Optimizing Linux for IDS

Alexey Sonkin

CIDC

Session C18
Wednesday April 30, 2008 4:40 – 5:40 PM
Platform: The Informix Edge

2008 IIUG Informix Conference

www.iiug.org

Abstract

- Shared Memory configuration on Linux for 32-bit and 64-bit platforms
- Virtual memory implementation in Linux kernel
- Linux kernel optimizations for Informix (virtual memory, network buffers, etc.)
- RAW devices vs. block devices vs. flat files on Linux
- KAIO vs. Informix AIO on Linux;
- Linux specific problems with Informix (OOM killer, GLIBC compatibility, etc)

What Linux to choose?

- IDS is officially supported on Suse, RedHat, Asianux, Debian, Ubuntu
- IDS can't start in most Linux distributions, because of GLIBC incompatibility
- Suse and RedHat 'Enterprise' kernels are significantly better tested, that standard kernels
- Suse and RedHat are supported by majority of hardware vendors (SUN, IBM, HP, EMC, etc)
- Suse and RedHat provide technical support for Linux
- Some parameters (e.g. default I/O scheduler) are better optimized for database usage

How to allocate memory to IDS

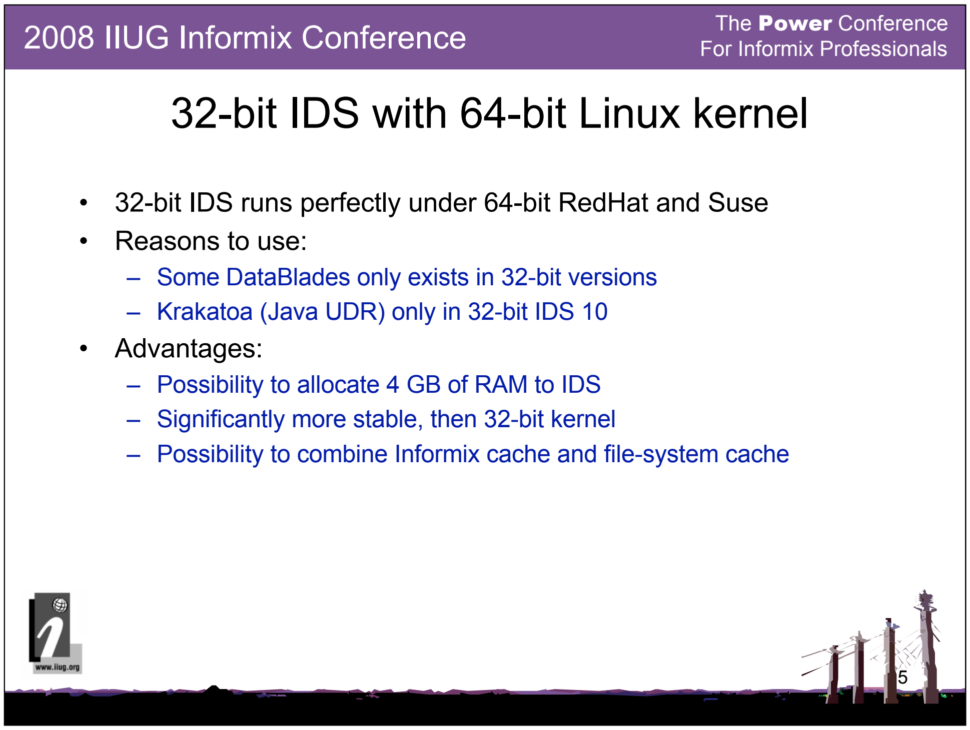
- 32-bit – kernel can only address 3 GB of virtual memory for a single process
- 64-bit kernel allows to address 4 GB for 32-bit process
- IDS requires contiguous space for shared memory segments; most Linux distribution tend to map shared libraries starting at 1 GB (0x40000000) address:

```
– ifmx@larcius:~$ cat /proc/$$/maps
– 08048000-080b5000 r-xp 00000000 08:01 16069  /bin/bash
– ………
– 40000000-40013000 r-xp 00000000 08:01 8052   /lib/ld-2.1.3.so
– 40013000-40014000 rw-p 00012000 08:01 8052   /lib/ld-2.1.3.so
– ….......
```

This decreases the virtual address space by another 1GB; fixed in RedHat, configurable in Suse:

```
– ifmx@db1:$ echo 33554432 > /proc/$$/mapped_base
– ifmx@db1:$ oninit -v
– ifmx@db1:$ cat /proc/8545/maps
– 02000000-02016000 r-xp 00000000 08:01 96489  /lib/ld-2.3.3.so
– 02016000-02017000 rwxp 00016000 08:01 96489  /lib/ld-2.3.3.so
– …….
```

(only applicable to 32-bit kernel)



If you need to run 32-bit IDS for some reason on Linux, consider running it under 64-bit kernel

Kernel shared memory parameters

- To display current kernel memory settings:
  - ifmx@dw-1g:~$ cat /proc/sys/kernel/shmmax
  - 34359738368
- Kernel dynamic parameter configuration file:
  - ifmx@dw-1g:~$ cat /etc/sysctl.conf
  - ………
  - kernel.shmmax = 34359738368 (64-bit)
  - kernel.shmall = 8388608 (64-bit Linux) <- number of 4k pages for IDS
  - vm.min_free_kbytes = 128000
  - vm.swappiness = 1
- To re-apply new configuration file at runtime:
  - dw-1g:~ # sysctl -e -p /etc/sysctl.conf
- Another way to dynamically change kernel settings:
  - dw-1g:~ # echo 34359738368 > /proc/sys/kernel/shmmax

This example shows parameters, used on machine with 32 GB of RAM

File system vs. Raw dev vs. Block dev

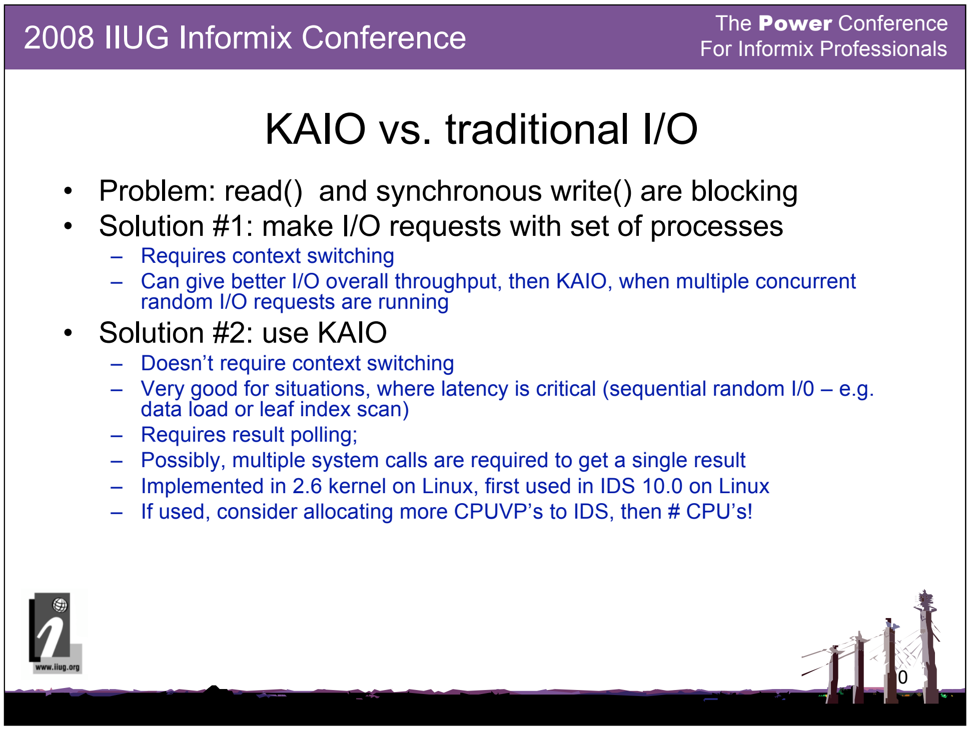
- O_SYNC flag for open() system call makes all writes synchronous
  - IDS always uses this flag, that effectively makes file system and block devices as secure, as RAW devices!
- With Files system and block devices, all I/O reads and writes go through file system cache
- With RAW devices, no extra copying to file system cache is made for both reads and writes
- For file system, Linux cache buffer size is usually 4096 bytes;
- For block device, buffer size is 512 bytes
- For each buffer, Linux (64-bit) allocates 96-byte buffer_head structure
  - Huge waste of memory for block devices

Direct I/O on Linux

- O_DIRECT flag was introduced in Linux 2.6
  - It makes all I/O operations go directly to applications buffers, without extra copying to OS cache (similar to RAW devices)
  - I/O size must be multiple of 512 bytes
  - Doesn't provide extra security vs. O_SYNC, provides memory optimizations
  - On Linux, works for both Block devices and files! (Block devices effectively turn into RAW devices, if opened with O_DIRECT)
  - Informix 10 uses this flag only for block devices, not files
  - Informix 10 only uses DIRECT with KAIO.
    (with KAIOOFF=1 block devices turn into REAL block devices!)
  - Informix 11.10 (Bug!!!) with block devices only uses O_DIRECT with AIOVP's
    (without KAIOOFF block devices turn into REAL block devices!)
  - IDS 11.10 can use O_DIRECT with files (both KAIO and AIOVP's)
  - IDS 11.10 doesn't use O_DIRECT for TEMP dbspaces

What file system to use for IDS chunks?

- RaiserFS (never use for IDS chunks!!!)
  - Combination of database and File system;
  - Optimized for storage of huge number of small files;
- EXT2: unjournaled file system
  - Very good database performance
  - Potentially very long FSCK at Linux restart
- EXT3: data+metadata journaling (‘data=journal’ mount option)
  - Improves reliability at a performance penalty
  - Default mount mode for EXT3
- EXT3: metadata-only journaling (‘data=ordered’ mount option)
  - Almost as fast, as EXT2 for normal DB operations
  - Significant acceleration of FSCK
  - Extra data security over EXT2
- VxFS - Veritas file system (commercial) – good for read-mostly DB’s
  - Page journaling (data+metadata) – inefficient for write-intensive DB’s
  - Optionally, provides ‘raw’ (uncached) mounting



Most artificial benchmarks benefit from imroved latency, while majority of real-life systems benefit much more from overall I/O throughput.

My measurements of overall loads under highly parallel I/O activity demonstrated much better results with Informix I/O, then with KAIO

2008 IIUG Informix Conference

The **Power** Conference For Informix Professionals

Partitioning on Linux

- root@myhost# **fdisk /dev/sdf**
- **Command action**
-    e   extended
-    p   primary partition (1-4)
- **p**
- Partition number (1-4): **1**
- First cylinder (1-54660, default 1): **1**
- Last cylinder or +size or +sizeM or +sizeK (1-54660, default 54660): **54660**
- **Command** (m for help): **p**
- Disk /dev/sdf: 449.5 GB, 449596882944 bytes
- 255 heads, 63 sectors/track, 54660 cylinders
- Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdf1		1	54660	439056418+	83	Linux

www.iiug.org

1

Linux partitioning in default mode…

2008 IIUG Informix Conference

The **Power** Conference For Informix Professionals

Advanced Partitioning on Linux

- **Command** (m for help): **x**
- **Expert command** (m for help): **p**
- Disk /dev/sdf: 255 heads, 63 sectors, 54660 cylinders

Nr	AF	Hd	Sec	Cyl	Hd	Sec	Cyl	Start	Size	ID
1	00	1	1	0	254	63	1023	63	878112837	83
2	00	0	0	0	0	0	0	0	0	00
3	00	0	0	0	0	0	0	0	0	00
4	00	0	0	0	0	0	0	0	0	00

- **Expert command** (m for help): **b**
- Partition number (1-4): 1
- New beginning of data (63-878112899, default 63): **256**
- **Expert command** (m for help): **p**
- Disk /dev/sdf: 255 heads, 63 sectors, 54660 cylinders

Nr	AF	Hd	Sec	Cyl	Hd	Sec	Cyl	Start	Size	ID
1	00	1	1	0	254	63	1023	256	878112644	83
2	00	0	0	0	0	0	0	0	0	00
3	00	0	0	0	0	0	0	0	0	00
4	00	0	0	0	0	0	0	0	0	00

www.iiug.org

2

This example shows, how to align Linux partitions for use with external SCSI or Fiber Channel arrays

Virtual vs. physical drive geometry

- Modern hard drives report virtual, not physical CHS (cylinder-head-sector) geometry to OS and BIOS: 63 512-byte blocks per track
- BIOS and OS is historically using this information for partition table optimization: first partition starts on Virtual Cylinder #1
- All modern hard drives physically have Multi-zone geometry:
  - All physical cylinder are grouped into zones (15 zones typical)
  - Each zone has different number of sectors per track
- Smart arrays also report misleading geometry
  - Use of virtual drive geometry causes misalignment between IDS I/O units (2k pages) and internal array I/O units (4k or 8k pages)

www.iiug.org

Creating raw devices on Linux

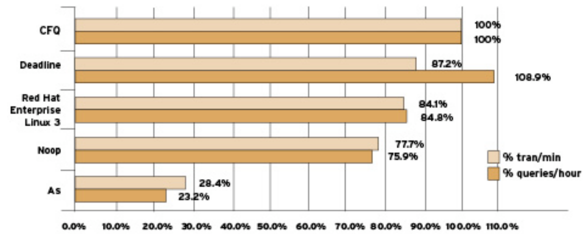
- RAW devices were introduces in kernel 2.4
- By default, Linux doesn't create RAW devices for disk partitions
- To create a RAW devices, one should use 'RAW' utility:
  - root@myhost# **raw /dev/raw/raw1 /dev/sda1**
  - `/dev/raw/raw1:  bound to major 8, minor 1`
  - root@myhost# **raw -qa**
  - `/dev/raw/raw1:  bound to major 8, minor 1`
  - root@myhost# **ls -l /dev/sda1**
  - `brw-rw----  1 root disk 8, 1 Jan 30 18:29 /dev/sda1`
  - root@myhost# **chown informix:informix /dev/raw/raw1**
  - root@myhost# **chmod 660 /dev/raw/raw1**
  - root@myhost# **ln –s /dev/raw/raw1 /data/informix_data/chunk1**
- After restart, Linux looses RAW devices (need redo in boot script)

I/O schedulers on Linux

- **Noop Scheduler**
  - This scheduler only implements request merging.
  - Good for use with smart arrays
- **Anticipatory IO Scheduler ("as scheduler")**
  - default scheduler in older 2.6 kernels
  - Optimimized for sequential I/O – never use for database!
- **Deadline Scheduler**
  - preferred scheduler for database systems
- **Complete Fair Queueing Scheduler ("cfq scheduler")**
  - Default for Suse, RedHat, and for generic 2.6.18+

I/O schedulers on Linux (cont.)

- **To display active scheduler:**
  - root@myhost# dmesg | grep scheduler
  - `Using cfq io scheduler`
- **To configure scheduler at boot time:**
  - **'Elevator=cfq' kernel boot parameter**
- **Effect of I/O scheduler on DB performance**
  - **(from http://www.redhat.com/magazine/008jun05/features/schedulers)**



Memory management on Linux

- Linux kernel implements ‘on-demand’ physical memory allocation for user-space applications (when memory page is physically accessed)
- Linux kernel internally operates in physical memory
- Linux kernel allocates memory to application, disk cache buffers, internal kernel structures from FREE pool
- **KSWAPD** is a kernel thread (kernel-mode-only process), responsible for keeping FREE pool at constant level
  - Releases ‘clean’ buffer pool pages, not used for a long time
  - Swaps out application pages, not used for a long time
  - Balance between these methods controlled by **/proc/sys/vm/swappiness: Range (0-100), default ’60’, minimal swapping – ‘0’**
- Size of FREE zone is controlled by **/proc/sys/vm/min_free_kbytes:**
  - Default value is extremely low for machines with big memory (>1GB);
  - Low setting causes swapping, kernel errors and triggers OOM killer

NUMA-specific memory considerations

- On NUMA machines (e.g. multiprocessor Opteron machines), different CPU's have different access latencies to different memory banks;
- DB performance significantly degrades, if most actively accessed pages are allocated in a single bank;
- Ways to avoid the problem:
  - BIOS-level 'memory inter-node interleaving'
  - OS-level memory allocation optimization (Implemented in RedHat 4, Suse 9,10, Solaris 10)

2008 IIUG Informix Conference

The **Power** Conference For Informix Professionals

Linux kernel OOM killer

- **Fragment from /usr/src/linux/mm/oom_kill.c**

```
– /** oom_kill - kill the "best" process when we run out of memory
–   * If we run out of memory, we have the choice between either
–   * killing a random task (bad), letting the system crash (worse)
–   * OR try to be smart about which process to kill. Note that we
–   * don't have to be perfect here, we just have to be good.   */
–  static void oom_kill(void)   {
–  ........
–          p = select_bad_process();
–           /* Found nothing?!?! Either we hang forever, or we panic. */
–          if (p == NULL)
–                  panic("Out of memory and no killable processes...\n");
–           /* kill all processes that share the ->mm (i.e. all threads) */
–          for_each_task(q) {
–                  if (q->mm == p->mm)
–                          oom_kill_task(q);         }
–  ........  }
```

www.iiug.org

This is a self-explaining fragment from the Linux kernel source code

2008 IIUG Informix Conference

The **Power** Conference For Informix Professionals

Linux kernel OOM killer (cont.)

- **from ONLINE.LOG:**
  - **09:41:19 Fatal error in ADM VP at mt.c:11462**
  - **09:41:20 Unexpected virtual processor termination, pid = 11130, exit = 0x9**
  - **09:41:22 Logical Log 12976 - Backup Completed**
  - **09:41:23 invoke_alarm(): /bin/sh -c '/home/informix/etc/log_full.sh 5 6 "Internal Subsystem failure: 'MT'" "Unexpected**
  - **virtual processor termination, pid = 11130, exit = 0x9" '**
  - **09:41:23 invoke_alarm(): mt_exec failed, status 256, errno 0**
  - **09:41:23 PANIC: Attempting to bring system down**
- **From dmesg output:**
  - **Out of Memory: Killed process 11130 (oninit).**

Linux kernel decided to kill Informix ADM VP process, because Linux needed to allocate memory for file system caching…

SOC VP kernel-mode usage

- Single process, multiple connections problem:
  - Single process is listening to 10 connected sockets;
  - Each socket receives request 1 time per second
  - Each second, the process spends N milliseconds in kernel mode, polling TCP requests
- The number of clients changes from 10 to 100, same request rate from each client;
- How much time the process should spend in kernel mode, polling 100 connections?

2008 IIUG Informix Conference

The **Power** Conference
For Informix Professionals

SOC VP kernel-mode usage (cont.)

- Answer: 100 times more!
- Traditional SELECT system call has quadratic dependency on the number of connected sessions:
  - SELECT call should accept the entire array of open sockets;
  - For each socket, kernel needs to make expensive verification;
  - SELECT should be called each time, the packet arrives to any socket;
  - SELECT frequency is proportional to number of open sockets!

www.iiug.org

2

It is not uncommon for poll threads to be 40% of CPU resources, consumed by Informix is very busy OLTP systems…

Configuring #Net VP's

- **Sulution #1: distribute active sockets across multiple processes (Informix Net VP's)**
- Never run TCP poll threads on CPUVP on heavily loaded OLTP systems
- How many NET VP's to configure? Standard approach:
  - Allocate 1 NET VP for each 100-200 connections
  - Doesn't work well in non-standard environment (e.g. Internet connection multiplexing)
- Alternative approach:
  - allocate as many NET VP's as necessary until NET VP CPU consumption drops below 10% of total CPU consumption

Informix MaxConnect

- **Solution #2: Connection Multiplexing**
- Informix MaxConnect reduces the number of connections by multiplexing multiple logical database client connection over a single physical network connection
- Should be located on a separate computer: upon first connection, consumes 100% CPU resources of the computer, running MaxConnect
- Has significant implications on multithreaded applications
- Not officially supported with IDS 10

www.iiug.org

Epoll() on Linux & FASTPOLL in IDS 10

- **Solution #3: Use EPOLL() instead of SELECT()**
- Epoll() is edge-triggered poll replacement for the 2.6 Linux:
  - Array of sockets is registered in kernel for event notifications;
  - Instead of socket list, array identifier is passed to the kernel
- Dramatically reduces kernel-time, required to poll connections;
- Implemented in IDS 10.0XC5 on several platforms: Linux, Solaris
- Requires ONCONFIG parameter FASTPOLL=1
- Initial bug in 10.0FC5: FASTPOLL should not be used on HDR Secondary (fixed in 10.0FC6)

Linux kernel network optimization

- Default Linux kernel network parameters are optimized for typical Client-Server environment:
  - Data blocks are relatively small (<32k);
  - Client doesn’t send extra packet until server responds;
  - Server doesn’t send package without request
- Streaming applications (Informix HDR):
  - Data blocks are relatively big;
  - Server can send extra block into a stream without delivery confirmation of the previous block

2008 IIUG Informix Conference

The **Power** Conference
For Informix Professionals

Network kernel parameters for HDR

- To display current kernel network settings:
  - ifmx@dw-1g:/etc$ cat /proc/sys/net/core/rmem_default
  - 2621440
- Kernel dynamic parameter configuration file:
  - ifmx@dw-1g:/etc$ cat sysctl.conf
  - ....
  - `net.core.rmem_default = 1048576`
  - `net.core.wmem_default = 1048576`
  - `net.core.rmem_max = 1048576`
  - `net.core.wmem_max = 1048576`
  - `net.ipv4.tcp_wmem = 8192 1048576 1048576`
  - `net.ipv4.tcp_rmem = 16384 1048576 1048576`
  - `net.ipv4.tcp_mem = 1568768 1570816 1572864`
- To re-apply new configuration file at runtime:
  - dw-1g:~ # sysctl -e -p /etc/sysctl.conf
- Another way to dynamically change kernel settings:
  - dw-1g:~ # echo 1572864 > /proc/sys/net/core/rmem_default

www.iiug.org

7

Streaming applications significantly benefit from increased kernel send and receive buffers. Both performance and reliability are improved.

Session: C18

Optimizing Linux for IDS

Alexey Sonkin

Cambridge Interactive Development Corporation

alexeis@cidc.com