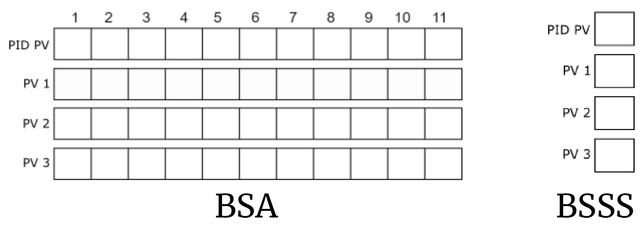
ATRIUM: Automated Test Regression for IOCs Under Measurement

Grace Chen[12], Márcio Paduan Donadio[2], Michael Skoufis[2]

1) University of California, Davis
2) Linac Coherent Light Source, SLAC National Accelerator Laboratory

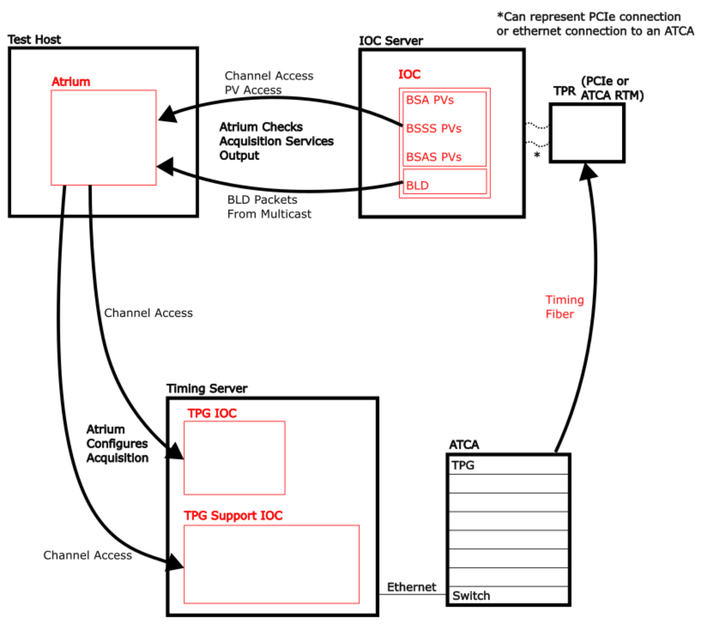
ACQUISITION SERVICES

The purpose of the acquisition services is to report the condition of the accelerator, outputting data measured by sensors for each electron bunch. Each electron bunch has a PID (pulse identification number) and other characteristics that are collected and presented either as PVs (process variables) or sent directly to a multicast network. BSA (beam synchronous acquisition) stores the data into arrays where the same index of the different PVs marks data belonging to one electron bunch. BSSS (beam synchronous scalar service) stores the most recently acquired data as a scalar for each PV.



BSAS (beam synchronous acquisition service) presents data with statistics processing in PVs in a tabular format and BLD (beamline data) reaches clients through a multicast network.

PURPOSE OF ATRIUM



Atrium is a Python software that performs regression tests by reading the output of the acquisition services to ensure that any changes in the IOC (input output controller) under test does not affect the output. Atrium configures the TPG (timing pattern generator) IOCs which will send the configuration to the TPG carrier board through the ATCA switch. The TPG board in the ATCA sends timestamps, PIDs, and other event codes to the TPR through a timing fiber. The TPR is connected through either a PCIe or ethernet connection to an ATCA to the IOC server. Atrium then acquires the PV data through Channel Access, PV Access, or BLD packet via multicast.

EPICS

EPICS is a toolkit used to develop control systems which simplifies the process of passing data between clients and servers in the network. Atrium utilizes PyEpics, EPICS Channel Access for Python, to configure the TPG IOCs and retrieve PV data from the IOC servers.

CODE SUMMARY

atrium.py

- Parses the command line and stores arguments to settings.py
- Gets the st.cmd (startup) file from the IOC under test and retrieves information about prefixes and suffixes of PVs
- Main function that calls on system and user buffer tester

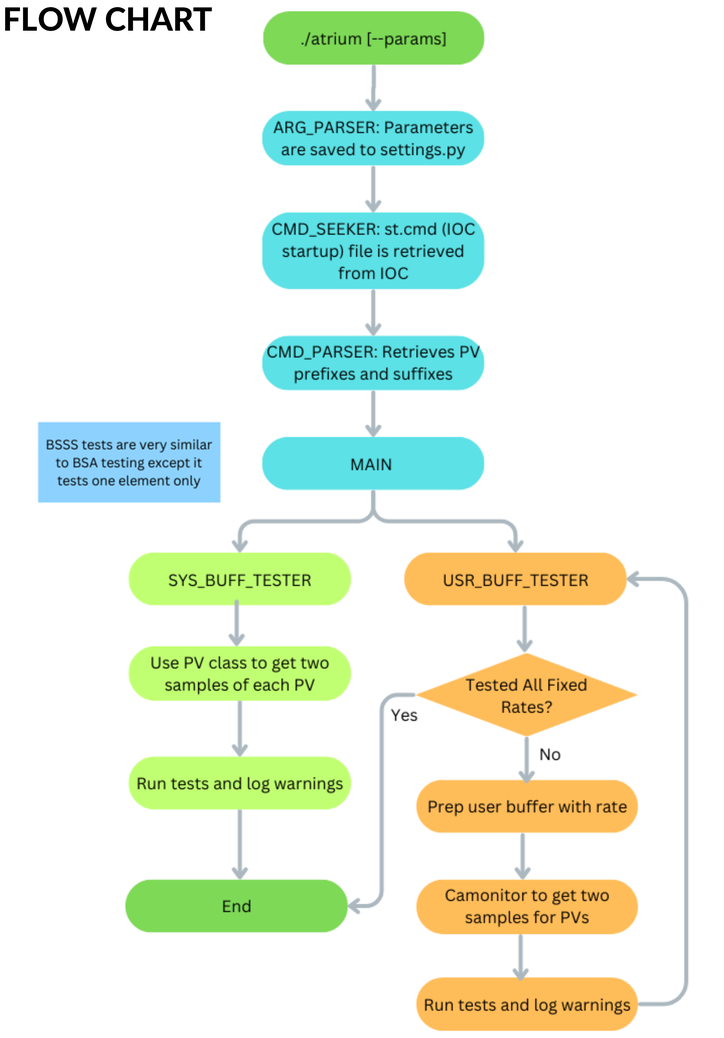
settings.py

- Shared global variable file
- Stores information about the CPU, TPG, IOC, user buffer numbers, system buffer destinations, and prefixes and suffixes of PV names

sys_buff_tester.py + usr_buff_tester.py

- Construct PV names based on prefixes/suffixes
- Get two samples of each PV; use PV class for system buffers and camonitor for user buffers
- Run tests on each PV data:
  - PV is populated (nonzero length)
  - There is no NaN's in PV data
  - Two samples are different (data changes over time)
  - Check if data changes within one sample
  - For PID PVs, check if PIDs are consistent with fixed rate
  - For user buffer PVs, check if number of elements is correct

FLOW CHART



ATRIUM INPUT

./atrium --cpu cpuName --ioc iocName --tpg tpgName --usr_buffs [bufferNumbers] --sys_buffs [bufferNames]
(--bsa_usr_buff_samples numOfSamples or --bsa_usr_buff_max_time maxTime) --test-type testType
./atrium --cpu cpu-b084-sp17 --ioc sioc-b084-gd01 --tpg TPG:B084:2 --usr_buffs 21 --sys_buffs SCS
--bsa_usr_buff_samples 5 --test_type both

ATRIUM OUTPUT SNIPPETS

BSA System Buffer Test Example

```
*****<< EM2K0:XGMD:HPS:milliJoulesPerPulsePIDHSTSCS1H >>*****

First sample   --> [3.25741227e+14 3.25741228e+14 ... 3.25759424e+14
 3.25759424e+14]
Second sample  --> [3.25741228e+14 3.25741229e+14 ... 3.25759424e+14
 3.25759425e+14]
PV Populated With Data:      OK
PV Does Not Have NaNs:       OK
Pair Of Samples Diff:        OK
PV Updated With New Data:    OK

Checking PID rate.:

Reading sample --> [3.25741229e+14 3.25741230e+14 ... 3.25759425e+14
 3.25759426e+14]
PID Update Rate:             OK  (1.0Hz)
```

BSA User Buffer Test Example

```
!!!!!!!-- RATE 1Hz --!!!!!!!
*****<< EM2K0:XGMD:HPS:milliJoulesPerPulseHST21 >>*****
First sample   --> [0.02311707 0.02197266]
Second sample  --> [0.15563965 0.02456665]
PV Populated With Data:      OK
PV Does Not Have NaNs:       OK
PV Updated With New Data:    OK
Pair Of Samples Diff:        OK

Tested only using 'First sample':
PV number of elements:       OK (2)
```

FEATURES I'VE WORKED ON

Additional command line arguments

- Add the IOC argument to specify the specific IOC to test
- Add an argument that limits the maximum time spent on user buffer acquisition
- Add an argument to test only user or system buffers

User Buffer Acquisition Structure

- Change from running user buffer acquisition for every PV to running acquisition for all the PVs together to improve Atrium's run time
- Run tests for all user buffer fixed rates instead of only 1Hz

FUTURE IMPROVEMENTS

- Change BSSS user buffer acquisition so that it could test more than one element
- Working on testing for BLD and BSAS acquisition services

ACKNOWLEDGEMENTS

I would like to thank my mentor Márcio Paduan Donadio for the guidance and support throughout the project. I would also like to thank all SLAC staff that made this internship a wonderful experience.