A Simple Software Architecture and a Proposed Data Format for e–VLBI

- Metsähovi Radio Observatory (MRO)
- Ari Mujunen; Ari.Mujunen@hut.fi

Contents:
- Software Description
- Performance
- Data Format Discussion
Principles

- VLBI recorder design based on commodity “office” PCs and hard disks
- Simple and inexpensive VSIB data I/O board
  - Just transfers VSI–H data to/from PC main memory
  - Now PCI32/33 burst–mode bus master DMA
    - Can be easily and quickly redesigned
- Data in normal files in Linux
  - Independent of hard disk (etc.) evolution
  - e–VLBI ready with normal networking tools (FTP...)
Software Architecture

- Core recorder written in plain C; three parts:
  - Loadable Linux device driver ‘vsib.o’
  - Disk writing/reading program/daemon ‘vsibcon’
  - VSI–S/Mak5A–style command interface ‘vsibcli’
- Mark 4 wrapper on top of ‘vsibcli’ (JIVE)
- Easier interface to JIVE Mark 4 correlator
- Additionally, could be used:
  - At stations: instant FS compatibility, mixed disk/tape ops
  - With Mark5A to make it look like a Mark 4 tape recorder
Driving the VSIB Board (‘vsib.o’)  

- Uses PLX scatter/gather DMA descriptors and Linux “bigphysarea=” memory for ring buffer
- Ring works in hardware without software assistance
- An economic and easy way to provide increasingly faster and larger disk/network buffer memories

- A normal Linux character mode driver
  - ‘write()’ mapped to “fill the ring buffer”
  - ‘read()’ mapped to “empty the ring buffer”
  - ‘ioctl()’s to start/stop, ‘lseek()’ to adjust playback ptr
‘vsibcon’ <-> ‘vsibcli’

- Both run continuously, communicate using a shared memory segment
- Decouples disk I/O (‘vsibcon’) from networking (‘vsibcli’)
- ‘vsibcon’ basically a small loop issuing ‘read() <-> write()’ calls to ’vsib.o’ driver and disks
- ‘open()’/‘close()’ files and start/stop according to flags in shared memory; uses ntpd network time
Performance Tests

- July 2001 ‘wr.c’ disk-only performance tests suggested maximum throughput <~420Mbits/s
- Using the July 2001 “office” PC architecture, now:
  - April 2002 ‘wrvsib.c’ VSIB board to disk tests suggest >~520Mbits/s sustained performance
  - Attained using a $160 Asus A7N–266 nVidia nForce 420 motherboard
    - 3x IBM 120GXP, 1GB PC–2100 DDR SDRAM, AMD Athlon XP2000+
Performance Issues

- Relatively insensitive to:
  - Block sizes, number of files, amount of Linux buffer cache RAM, etc.

- Multiple disks managed with standard Linux software ‘raid0.o’ driver
  - Initial tests (‘wr.c’) showed that using Raid0 incurs no additional overhead vs. own disk management
  
- “Last 10%”? Raid0, ext2, UDMA133,...?
The PCI Story of PC Chipsets

- VIA KT133, KT266, KT266A
- AMD 762MPX dual CPU PCI64/66 <$250
- nVidia nForce 420 “the sister of Xbox”
Implications

- Still disk-bound
  - Both speed and capacity continues to improve
  - Leaves >80% of CPU power for other uses
  - As BM DMA should; can use CPU for SW formatting
- Because nVidia sustains VSIB using PCI32/33 it will sustain 1Gbit/s Ethernet at disk speed
  - Without actually having PCI64/66
  - VSIB board and 1Gbit/s Ethernet can coexist on the same PCI32/33 bus, not used at the same time
No Hot-Swap?

- Two PCs required for 1Gbit/s => can provide 512Mbits/s “hot-swap” for free
- “Real” hot-swap is expensive
  - (Until “office” PCs include it by default)
- Alternating PCs provide the same functionality:
  - Swapping disks while recording (==“continuous recording” without scheduling of disk/tape changes)
  - Recording (512Mbit) and transmitting (500–800Mbit) over the Internet simultaneously
Total System Sizing

- Scalable now and in the future
  - PC technology scales “automatically” on its own
  - 1 Gbit/s: May 2001, 4 PCs --> April 2002 only 2 PCs
  - 1, 2, 3, 4, 5... PCs... doesn’t really matter
  - Track–slicing, time–slicing, word–slicing...
- 1 PC replaces the current 2–head recorder
- “Pioneers” of high–bandwidth VLBI can buy 3 additional PCs for 2Gbits/s peak, 1Gbit/s uninterrupted continuous operation
Costs

- DAQ system 4--10 kEUR (2--5 PCs)
- Disks in April 2002
  - A set of 16 120GB disks 4576 EUR
    - Capacity of 3.2 thin tapes, 1430 EUR each
    - Will be around 3300 EUR in November 2002!
  - See what happened a year ago:
    - Disks in November 2001
      - A set of 16 80GB disks 3300 EUR
      - Capacity of two thin tapes, 1650 EUR each
      - Reduced 21% since May 2001, was 4200 EUR then
“VSI−F” File Format?

- Since both Mark5A and VSIB can produce files with compatible binary data contents it would make sense to start forming a common header.
- The basic idea of FITS of putting ASCII header into the beginning of each file seems good.
  - Easy and quick to read the headers of many files.
  - Can use commands like ‘more’ to browse them.
  - Just reserve some space for it when creating the file.
A Sample Header

VSIF_VERSION=1.0
START_OF_DATA=16384  # byte offset, first byte ==0
UTC_START=2002-04-12T23:59:59.0  # ISO 8601
DURATION=4.0  # seconds
SAMPLERATE=32000000  # integer times per second
CHANNELS=8
BITS_PER_CHANNEL=2
CHANNEL=0,LCP,skyl,skyu
CHANNEL=1,LCP,skyl,skyu
...
CHANNEL=7,RCP,skyl,skyu
NATIVE_WORD_BITS=32
NATIVE_WORD=7,6,5,4,3,2,1,0,15,14,13,12,11,10,9,8,23,22,21,20,19,18,17,16
,31,30,29,28,27,26,25,24  # location of VSI bits 31(MSB)..<0(LSB)
#Channel bits to native VSI bits:
MAP=c0b1t0,b0t0
MAP=c0b2t0,b1t0
MAP=c1b1t0,b2t0
...
MAP=c7b1t0,b14t0
MAP=c7b2t0,b15t0
MAP=c0b1t1,b16t0
MAP=c0b2t1,b17t0
...