Informix High Performance Loader
"You got it! Use it!"

Robert Beal – Senior I/T Specialist
IBM Information Management

Session Code – C12
Tuesday, April 29, 2008 • 4:40 p.m. – 5:40 p.m.

2008 IIUG Informix Conference

www.iiug.org

Robert Beal
Certified Senior I/T Specialist
IBM Information Management
rbeal@us.ibm.com

Agenda

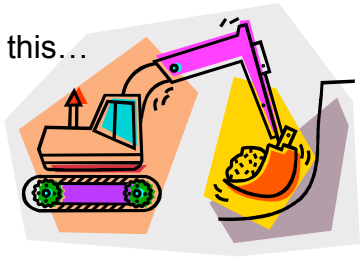
- HPL – What is it?
- When to use HPL
- HPL Details
- Demo
- Example
- Timings
- Conclusions and Questions

HPL – What is it ?

- High Performance Loader
  - Loads (and unloads) into/from Informix tables
  - Its Fast!
    - Multi-threaded
    - Bulk load/unload operations
  - Its Easy!
    - GUI available
    - Command line options
  - Its Free!
    - Comes with IDS

HPL helps you turn this…

Into this…



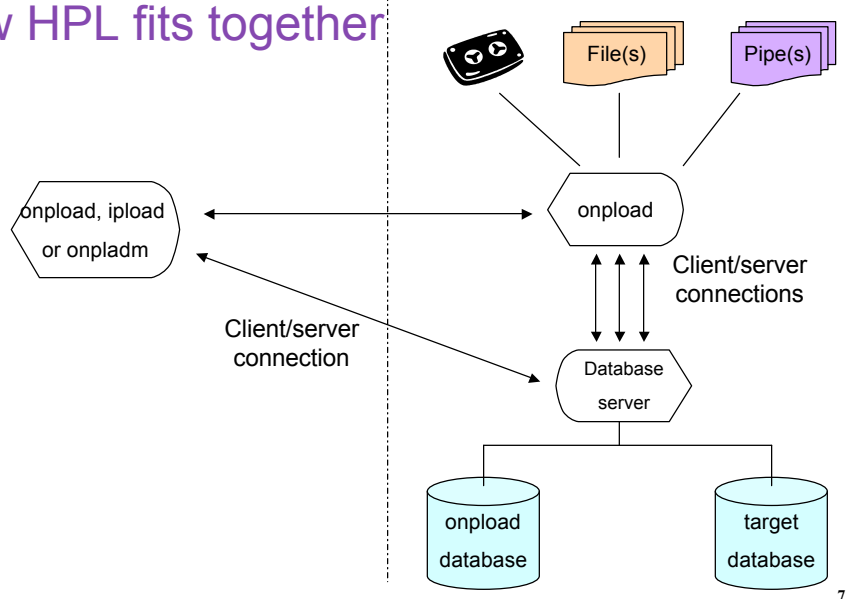
When to Use HPL

- Data to load or unload
  - Use instead of LOAD or UNLOAD SQL Statement
    - Loads always more efficient
    - Unloads make more sense with larger tables
- Need to change format of unload
  - UNLOAD statement inflexible
- Need to change mapping of input records to table columns

HPL Components

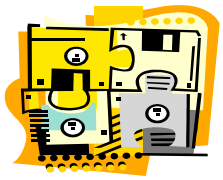
- Command Line Interface
  - onpload
  - onpladm
- onpload process
- GUI tool (ipload)
- Onpload database
  - Stores info about load jobs
- Config file
  - $INFORMIXDIR/etc/plconfig.std

How HPL fits together

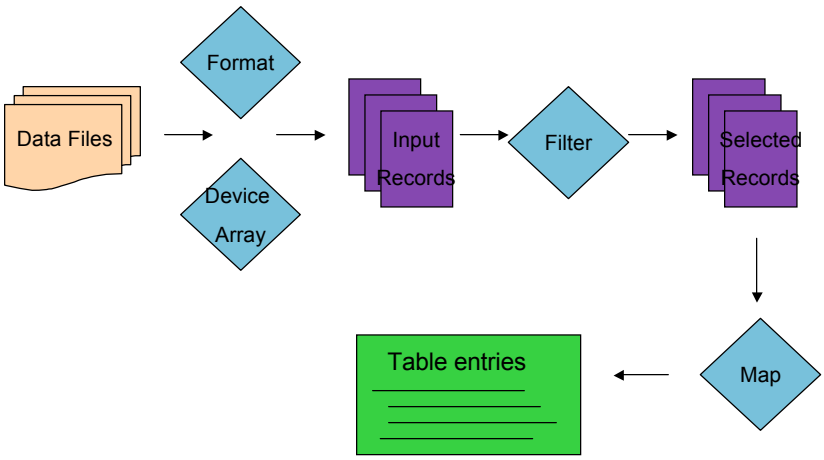


ONPLOAD logical pieces

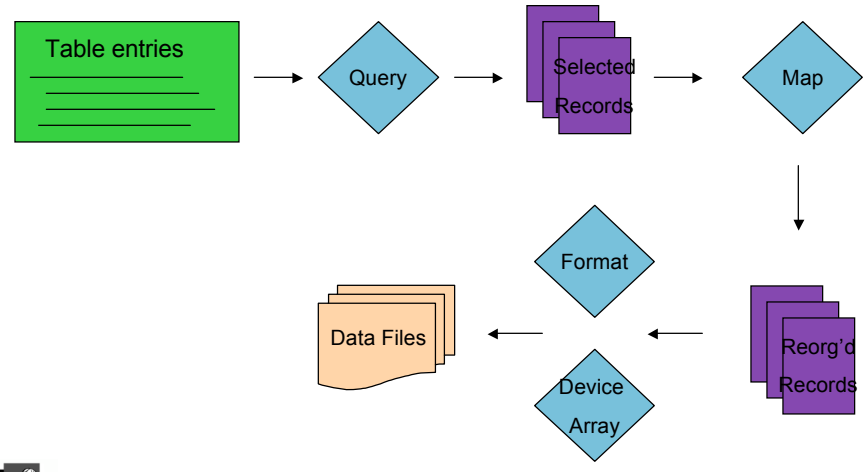
- Project
- Job
  - Load or Unload
  - Mode (Deluxe or Express)
  - Non-data outputs (log, rejects, filtered records)
- Map (columns to device format items)
  - Table/columns and target db
- Format (names parts of record in I/O device)
- Device (file(s), pipe(s) or tape(s))
- Filter (loads)
- Query (unloads)



How Loads fit together



How Unloads fit together

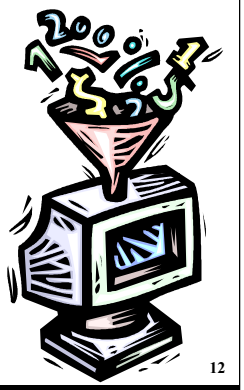


Loading Modes

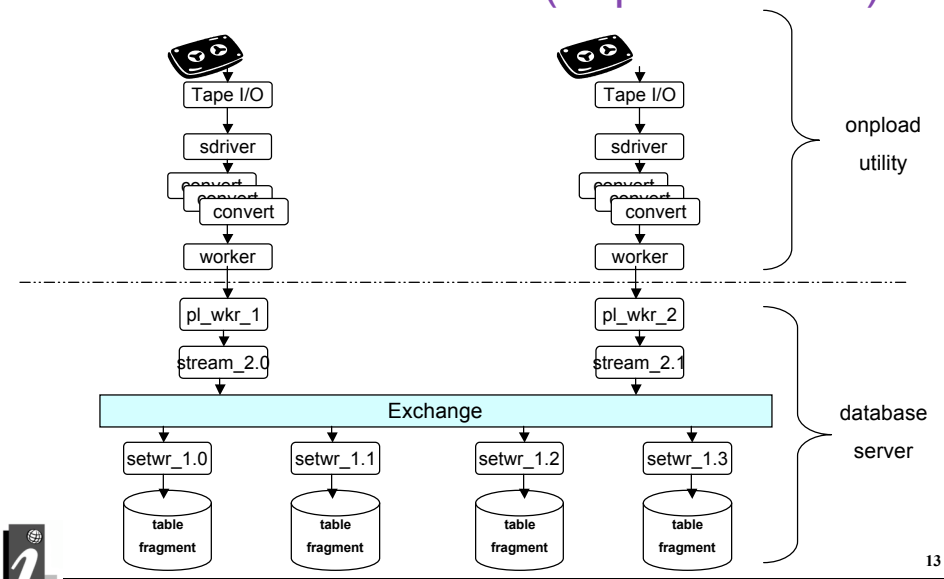
- **Deluxe Mode**
  - Table is available for updates
  - Indexes maintained during load
  - Constraints evaluated & Triggers fire
  - Slower
- **Express Mode**
  - Faster (bypasses bufferpool and logging)
  - Disables indexes, constraints & triggers
    - Re-enables when load is done
    - Triggers do not fire on loaded data
  - Table not available during load process
    - Must do a Level-0 backup to make table available

Input / Output Formats

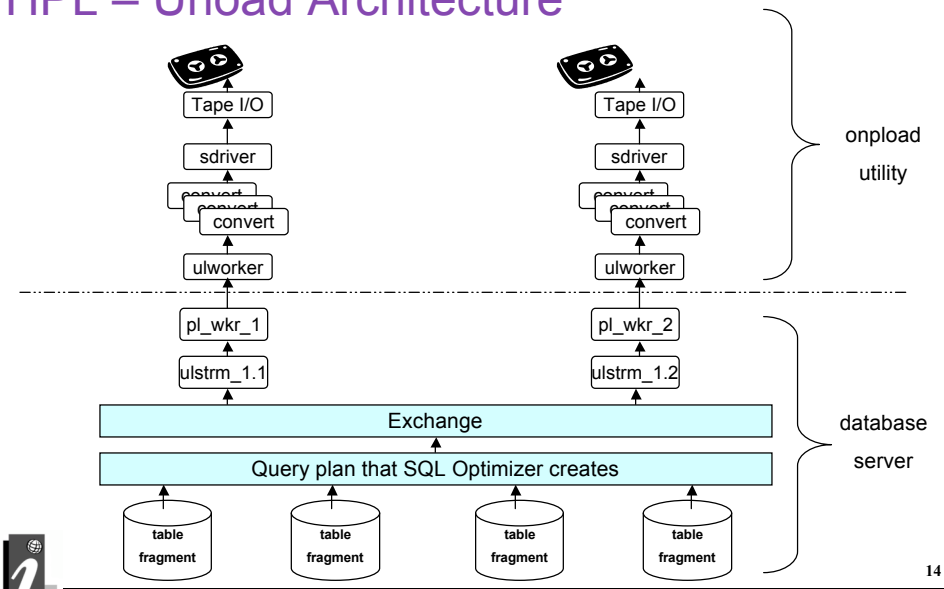
- Fixed length
  - based upon table definition
- Delimited (Default, pipe delimited)
  - Can choose delimiter character
  - Can choose field start / end characters
- COBOL
  - Uses cobol-like syntax to describe fields



HPL – Load Architecture (Express Mode)



HPL – Unoad Architecture



Onpload threads

- Separate from database threads
  - Do not run in database CPU VPs, but in onpload
- Tape I/O threads
  - Handle I/O to tapes and pipes
  - AIO threads used for files
- Worker threads
  - Communicate with Database Server
  - One per input device
- Conversion threads
  - Do things like case conversion, code-set, etc.
  - One per device
- Sdriver threads
  - Handle device abstraction from I/O threads
  - Pass records to conversion threads

Database Server Threads

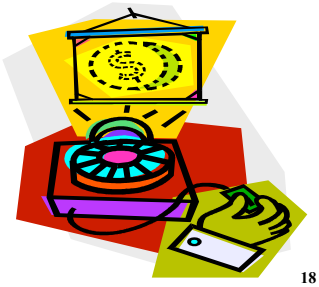
- pl_wkr thread
  - Handles exchange of data from onpload worker thread
- cadiload thread (deluxe mode)
  - Standard insert thread, uses buffer cache
- stream threads (express mode)
  - Hand off data from pl_wkr threads
- setwr threads
  - append data directly to extent

Ways to use HPL

- GUI (ipload)
  - Can automatically build jobs for you
  - Can invoke onpload from here
- onpload
  - CLI for running jobs from command line
- onpladm
  - Can create projects, jobs, etc

ipload demo

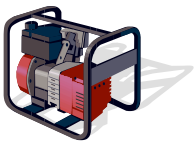
- VMWare Linux image used
- Invoking ipload
- How to generate a job
  - Ease of use
  - What you end up with
- How to modify a job



onpladm utility

- Command line way to create projects, jobs and elements
- Command elements:
  - Create (project, job, map, object)
  - Modify object
  - List (project, job, map, format, query, filter, device, machine and defaults)
  - Describe (project, job, map, format, query, filter, device and machine)
  - Delete (project, job, map, format, query, filter, device and machine)
  - Run (job or project)
  - Configure defaults

onpladm



- Quick way to generate a Unload/Load job
  - Create a project
  - Create a job – generates the rest
    - Defaults use job name
- How to update/enhance
  - onpladm modify object –F <spec file>
  - Specification file described in manual
    - Hint: use onpladm describe <object> to an output file then modify it.
- Link to HPL Manual (IDS 11.10)
  - http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=G229-6377-00

Example

- Want to create a job to load a table customer in tpch db.
  - Want 4 files for input
- Create a project

  onpladm create project foobar
- Create a load job

  onpladm create job ld_cust –p foobar -d <filename>
  –D tpch –t customer –fl –zD
- Just created a load job with one file
  - Want to modify job to use 4 files

Example (continued)

- Use onpladm ‘describe job’ to see what you have

onpladm describe job ld_cust –p foobar –fl

- Add the ‘-R’ switch to see all objects for the job

```
BEGIN  OBJECT  LOADJOB    ld_cust

PROJECT              foobar
DEVICE             ld_cust
MAP                ld_cust
FILTER
SERVER             ids10_shm
DATABASE             tpch
FLTFILE           /tmp/ld_cust.flt
REJECTFILE          /tmp/ld_cust.rej
LOGFILE           /tmp/ld_cust.log
RUNMODE              E
GENERATEVIORECS      Y
TAPES              0
NUMRECORDS           0
STARTRECORD          0
MAXERRORS            0

END  OBJECT
```

Example (continued)

- Use onpladm 'describe device' to see what device looks like

onpladm describe device ld_cust

```
BEGIN  OBJECT  DEVICEARRAY    ld_cust
BEGIN  SEQUENCE
TYPE                  FILE
FILE
      /home/informix/pload/unload/customer.dat
TAPEBLOCKSIZE          0
TAPEDEVICESIZE         0
PIPECOMMAND
END  SEQUENCE

END  OBJECT
```

- Save output in a file
  - ld_cust_dev.job

Example (continued)

- Modify device file to have four files instead of one.
- Use onpladm to modify device description

onpladm modify object –F ld_cust_dev.job

```
BEGIN  OBJECT  DEVICEARRAY    ld_cust
BEGIN  SEQUENCE
TYPE                 FILE
FILE
      /home/informix/pload/unload/customer_1.dat
TAPEBLOCKSIZE          0
TAPEDEVICESIZE         0
PIPECOMMAND
END  SEQUENCE
            .
            .
            .
BEGIN  SEQUENCE
TYPE                 FILE
FILE
      /home/informix/pload/unload/customer_4.dat
TAPEBLOCKSIZE          0
TAPEDEVICESIZE         0
PIPECOMMAND
END  SEQUENCE

END  OBJECT
```

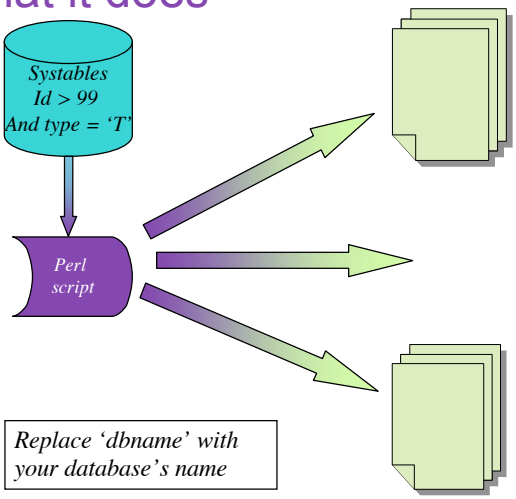
Practical Considerations

- Other delimiters
  - IDS default is '|', can update *formats* table to change
  - Can also use double-quotes to surround strings
- Non-IDS database targets or sources
  - IDS will put a backslash '\' in front of delimiter characters contained in strings
    - Unless string is quoted
  - Make sure date formats match
    - If not, loads take forever w/no data inserted

Free Stuff !!!

- gen_hpl_4_db.pl script to generate load and unload HPL jobs for a database
- What it does:
  - Generates a shell script to create load and unload jobs for each table in a database
  - Generates two other shell scripts to run the jobs: one for load, one for unload
- What it does NOT do:
  - Does not run the shell scripts
  - Does not split output into multiple files
  - Does not make any guarantees
- ***You get what you pay for!***

What it does



cr_dbname_hpl_jobs.sh: *creates load and unload jobs. Run this first.*

load_dbname_hpl.sh: *Runs the load jobs*

unload_dbname_hpl.sh: *Runs the unload jobs*

Replace 'dbname' with your database's name

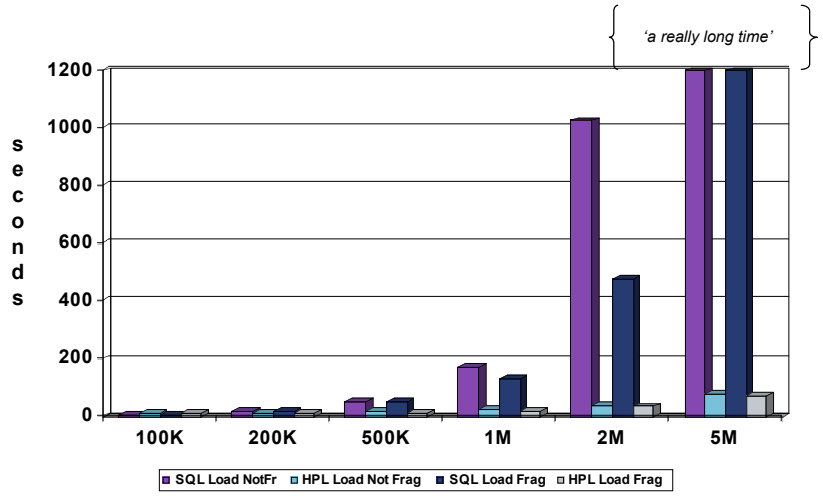
Extra Credit

- Can input / output to pipes or tapes
- Can use external programs to filter or convert input/output
- Use plconfig.std to change buffer sizes, threads per device
- Can use one onpload database for multiple target databases on multiple machines

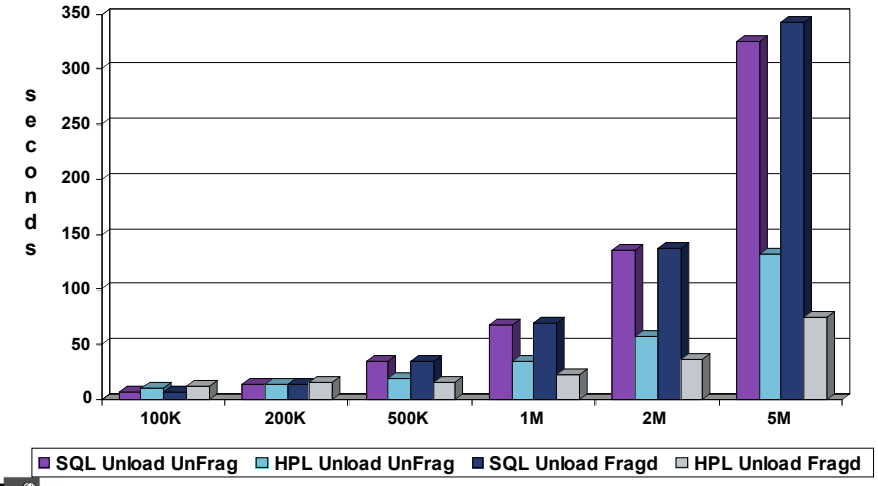
Timing comparisons

- Ran tests using Informix 10.0.FC4
  - pSeries (p570) 4 CPUs, 4G ram
  - Loaded / Unloaded tpch 'lineitem' table. 146 bytes/row
    - Two versions, one fragmented (4) and one un-fragmented
    - No indexes, constraints, etc.
  - Six tests, 100K, 200K, 500K, 1M, 2M and 5M record loads
- Comparisons with
  - SQL Load vs HPL Load
  - SQL Unload vs HPL Unload
  - HPL Load/Unload 4 files vs 1 file
- Not a lot of effort put into tuning

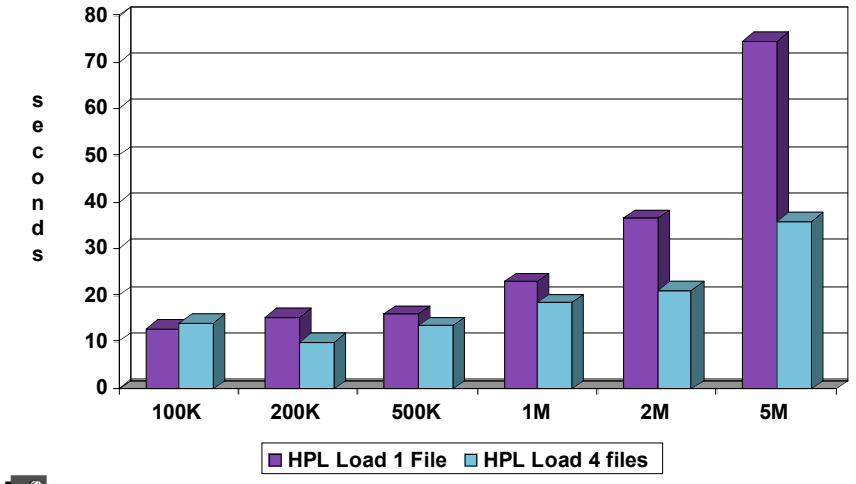
SQL vs HPL Loads, Fragmented and Not



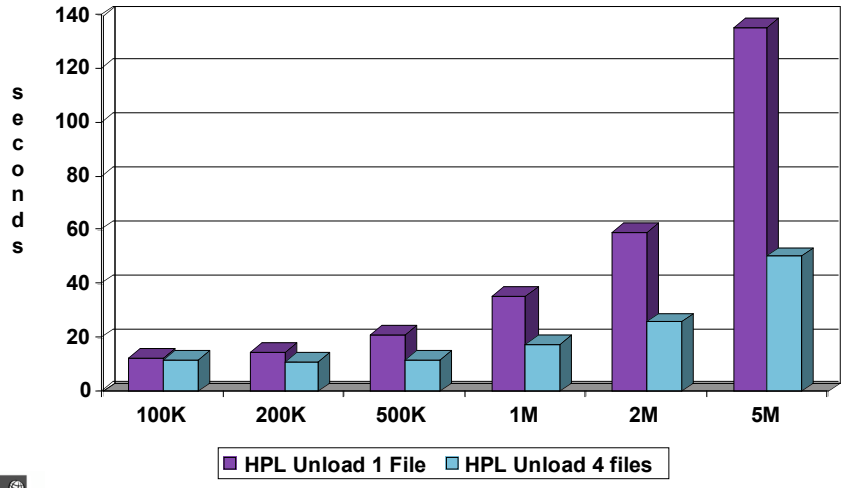
SQL vs HPL Unloads, Fragmented and Not



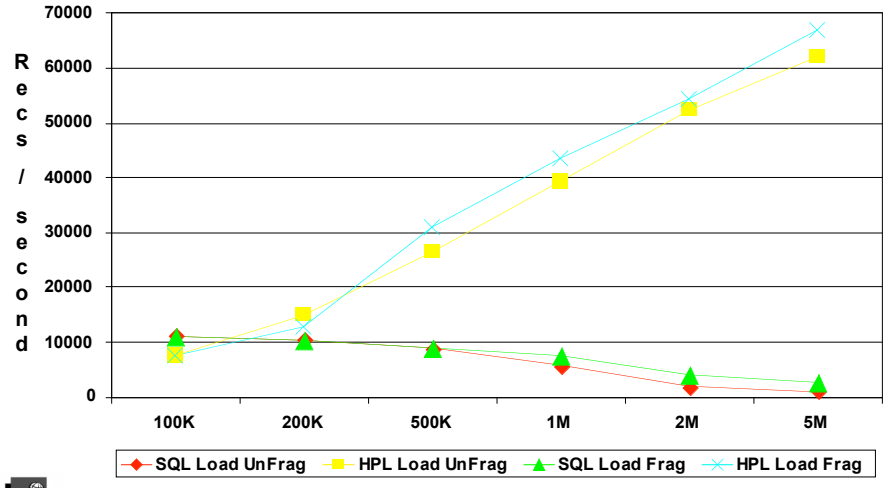
HPL Load Fragmented table, 1 vs 4 files



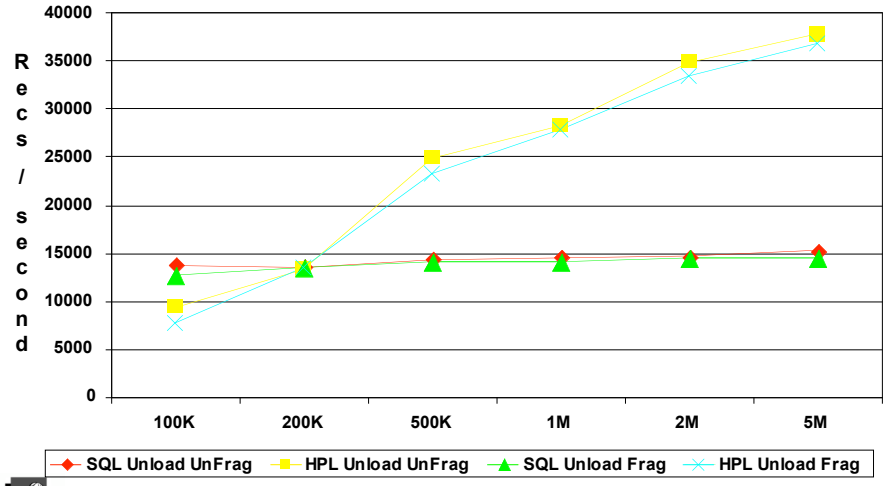
HPL Unload Fragmented table, 1 vs 4 files



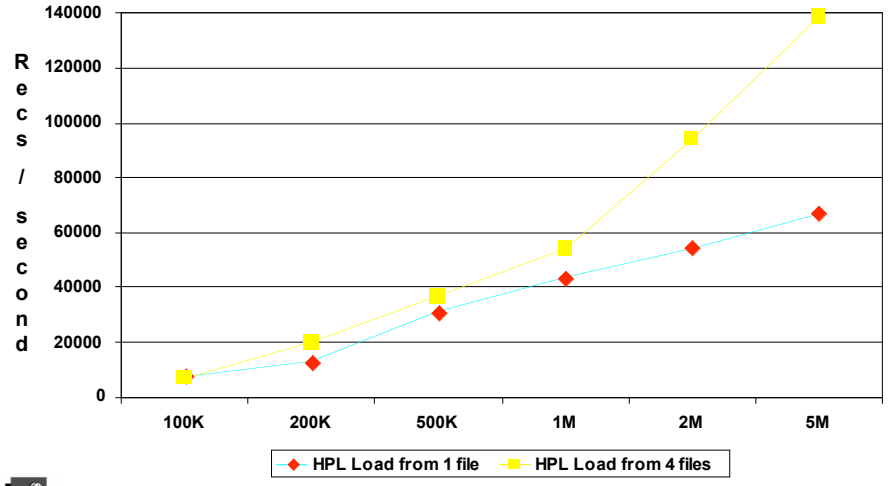
SQL vs HPL Loads, Records per Second



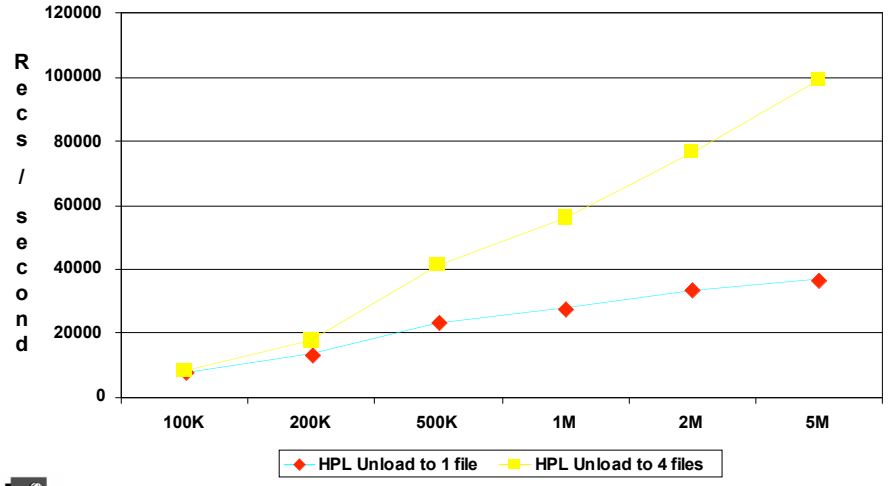
SQL vs HPL Unloads, Records per Second



HPL vs HPL Loads, Records per Second



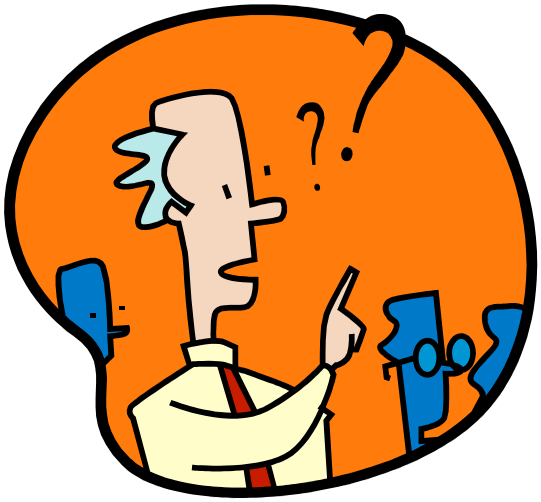
HPL vs HPL Unloads, Records per Second



Conclusions

- HPL and SQL do better with Fragmented tables
  - (duh!)
- HPL benefits start kicking in at around 200K records
- HPL not quite as simple as LOAD/UNLOAD SQL
  - Not that much harder either…..
  - But really worth it when volumes are high and time is scarce

Questions ??




Vielen Dank
Obrigado!
Thanks
Gracias
Merci