DataFlow Performance Projections

Amedeo Perazzo, Matt Weaver, Christopher O’Grady

We’re “the deadtime people” who do high performance software and some electronics.

Overview of the system:
• 194 CPU system that reads out Babar over ~325 optical fibers
• At current event sizes, able to read out Babar at 5.7kHz with a pulser trigger (1% deadtime), and 4kHz with Poisson triggers (4% deadtime). Difference created by a firmware problem in trigger data readout: being fixed by Su Dong

In the future, event sizes and trigger rates will be larger.
• Have devoted a significant amount of dataflow manpower to helping PEP (real-time injection monitoring, and the coming real-time analyzed babar plots and archives).
• Hope to devote more time to PEP in the future, in part because the original dataflow designers created a system that “kicks ass”, so we can work on other things, and PEP is the most important.
• Not trivial to figure out where we can be most useful.
“Picture” of System

- A series of parallel assembly lines, with event buffering in between the workers
- System runs as fast as the slowest worker
How We Predict Future Performance:

- Wrote software (~1 year) to measure critical size/time parameters
- Measure slope+intercept of size/time parameters versus sum-of-beam-currents (using one no-beam run, and one colliding beam run)
- Simplified view: “Less Projecting, More Fixing”
- Check the linear assumption with 1 years worth of data
- Use the sizes to compute the speed of each “worker” and compare with trigger-rate predictions (e.g. 5kHz implies each worker must be <200us)
- Focus on bottlenecks, and do more in depth checks (e.g. with 2002 background runs). This talk purposefully leaves out this detail.
- Even the crude approach allowed us to predict the DCH readout bottleneck we see today during trickle injection years in advance
2007 thru 2009 (Sept. 2003 Seeman Model)

This reflects a considerable amount of work already completed (“event batching”, custom gigabit ethernet, re-written software, split crates). Trigger requirement (with Z-trigger upgrade and “times-two background”): 5.5kHz (180us). In discussions to see if we can fix DCH bottleneck in 2005 (Colin Jessop at Notre Dame).

SVT/DRC bottlenecks relatively easy to solve by scaling the system. EMC might be tougher, but likely doable (more batching).

<table>
<thead>
<tr>
<th>Component</th>
<th>I+ (mA)</th>
<th>I- (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVT (1173/1217)</td>
<td>82</td>
<td>63</td>
</tr>
<tr>
<td>DCH (9614/2464)</td>
<td>266</td>
<td>133</td>
</tr>
<tr>
<td>DRC (2616/1369)</td>
<td>21</td>
<td>104</td>
</tr>
<tr>
<td>EMC (9600/147)</td>
<td>0</td>
<td>105</td>
</tr>
<tr>
<td>IFR (3800/884)</td>
<td>19</td>
<td>104</td>
</tr>
<tr>
<td>IFR (2700/474)</td>
<td>19</td>
<td>96</td>
</tr>
<tr>
<td>EMT (5900/1656)</td>
<td>80</td>
<td>91</td>
</tr>
<tr>
<td>DCT (1229/368)</td>
<td>19</td>
<td>55</td>
</tr>
<tr>
<td>GLT (610/830)</td>
<td>73</td>
<td>31</td>
</tr>
<tr>
<td>BBR(989781/55035)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison With 1 Year of Data Taking
Conclusions/Thoughts

- **The big bottleneck is the DCH readout** (seen already in trickle injection) and the trigger readout firmware. Sudong is fixing the latter. We’re talking about the former (that’s the tough one).

- For other components (SVT/DRC), we’re getting near the edge in 2007 for other systems. Believe we can control it.

Other thoughts:

- Maybe background “backpressure” to PEP from SVT+IFR will prevent event sizes and trigger rates from rising too far. Difficult to say.

- **We have periods of high deadtime we don’t understand** (e.g. 5/14/03 at 7:36 am, 15% dead at 3.8kHz). It may be that DCH crates create deadtime “independently”, so we must sum the deadtime from separate crates. Working on this.

- Sudong has shown that trigger rate projections for 2003 were too optimistic by 50%. Maybe we have to change “times two” background to “times four”? 