

# SPYMON Status Report

---

3<sup>rd</sup> SVT Workshop

U.C. February 5-6, 2001

**M.R.** for the Rome Group



# Outline

---

- SPYMON goals
- Work Force
- SPYMON structure
  - Crate process(monitor)
  - Spy manager
  - The (G)UI
  - The Viewer
- Experience from the CR
- FISION bug
- Further Studies on spy buffer readout
- To do list
- Future developments
- Critical areas
- Conclusions



# SPYMON goal

---

- SPYMON will monitor SVT performance mostly by looking at Spy-Buffers
  - “Spy Buffer can be read out via VME without interfering with the event processing” (not really true, see later)
- Accessing these data can be very useful to catch various potential problems:
  - SVT hardware faults
    - Cable check
    - Board level failures
    - Data transmission
  - SVXII strips noise
  - Inappropriate SVT parameters (e.g. wrong beam position)



# SPYMON goal

---

- Spy-buffer monitoring program will implement different tasks and may or may not be composed of different subprograms:
  - collect statistics of occupancy - SVXII noise monitor (?)
  - check non-severe errors and count/display them
  - Collect status data from each board and display them
  - validate data through hardware emulation
  - use track parameters from the last merger to act as a (very fast) beam monitor (?)
  - Expert tools for debugging and analysing spy data

# SPYMON vs. TRIGMON



- There is some overlap in scope between Spy Buffer analysis and TRIGMON analysis off L3 data e.g.:
  - SVXII hits monitored elsewhere (but noise could affect differently SVT and SVX readout)
  - Could check hardware comparing input and output of SVT (but intermediate step crucial for a useful diagnostic )
  - SVT error info are propagated through End Event words to Level2 (but not easy to understand where error originate)
- Moreover SPYMON can access all level1 events
  - unbiased statistic (i.e. level2 reject and level2 accept)
  - greater rate (i.e. faster diagnostic)

# SPYMON Work Force



Coincide with Rome SVT group :

- Lucia Zanello
  - Overall, group coordination
- Subir Sarkar
  - Java GUI, root viewer, Smartsockets
- M.R.
  - VxWorks, SVTVME, FISION, RC

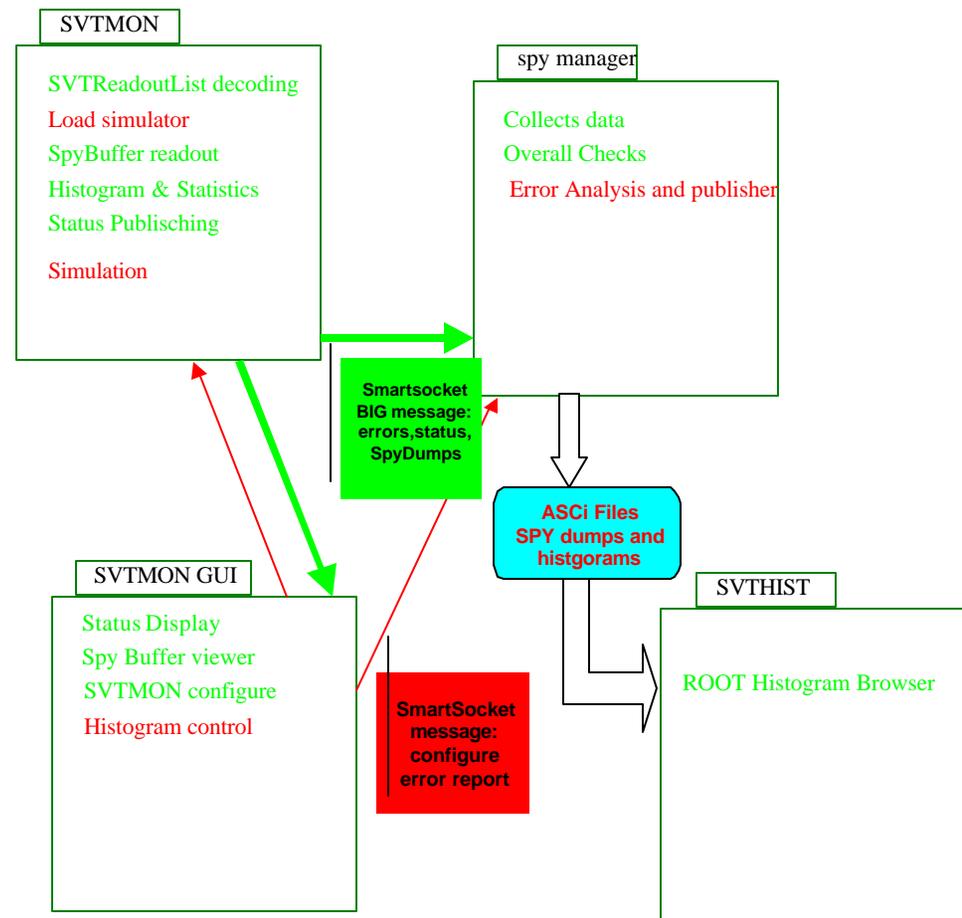
Alex Barchiesi doing mostly SVTVME GUI and (right now) working on the cold start..

- Hope to get important feed-back for SPYMON in the SVT configuration area of the program



# SPYMON Structure

- i. SPYMON is a VxWorks monitor program (svtmon) in communication through SS with the outside world
- ii. SPYMON is also a UNIX (spy-manager) process for the overall coordination of the monitoring task
- iii. SPYMON has a **java GUI** for displaying results and specifying the task
- iv. SPYMON includes also a ROOT viewer and converter SVTHIST





# Svtmon: status

---

- Svtmon start waiting for the partition command from RC
- Once it receives the r/o list allocates space for receiving spy-buffer data and for histo
- Wait for a specific command from the spy-manager to start a *freeze/read/analyse* iteration
- When done **publish** requested info over **smartsocket**



# Svtmon: future

---

- Svtmon start retrieving RC related status **without explicitly waiting for a partition comand**
- It allocates space for receiving spy-buffer data and for histo **using a default configuration for each crate and wait for the coldstart only to load simulators.**
- Wait for a specific command by the spy-manager to start a freeze/read/analyse iteration **or initiate it with a timer or when error detected (what about SC interrupt?)**
- When done publish requested info over smartsocket



# Spy-manager

---

- Right now it does very little yet important things:
  - Issue a /spymon/freeze command every n sec.
  - Collect data from the crate, organize it and write big ASCII file for histograms and spy raw-data, time stamping them.
  - Some of the parameters will be configurable through the GUI.
  - Will do some further checks on the data/error received including simulation as an option.
  - Will likely publish error messages to merlin/run control
  - Need some way to start automatically even if not tightly coupled to RC

# SPYMON GUI

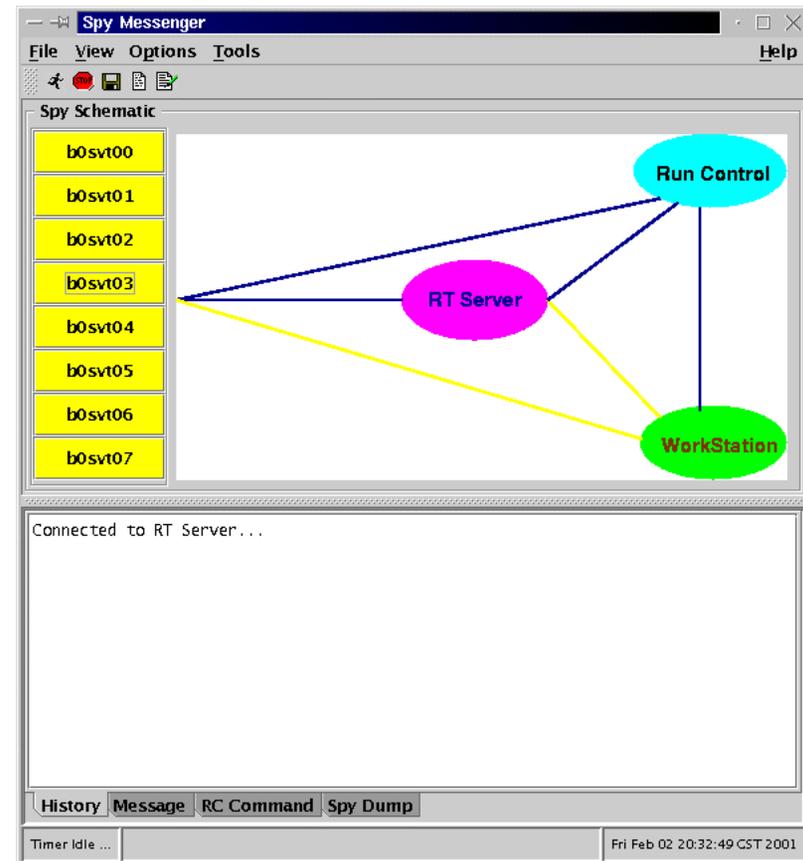


- Is the major addition to the soon to be released version 1.0
- Much work from Lucia/Subir done in November/December, slowed down because of Subir leave
- Still very rough graphic (like we need more of an artist...)
- Basic functionality and hierarchy of windows/menu enough to start with
- For the time being the display part is far more developed
- Need to test extensively and to implement many options and configuration tools

# SPYMON GUI



- Typing the appropriate command the user is presented with this window that allows to open all of the SPYMON related windows and menus
  - SVT status
  - SPYMON configuration
  - Board status
  - Start histogram viewer
  - Debugging tools

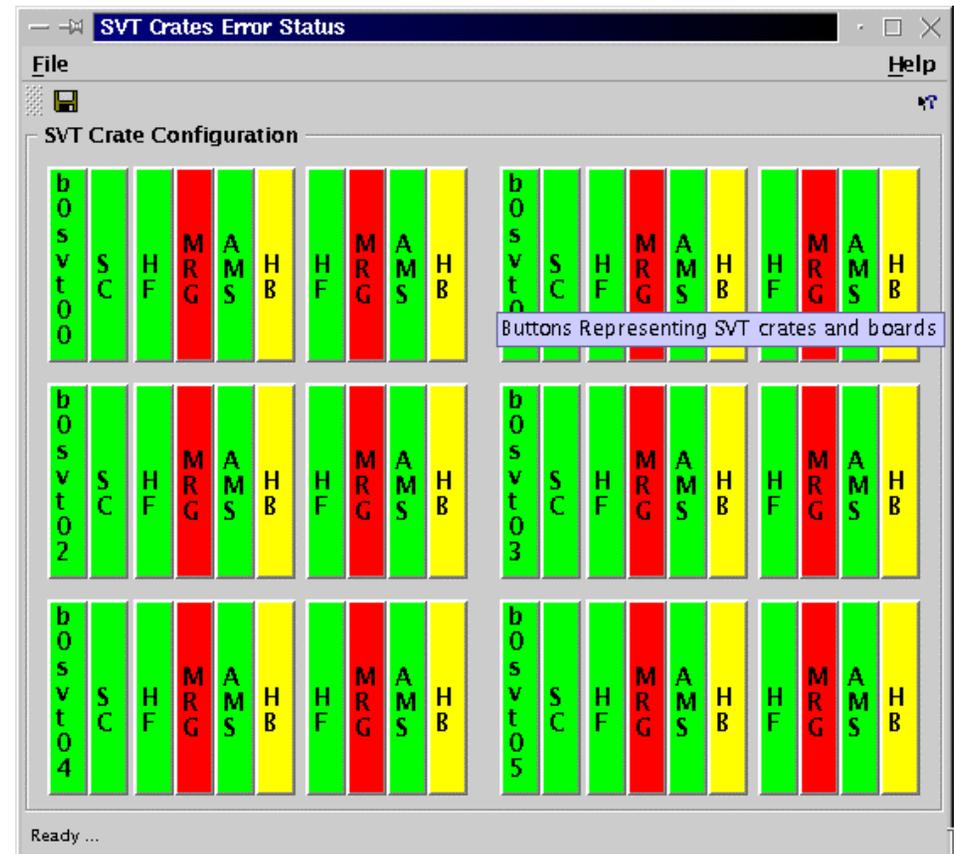


# SPYMON GUI

## SVT status prototype

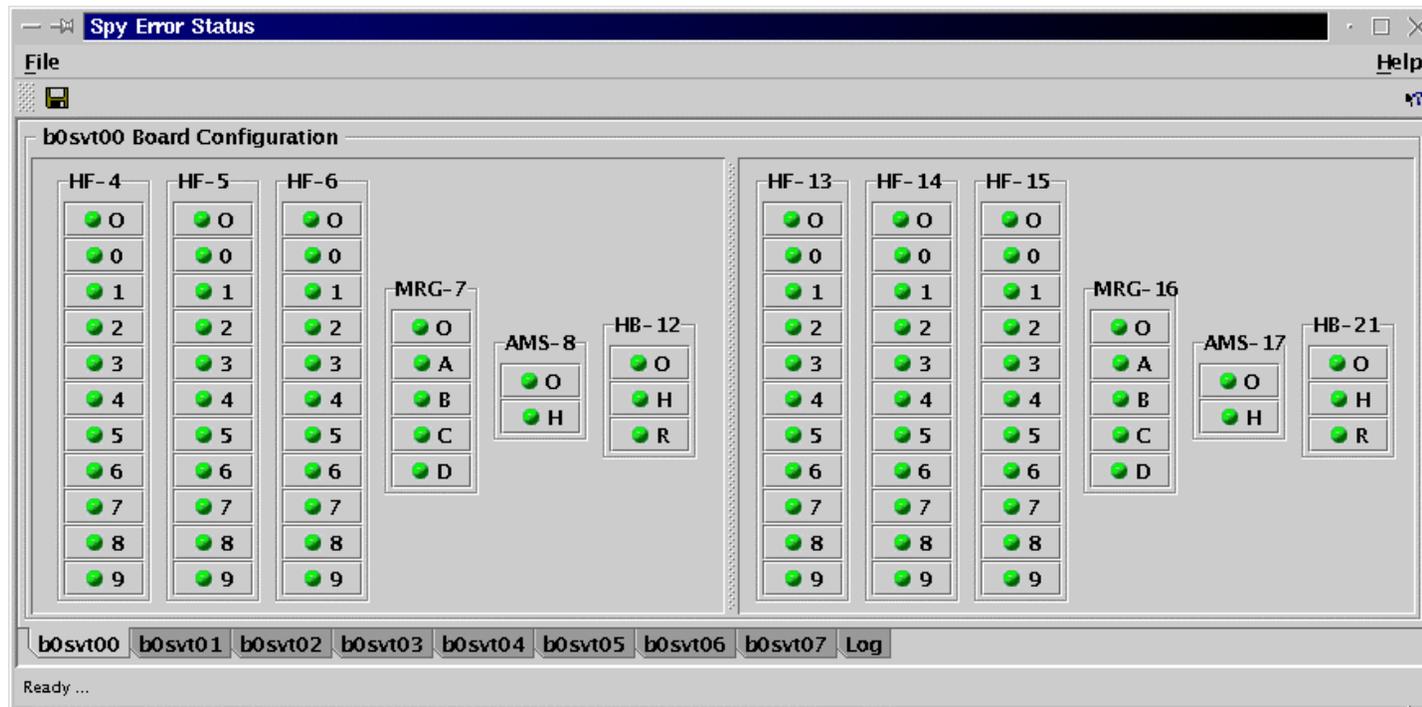


- Eventually it should convey enough information that everybody can understand the basic question: SVT OK or NOT OK?
  - Color code for typical errors
  - CDF error/SVT error
  - reproduce the status of the tiny/pretty SVT lights
- Need some artistic touch!



# SPYMON GUI

## SVT status prototype



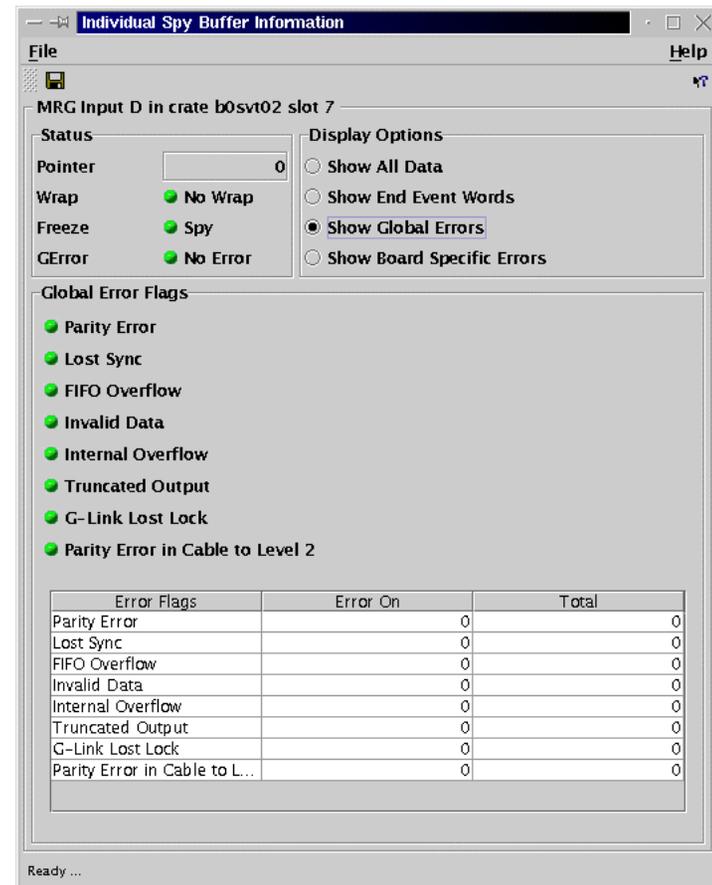
Clicking on the appropriate crate/board a summary info from each of the SVT spy buffer is presented

# SPYMON GUI

## SVT status prototype



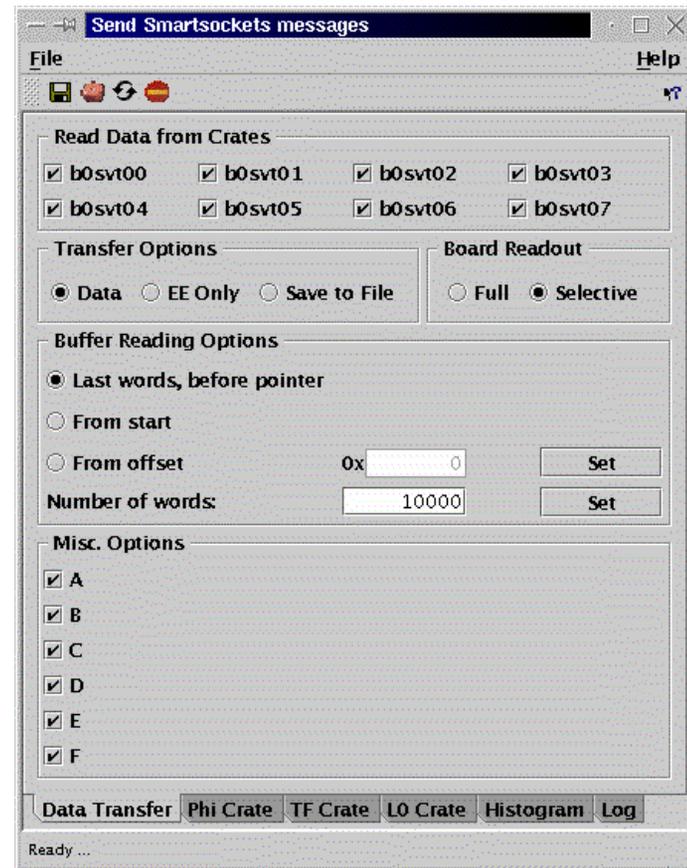
- A click on any spy buffer summary info take the user to the detailed info window for the appropriate buffer, where spy dump fragment and statistics is displayed



# SPYMON GUI Configuration Panel



- Configuration window:
  - Readout frequency
  - Readout list for svtmon
  - Disk data saving parameters
  - Histograms/statistics related command (reset,publish etc.)
  - Option to ignore DAQ status...



# Experience From The CR



- Despite the failure to run a fully functional SVT monitoring we learnt much:
  - Better understanding of the DAQ constraints for the SVT monitoring design
  - Clearer picture of the pros and cons of letting SPYMON be a full client of run control
  - Understood how harmful could be a FISION bug



# The FISION bug

---

- We discovered a bug potentially dangerous for every system:
  - Two tasks running at the same time on a single crate controller could perform wrong vme I/O due to a bug in the semaphore handling inside FISION
- FISION version 2-15 addresses this problem (Thanks Jim and Yuji).
- Tested OK on **b0svt05**
- Jim very cautious on loading that version on all the crates, but no other choice for us.

# Further Study on VME I/O

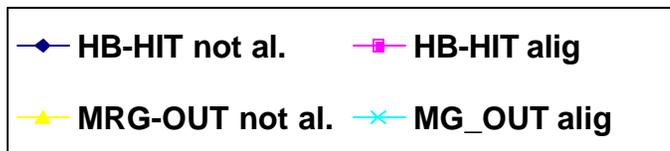
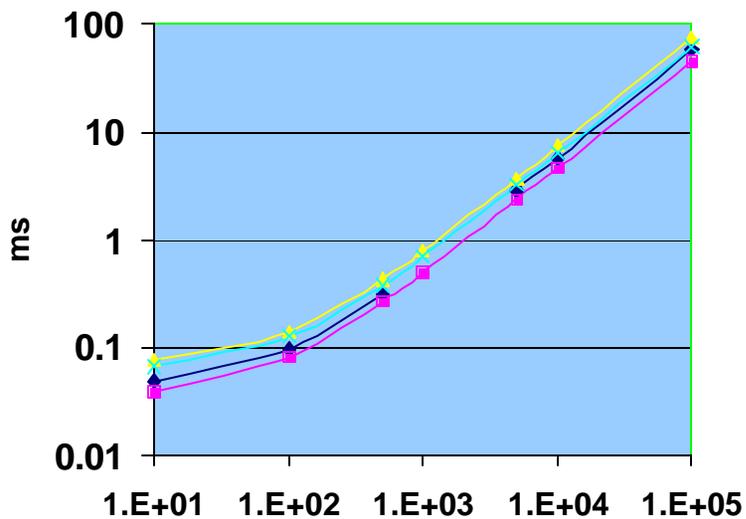


- The most important point for the peaceful coexistence of DAQ process and SVTMON is to let the former access the VME bus (to set DONE register on the TRACER) with a max latency of  $\approx 0.5$  ms (this being the longest time a typical event is readout from the slowest crate).
- Beyond this limit unwanted dead time would be introduced by SPYMON.
- To handle this we could split spy-buffer readout in many smaller read operation, but how much will it cost?

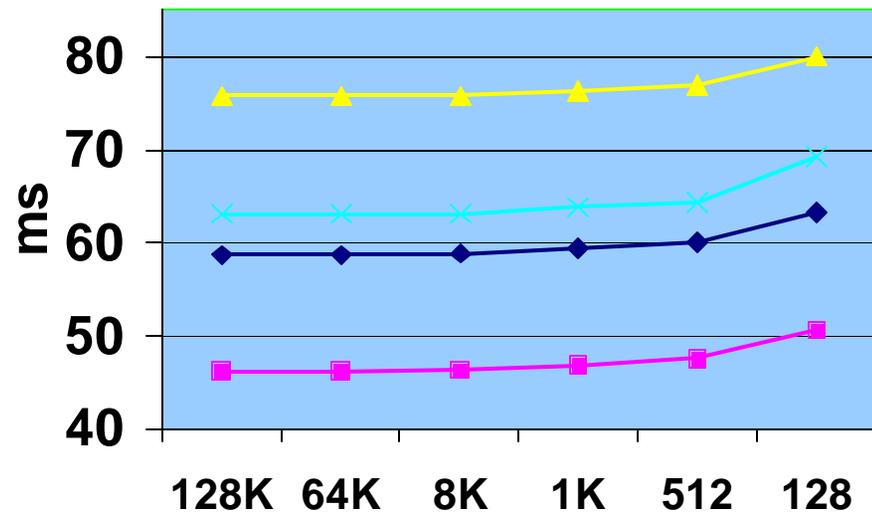
# VME I/O optimization 1



R/O time vs Total Words Transferred (for 128K Max Transfer Size)



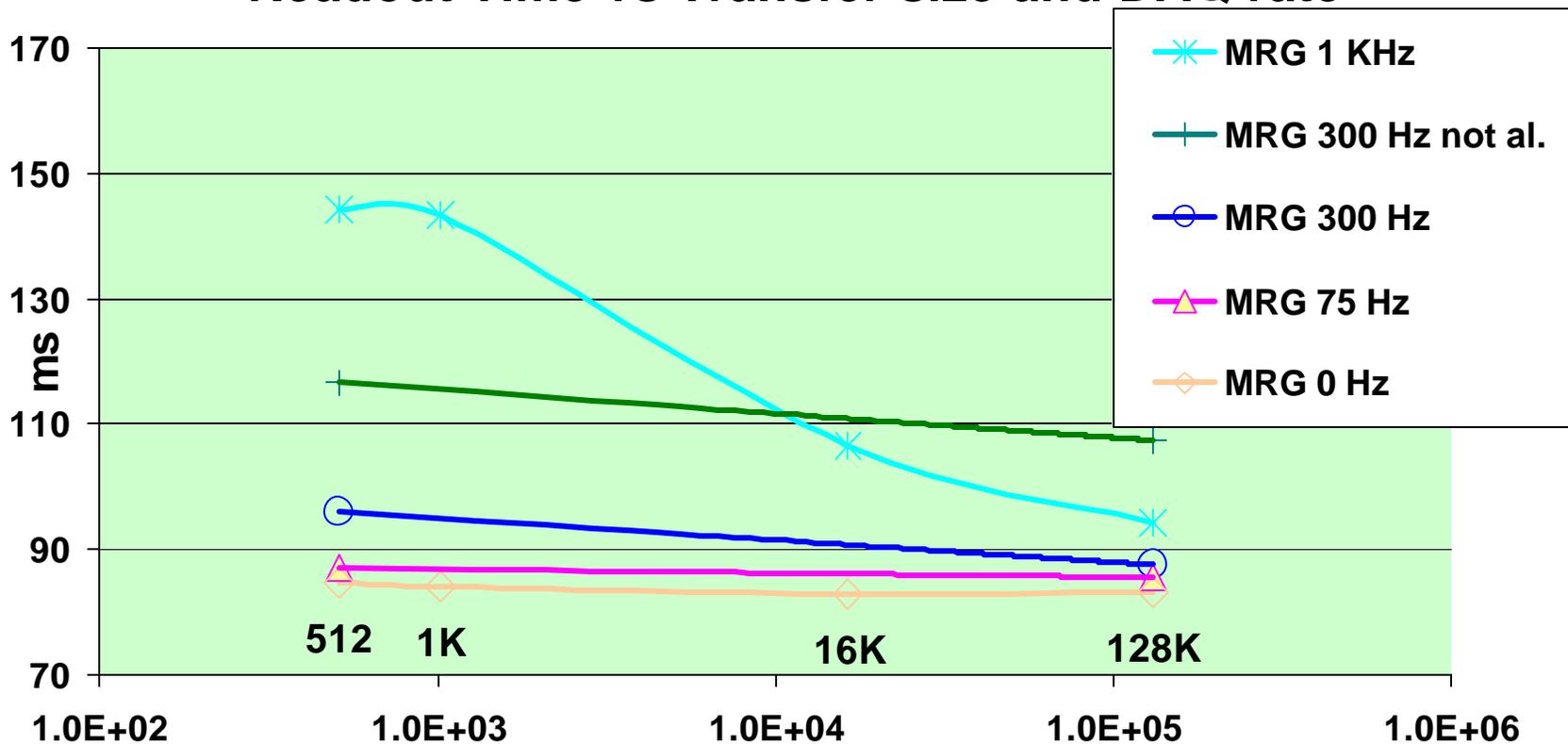
R/O Time for 100K Words vs Max Transfer Size



# VME I/O Optimization 2



## Readout Time vs Transfer size and DAQ rate





# VME I/O Optimization 3

- If RC has to be there we might have to be smart enough
- VME readout speed for crate process depends on L2 rate with an acceptable degradation up to the design 300 Hz

<b>A</b>	128 K	64 K	1K	512
1 KHz	915	945	995	1KHz
300 Hz	281			300
75 Hz	74			75

<b>B</b>	128 K	64 K	1K	512
1 KHz	85 ms	124 ms	159 ms	160 ms
300 Hz	74 ms			81ms
0 Hz	69 ms			71 ms

(A) Effective L2 rate and (B) VME average I/O speed (MRG-OUT) vs. nominal L2 rate and packet size



# To Do List

---

- Improve histogramming
- Test with a fully loaded prototype (hope not to discover new problems: e.g. how good is the SmartSocket for managing medium/large amount of data)
- Measure how fast we are at digesting the spy-buffer data (what's the exact deal with MVME2304?)



# Future Developments

---

- With respect to the “original” plan some pieces still missing:
  - Display of spymon results (histos, tables etc.) in the consumer display program (some news ... *do we break here the no C++ rule?*)
  - Remote user interface: thinking on how to adapt much of the Java to WEB based display programs.
- For some other plans have changed:
  - Can we assume nobody will need online spy-buffer data apart from Luciano’s beam alignment program?



# Critical Areas and plan

---

- I'm willing to respond to Bill call for work on simulation issue:
  - Find best resource allocation on crate cpu's
  - Make a better case to INFN for purchasing 2304
  - Understand how to validate simulation vs hardware and being able to tell who's wrong
- Need to extensively test svtmon/spy manager robustness against every day usage by shift crew (integration test)
- Better GUI/display much needed (Subir/Alex)



# Conclusions

---

- It's time to rush!
- We are excited that many pieces are finally emerging
  - Simulation
  - Coldstart
  - Databases
- We are on schedule for delivering the basic functionality (CR style + histogramming) in 1 month time.
- Nicer or more ambitious functionality will come later (and help here more than welcome!)