Argonne NATIONAL LABORATORY

EPICS V4 IOC/DB Requirements from APS

EPICS V4 Plans for SPX Applications

Andrew Johnson

U.S. DEPARTMENT OF ENERGY | Office of Science

SynApps Experience of V3 Database

- Comments on the V3 database model:
  - Processing model is good
    - Can develop complex processing and control behaviors using general-purpose records
    - Units string is important annotation (3.15 records copy units strings with input links)
  - Put-notify with completion signaled by forward link processing works well
    - Record type gets to decide when a complex operation has completed, not the IOC framework
  - DB design without considering locks, threads (other than lock-sets), or loops
    - Example: User can set energy or wavelength, DB can update the other without looping
  - IOCs are great for creating middle-ware
    - This ability to create layers makes even unintentional collaboration easy
  - Non-blocking requirement for records makes it hard to implement complex operations
    - Record has to keep track of its current state, work out what to do when next called
  - DB equivalents of complex operations (e.g. scalar record) are hard to do
    - Must be able to write record types to do complex operations

Recommendations for V4 Database

- Specific new functionality suggested
  - Analog value type with error bars
    - New Normative Type?
- Data Analysis use-case?
  - Beam-line scientists have never shared much data analysis code in practice
  - For the kinds of analysis some of them do, a V4 IOC database could be sufficient for their first-line data analysis
- Tim is positive about V4, but did make this comment:
  - “I don’t want to re-implement the 15 years of work that went into SynApps using the V4 database”

Additional Ideas

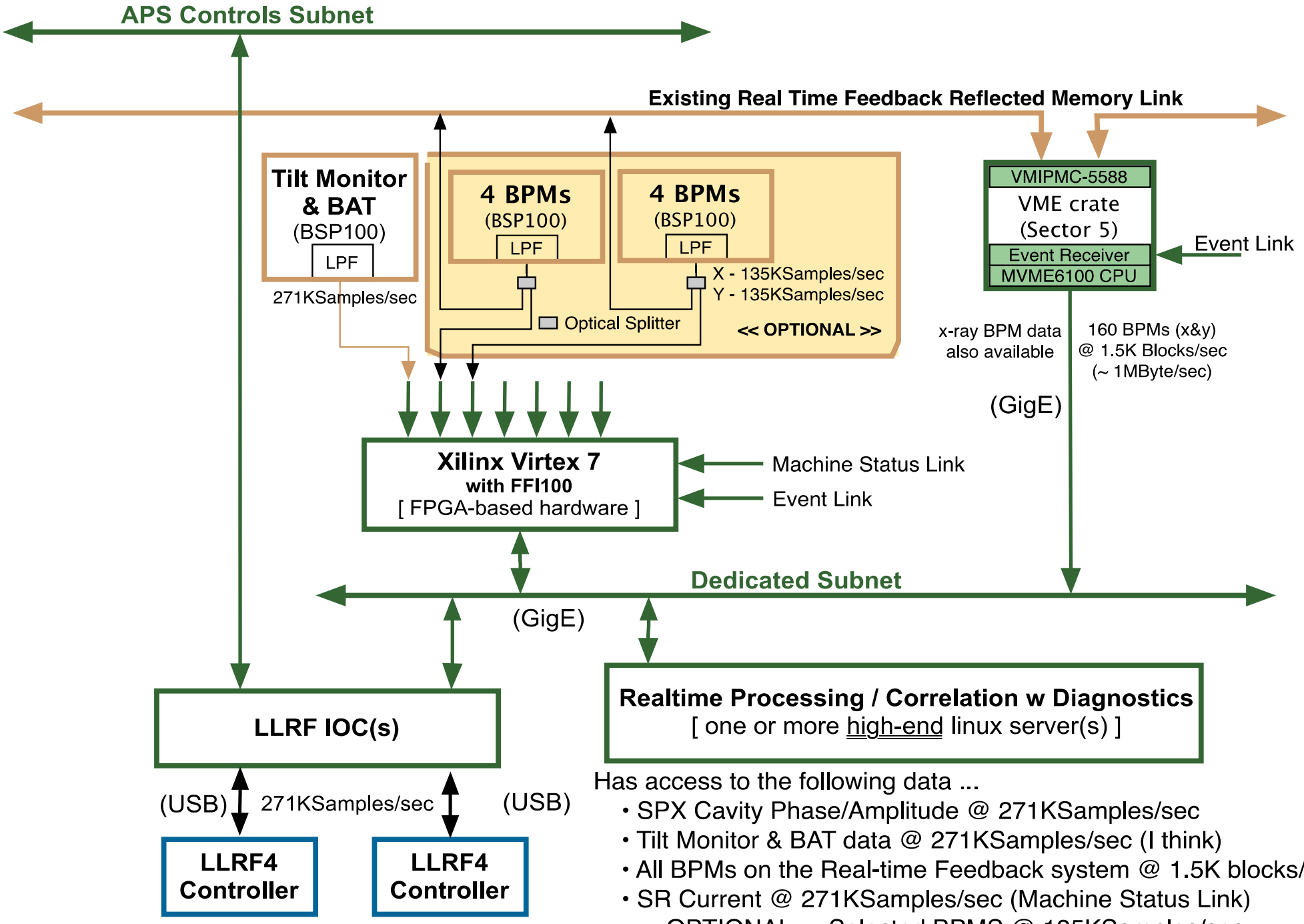
- Subscription parameters (may need a pvAccess extension?)
  - Support the same kinds of things we can do with server-side filtering on 3.15:
    - Client-specific monitor dead-bands, rate limits, conditional monitors etc.
- Reflection API, i.e. be able to ask an IOC what PVs it holds
  - Beamline IOC configurations tend to change frequently
    - We'd like GUIs to be able to adjust automatically to a change, when replacing a detector say
  - A reflection API (RPC?) could be used to populate the ChannelFinder directory and update the Directory Service automatically

SPX Diagnostics DAQ

- Need to collect different types of data from several input sources (EPICS v3 IOCs)
- All data has collection time and rates
- Different types of data are produced at different rates:
  - LLRF Fast Data (IQ Waveforms): ~13.6MB/s
  - LLRF Slow Data (RF Setpoints): ~216KB/s
  - BPM Data: ~1MB/s
- Data has to be processed and correlated for monitoring, diagnostics, and control purposes
- May need to store some of the collected data into files (SPX uses SDDS format) and catalog those in a relational database

- Looking at the Gather service for data collection. May run both V3 and V4 protocols

SPX0 Diagnostic Data Acquisition Option



Has access to the following data ...

- SPX Cavity Phase/Amplitude @ 271KSamples/sec
- Tilt Monitor & BAT data @ 271KSamples/sec (I think)
- All BPMs on the Real-time Feedback system @ 1.5K blocks/sec
- SR Current @ 271KSamples/sec (Machine Status Link)
- <<OPTIONAL>> Selected BPMS @ 135KSamples/sec

nda
1/10/13

SPX LLRF MATLAB Simulator

- MATLAB application will be able to simulate LLRF board and needs to be able to communicate with EPICS v3 IOC
- Fast (IQ waveform) and slow (RF Setpoints) data are fed into the LLRF IOC (EPICS v3), either from the LLRF controller, or from the LLRF simulator
- LLRF IOC provides access to various diagnostics and control utilities (visualization clients, various special purpose clients)

- If V4 is more suitable we want to use it, but it’s currently far from production ready

SPX LLRF MATLAB Simulator