DD:INPUT |
gzip  –c |
tofile –b /tmp/out.gz
exit 4
```

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *



//
```

## Linux on z / zBX

```
echo “Hello Linux!”
fromdsn -b DD:INPUT |
gzip  -c |
tofile -b /tmp/out.gz
exit 4
```

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *




//
```

## Linux on z / zBX

```
echo “Hello Linux!”
fromdsn -b DD:INPUT |
gzip  -c |
tofile -b /tmp/out.gz
exit 4
```

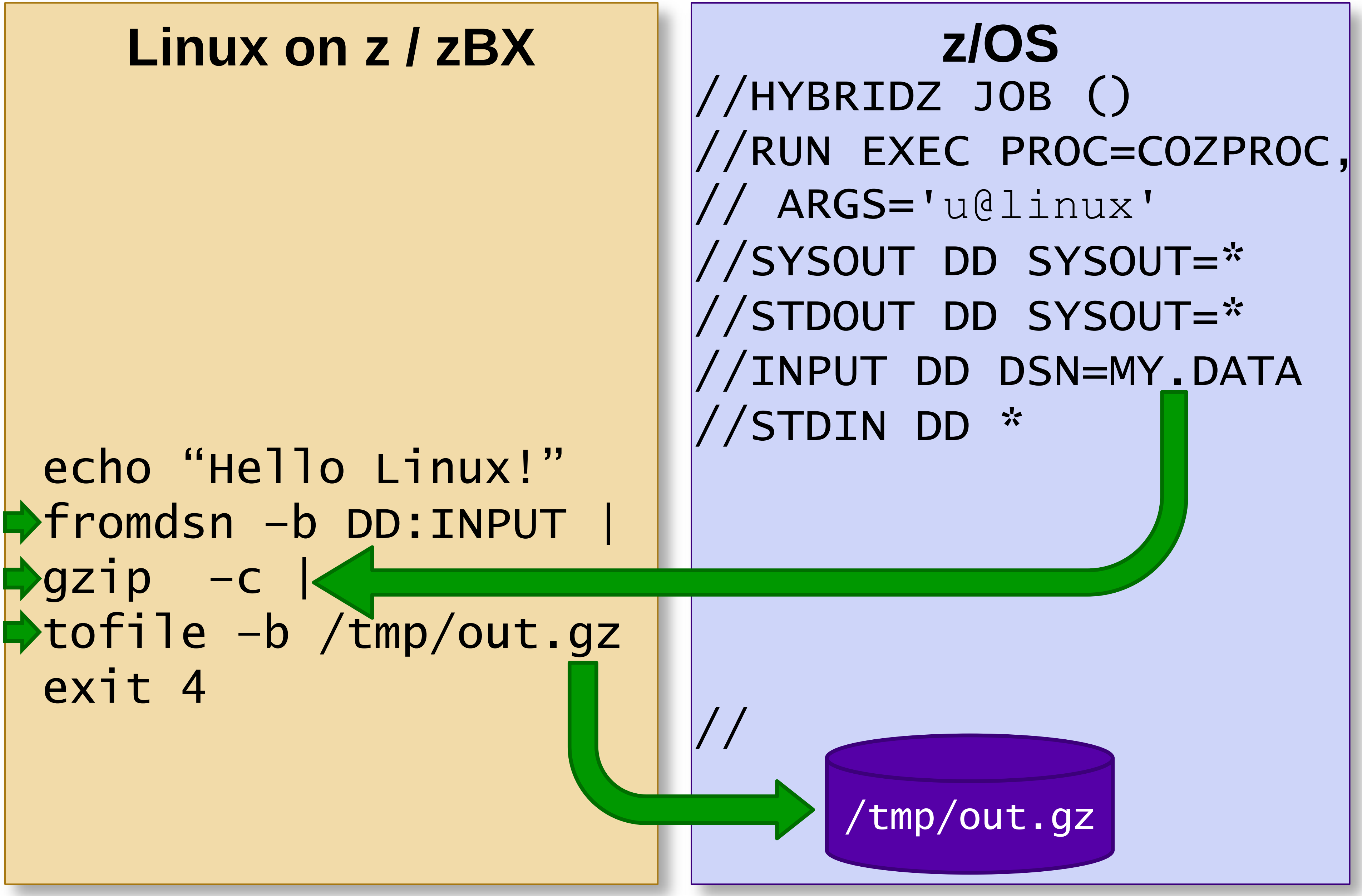
## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *





//
```

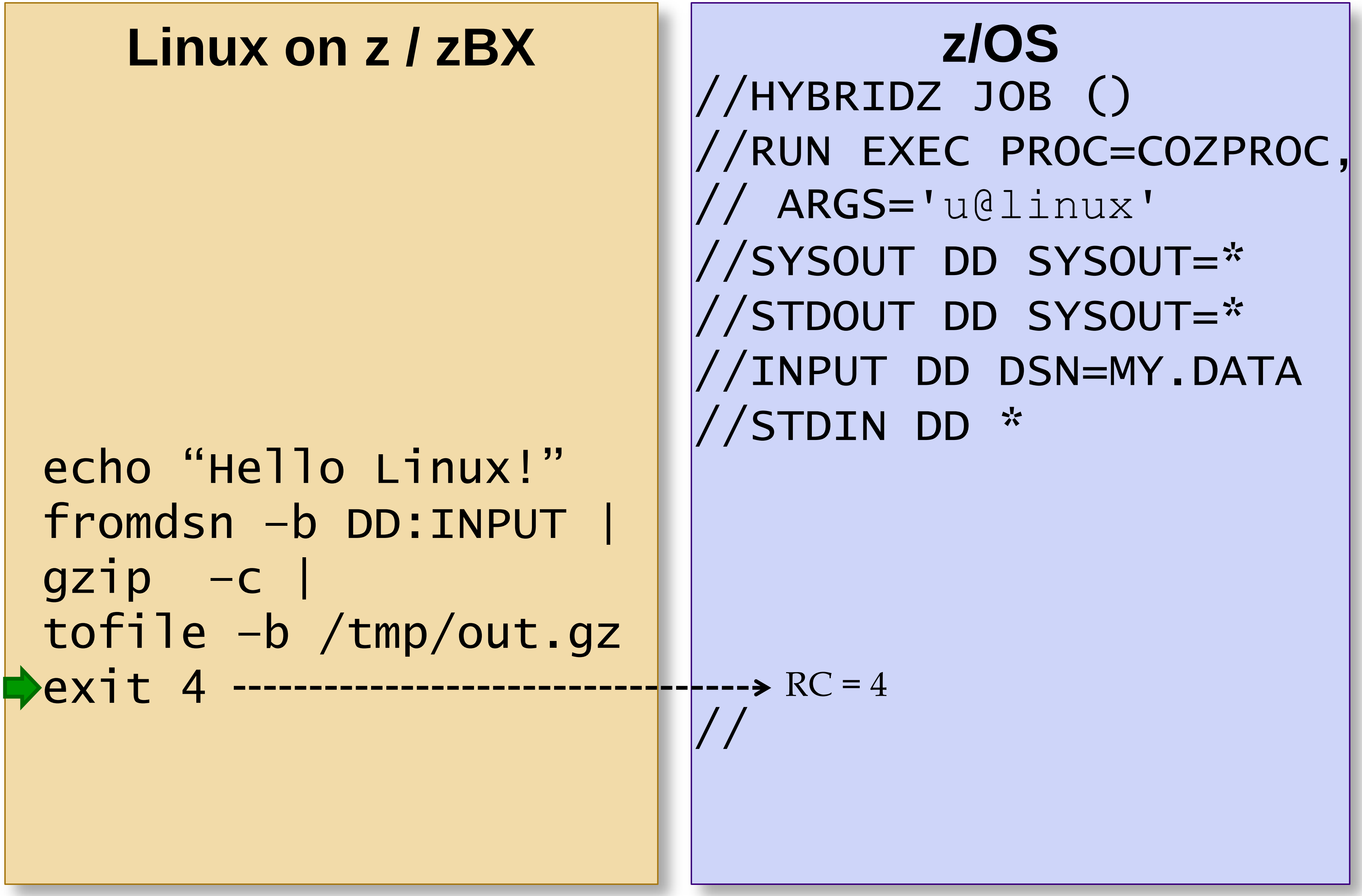


## Linux on z / zBX

```
echo “Hello Linux!”
fromdsn –b DD:INPUT |
gzip  –c |
tofile –b /tmp/out.gz
exit 4
```

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *
//
```



## Linux on z / zBX

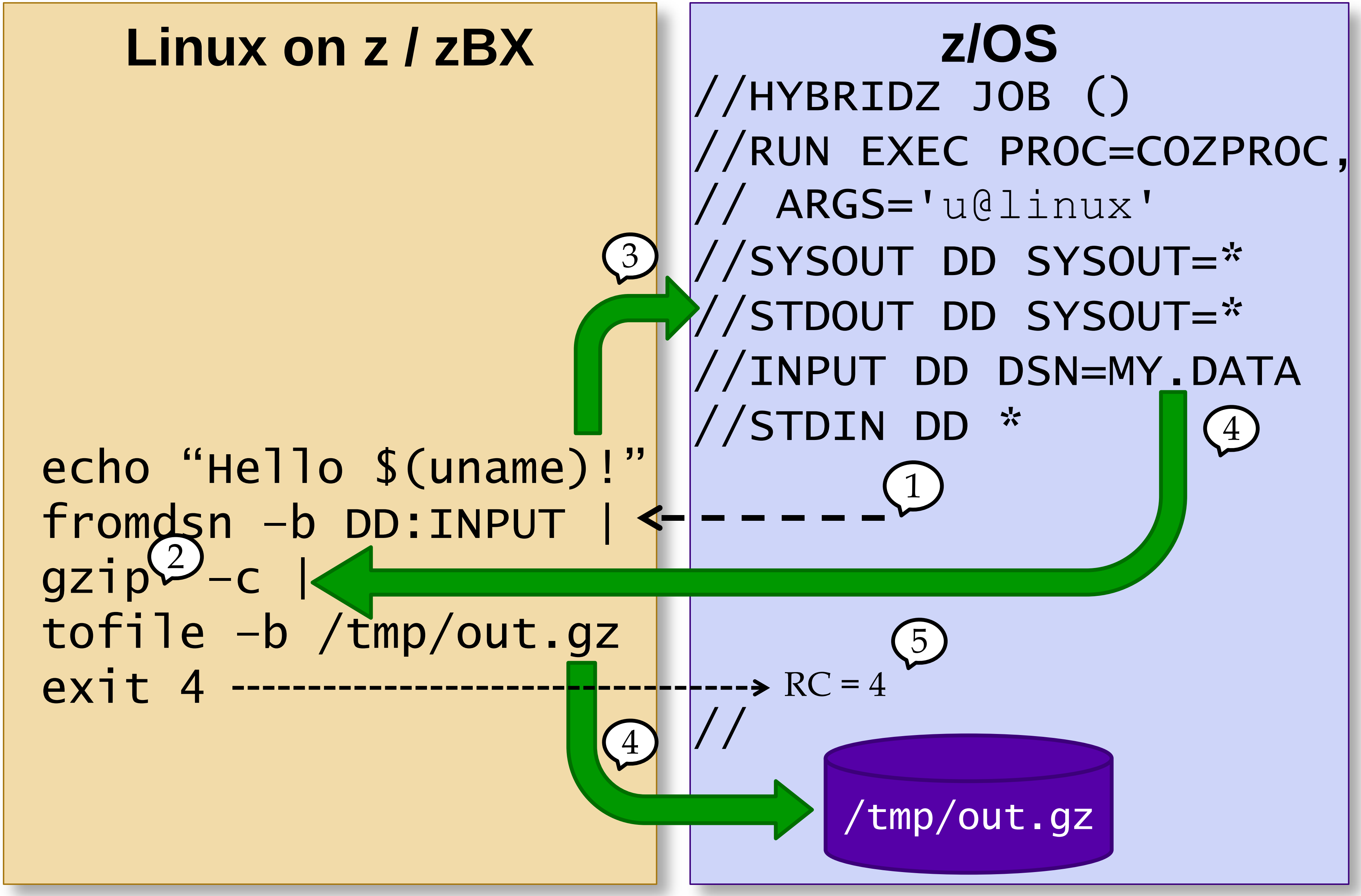
```
echo “Hello Linux!”
fromdsn –b DD:INPUT |
gzip  –c |
tofile –b /tmp/out.gz
exit 4
```

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *
```

RC = 4

```
//
```



## Linux on z / zBX

```
echo “Hello $(uname)!”
fromdsn -b DD:INPUT |
gzip -c |
tofile -b /tmp/out.gz
exit 4
```

## z/OS

```
//HYBRIDZ JOB ()
//RUN EXEC PROC=COZPROC,
// ARGS='u@linux'
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD DSN=MY.DATA
//STDIN DD *
//
```

# Hello World:  Hybrid Batch

1. A script is executed on a virtual server from a z/OS batch job step
2. The script uses a program that already exists -- gzip
3. Script output is redirected to z/OS spool
4. z/OS resources are easily accessed using fromdsn, tofile, etc…
5. The script exit code is adopted as the z/OS job step CC

# Hello World – DD:STDOUT

```
Hello Linux!
```

# Hello World – DD:SYSOUT

```
CoZLauncher[N]: version: 2.2.0 2012-09-01
cozagent[N]: version: 1.1.0 2012-03-16
fromdsn(DD:STDIN)[N]: 5 records/400 bytes read…
fromdsn(DD:INPUT)[N]: 78 records/6240 bytes read…
tofile(/tmp/out.gz)[N]: … 1419 bytes written
todsn(DD:STDOUT)[N]: … 13 bytes written
todsn(DD:STDERR)[N]: … 0 bytes written
CoZLauncher[E]: u@linux target … ended with RC=4
```

# Hello World – DD:JESMSGLG

```
JOB01515 ---- FRIDAY,  7 SEPT 2012 ----
JOB01515  IRR010I  USERID GOETZE   IS ASSIG…
JOB01515  ICH70001I GOETZE   LAST ACCESS AT…
JOB01515  $HASP373 HYBRIDZ  STARTED – INIT…
JOB01515  -
JOB01515  -STEPNAME PROCSTEP    RC   EXCP…
JOB01515  -RUN      COZLNCH     04   1345…
JOB01515  -HYBRIDZ  ENDED.  NAME-
JOB01515  $HASP395 HYBRIDZ  ENDED
```

# Co:Z Data Security

- Remote processes are securely launched using proven OpenSSH technology
- Access to z/OS resources controlled by launching userid's SAF profile
- By default, data transfer is tunneled (encrypted) over the ssh connection
  - ✓ Optionally, data can be transferred over raw sockets (option: ssh-tunnel=false)
    - This offers very high performance without encryption costs
    - Ideal for a secure network, such as zEnterprise HiperSockets or IEDN

# Hybrid Batch Examples

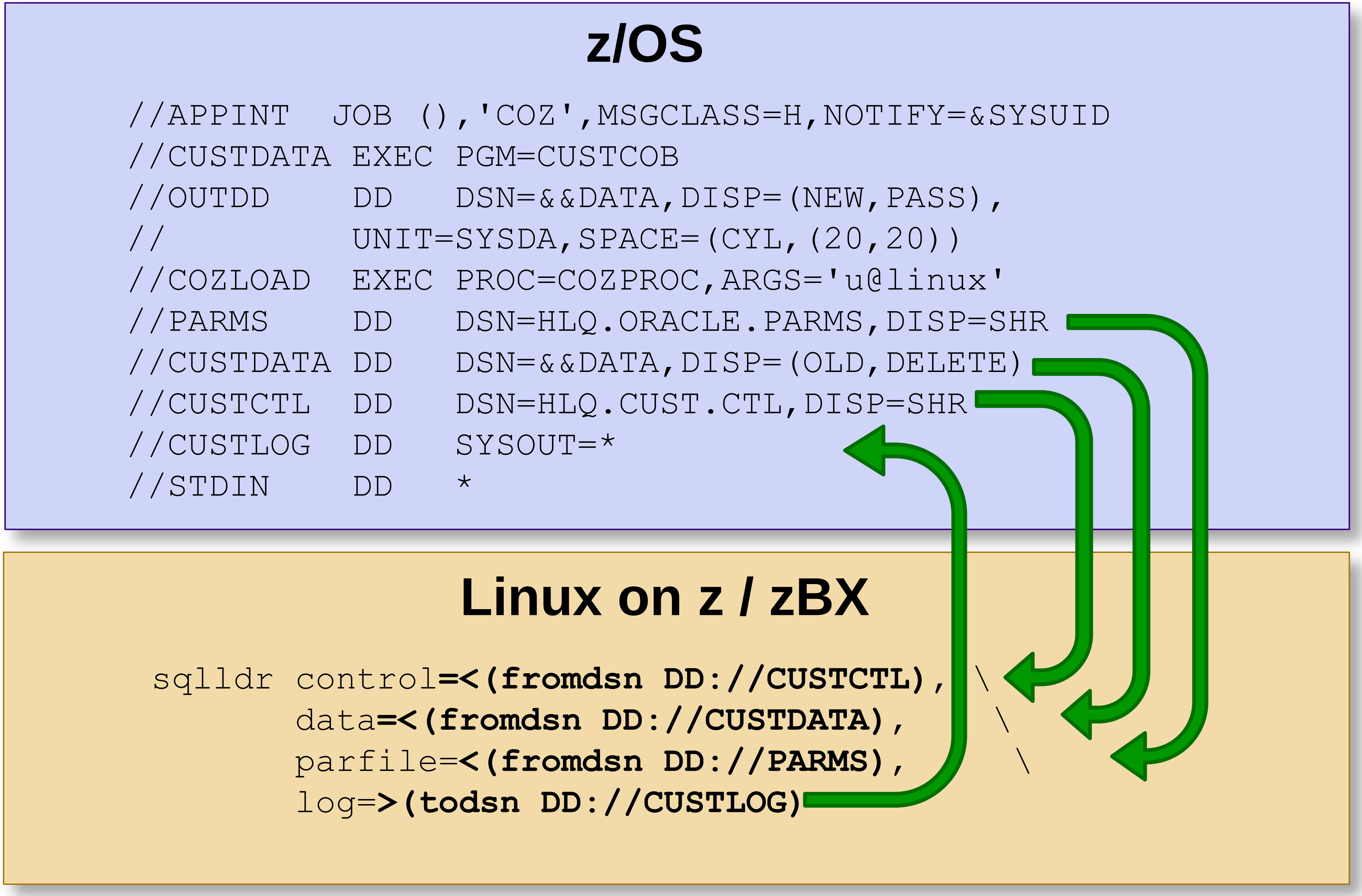
- Virtual Server Batch Administration
  - ✓ Data integration between platforms
  - ✓ e.g. maintain Oracle databases from z/OS batch
- "Fit For Purpose"
  - ✓ Moving z/OS SAS Programs to the Linux blade
  - ✓ Moving resource intensive (PDF generation) application to the Linux/Windows blade

# Virtual Server Batch Administration

- Extend z/OS batch schedules to maintain virtual servers
  - ✓ AKA "Herding Penguins"
- More than an Enterprise Scheduler
  - ✓ Enables cooperative data exchange
- Retains full operational control from z/OS even as hybrid computing model expands

# Data Administration Example

```
//APPINT  JOB (),'COZ',MSGCLASS=H,NOTIFY=&SYSUID
//CUSTDATA EXEC PGM=CUSTCOB
//OUTDD    DD   DSN=&&DATA,DISP=(NEW,PASS),
//              UNIT=SYSDA,SPACE=(CYL,(20,20))
//COZLOAD  EXEC PROC=COZPROC,ARGS='u@linux'
//CUSTCTL  DD   DSN=HLQ.CUST.CTL,DISP=SHR
//CUSTDATA DD   DSN=&&DATA,DISP=(OLD,DELETE)
//CUSTLOG  DD   SYSOUT=*
//PARMS    DD   DSN=HLQ.ORACLE.PARMS,DISP=SHR
//STDIN    DD   *
sqlldr control=<(fromdsn DD://CUSTCTL), \
       data=<(fromdsn DD://CUSTDATA), \
       parfile=<(fromdsn DD://PARMS), \
       log=>(todsn DD://CUSTLOG)
```



# z/OS

```
//APPINT  JOB (),'COZ',MSGCLASS=H,NOTIFY=&SYSUID
//CUSTDATA EXEC PGM=CUSTCOB
//OUTDD    DD   DSN=&&DATA,DISP=(NEW,PASS),
//              UNIT=SYSDA,SPACE=(CYL,(20,20))
//COZLOAD  EXEC PROC=COZPROC,ARGS='u@linux'
//PARMS    DD   DSN=HLQ.ORACLE.PARMS,DISP=SHR
//CUSTDATA DD   DSN=&&DATA,DISP=(OLD,DELETE)
//CUSTCTL  DD   DSN=HLQ.CUST.CTL,DISP=SHR
//CUSTLOG  DD   SYSOUT=*
//STDIN    DD   *
```

# Linux on z / zBX

```
sqlldr control=<(fromdsn DD://CUSTCTL),  \
       data=<(fromdsn DD://CUSTDATA),     \
       parfile=<(fromdsn DD://PARMS),      \
       log=>(todsn DD://CUSTLOG)
```

# Data Administration Summary

- Scheduled via nightly batch stream
- sqlldr exit code seamlessly becomes jobstep CC
- Concurrent transfer and loading: *No data at rest!*
  - ✓ *Enabled via process substitution*
- High performance
- Operations can observe real-time job output in the JES spool
- Credentials are restricted by SAF data set controls

# Moving SAS Programs to Linux

- SAS programs have a rich legacy on z/OS
  - ✓ Data analysis
  - ✓ SMF report generation
- Interest in moving processing off platform
  - ✓ Licensing cost consideration
  - ✓ Overall workload reduction
- Several popular tools already exist
- Hybrid Batch processing offers new options
  - ✓ Program source and data can stay on z/OS
  - ✓ Job Step integration of output and return codes

# SAS Language Population Analysis Example

- Performs analysis of Birth/Death population data
- Program source and data reside on z/OS
- Hybrid Batch used to move **execution** to a blade
- For more information see: http://dovetail.com/products/casestudysas.html

# Multipage PDF Generation

- z/OS hybrid batch computing can be used to locate resource hungry jobs to the best architecture
- Java driven PDF generation can be time consuming on the zSeries architecture
- Co:Z can be used to:
  - ✓ Target Java execution to a zBX or zLinux engine
  - ✓ Enable z/OS operations to retain control of scheduling
  - ✓ Keep all data-at-rest on z/OS
- For more information see: http://dovetail.com/products/casestudyitext.html

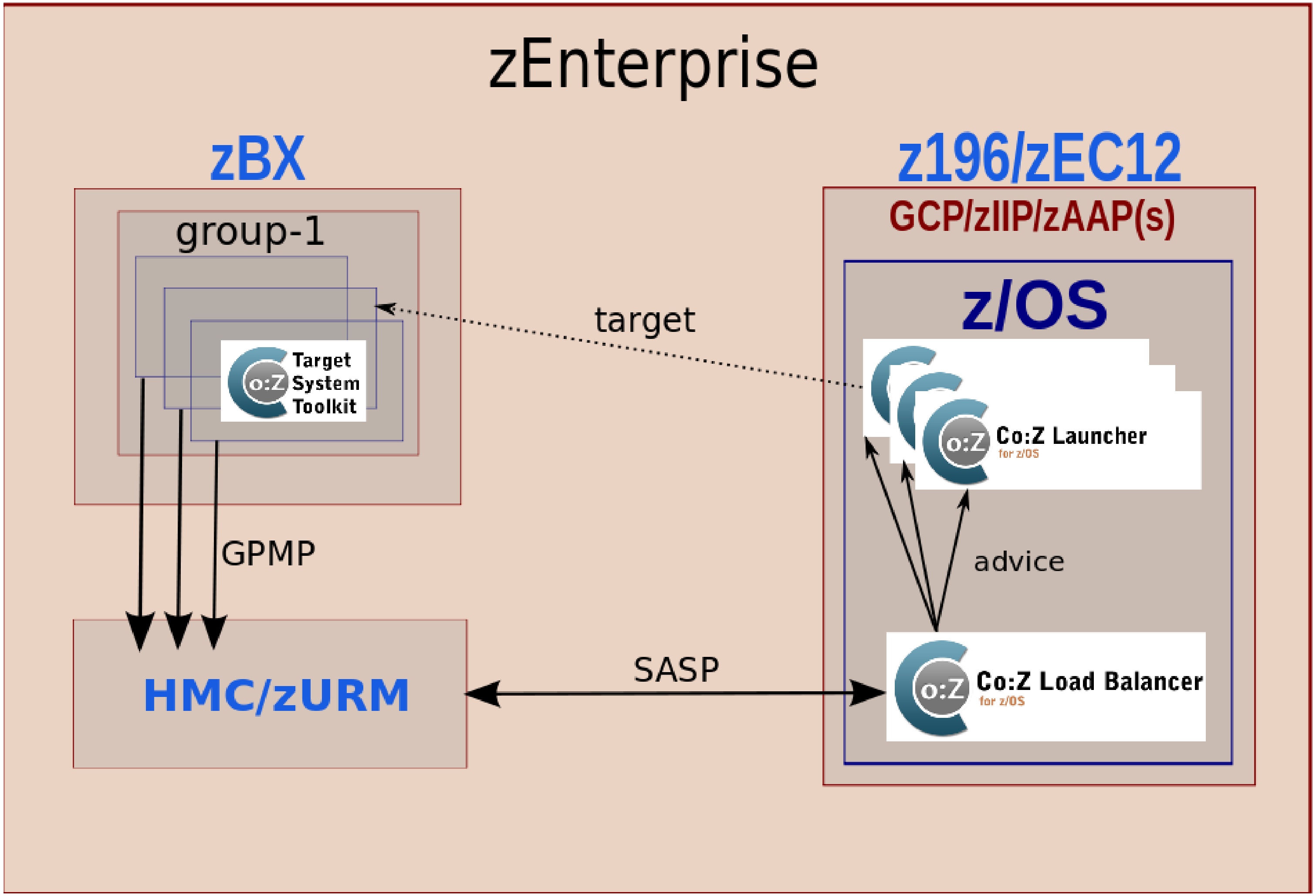
# PDF Success Story

- Generate PDFs from XML dataset in z/OS batch
  - ✓ Java application using open source iText framework
  - ✓ 59 docs/minute on existing 2094-405 workload
- Added (1) zIIP (zAAP mode)
  - ✓ 80 docs/minute
- Hybrid batch with Co:Z Toolkit
  - ✓ Targeted Linux on IFLs
  - ✓ Exploited ssh-tunnel=false and HiperSockets
  - ✓ Simple JCL change; no program changes required
  - ✓ >900 docs/minute
- Surprising new z/OS operator training required:
  - ✓ Don’t cancel job if it doesn’t use CPU time!

# Load Balancing

- As hybrid batch computing is adopted on a large scale, pools of target virtual servers become necessary
  - ✓ Some hybrid batch jobs are short, others require more time
  - ✓ Different servers have different capabilities
- Ideally, new hybrid batch work should be dispatched to the most suitable virtual server
- Need real-time performance load feedback
- zManager to the rescue!

# Co:Z Load Balancer on zEnterprise



# Co:Z Load Balancer Features

- Implements Server/Application State Protocol (SASP)
  - ✓ Interfaces with IBM zEnterprise Unified Resource Manager (zManager)
- Distributes hybrid batch work to zBX virtual servers
- Integrates with console when deployed on z/OS
  - ✓ WTO logging
  - ✓ Console commands for restart and shutdown
- Web service for status, advice, restart shutdown
- For more information see: http://dovetail.com/products/loadbalancer.html

# Summary

- zEnterprise / zBX
  - ✓ Provides hybrid computing environment
- Co:Z Launcher and Target System Toolkit
  - ✓ Provides framework for hybrid *batch* processing
- Co:Z Load Balancer and zManager
  - ✓ Provides load balancing capabilities for hybrid batch processing workloads

# For More Information

- Visit our website: http://dovetail.com
  - ✓ Hybrid Batch Information: http://dovetail.com/solutions.html
  - ✓ View Pre-recorded webinars: http://dovetail.com/webinars.html

- Or, email us at: info@dovetail.com