Analysis of Bubble Sorting Stratergies

Nicholas Mead, University of Manchester

March 10, 2016

Sorting Stratigies

There are 3 ways to decide is when the bubble sort is complete

~~Static~~: Sort for the worst case senario (64 SPP)

Semi Static: Sort for the number number of SPP in BCID
(Requires informaton on train size, and a counter.)

Dynamic: Complete when no more swaps are being made.
(Requires two cycles of zero-swaps. One per odd and even comparision.)

We need to find the most appropriate stratigy for time and resources.

Sorting Acceptance

The sorting unit's must have a maximum number of SPP

Any BCID that is not sorted must *'bypass'* the sorting

Again this is a issue of functionality over resources

Initial estimates for semi-static resources are low.

Input Data

Both *Semi Static* and *Dynamic* algorithums were run over the same dataset

The dataset used was the MC velo simulation from Dr Karol Hennessy

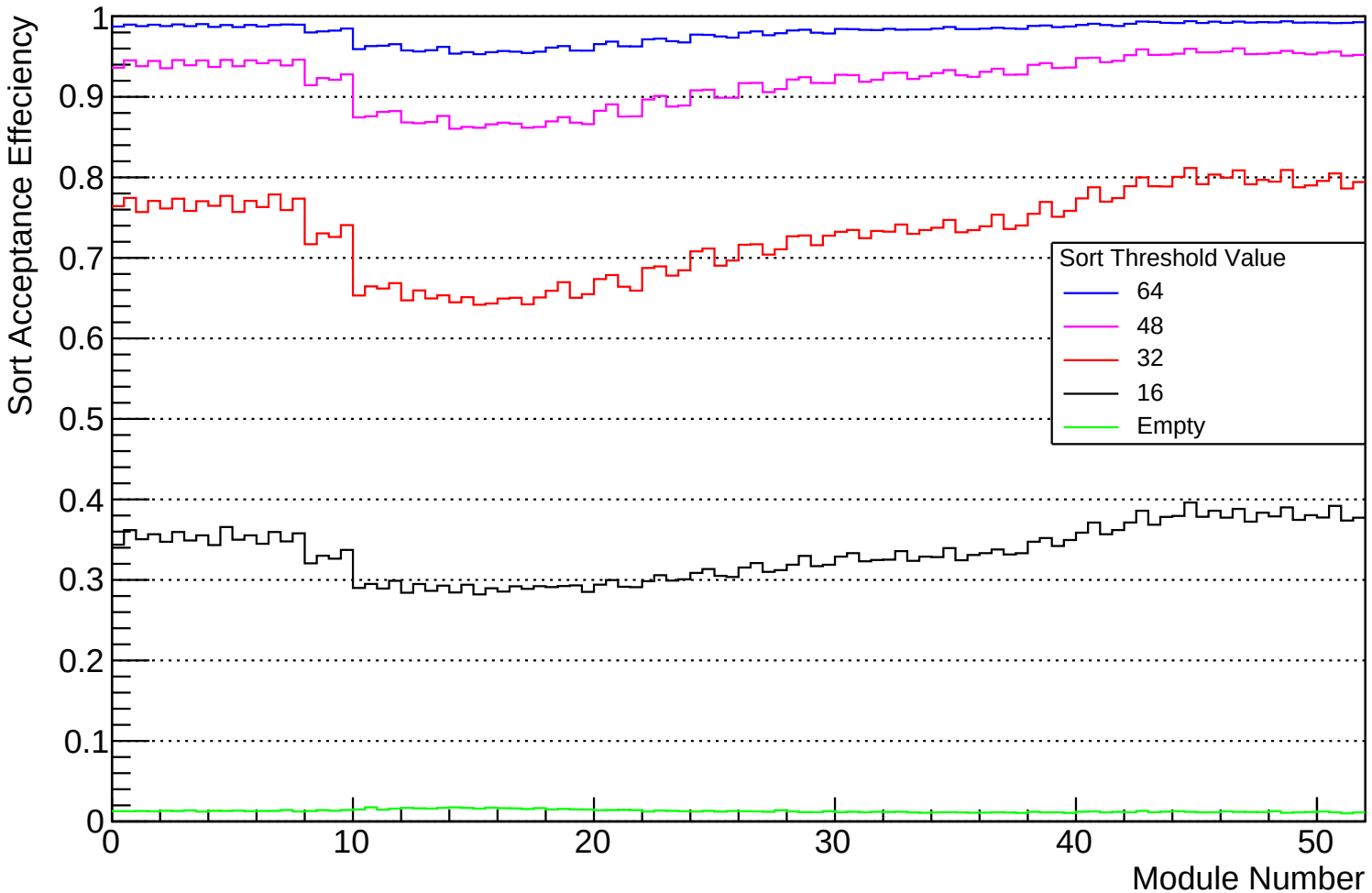
The data was organised by half module - as in the DAQ

$\sim$ 1 Millions BCID's worth of data was analysed.

Results

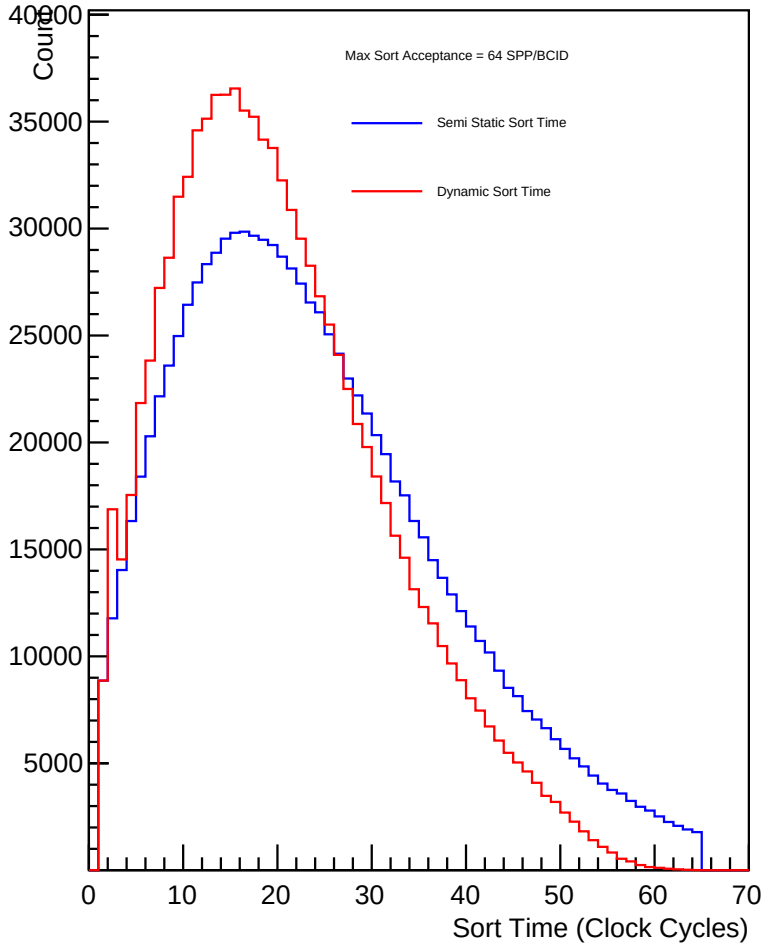
Sort Acceptance

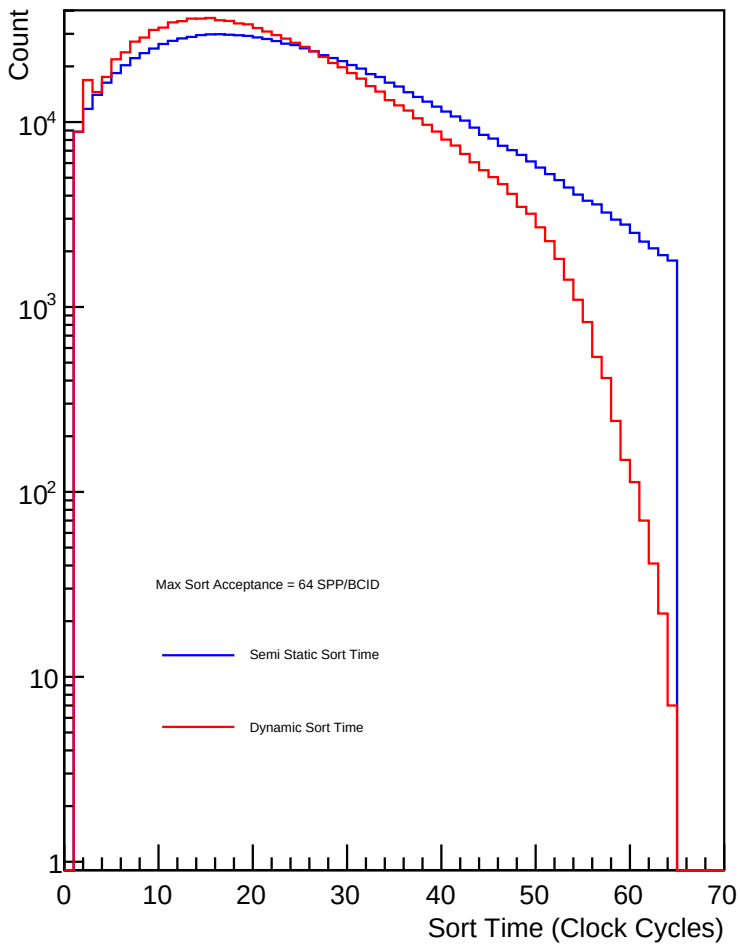
Figure 1: The fraction of accepted BCID, given for a range of sorting acceptance values.



Sort Time Comparison

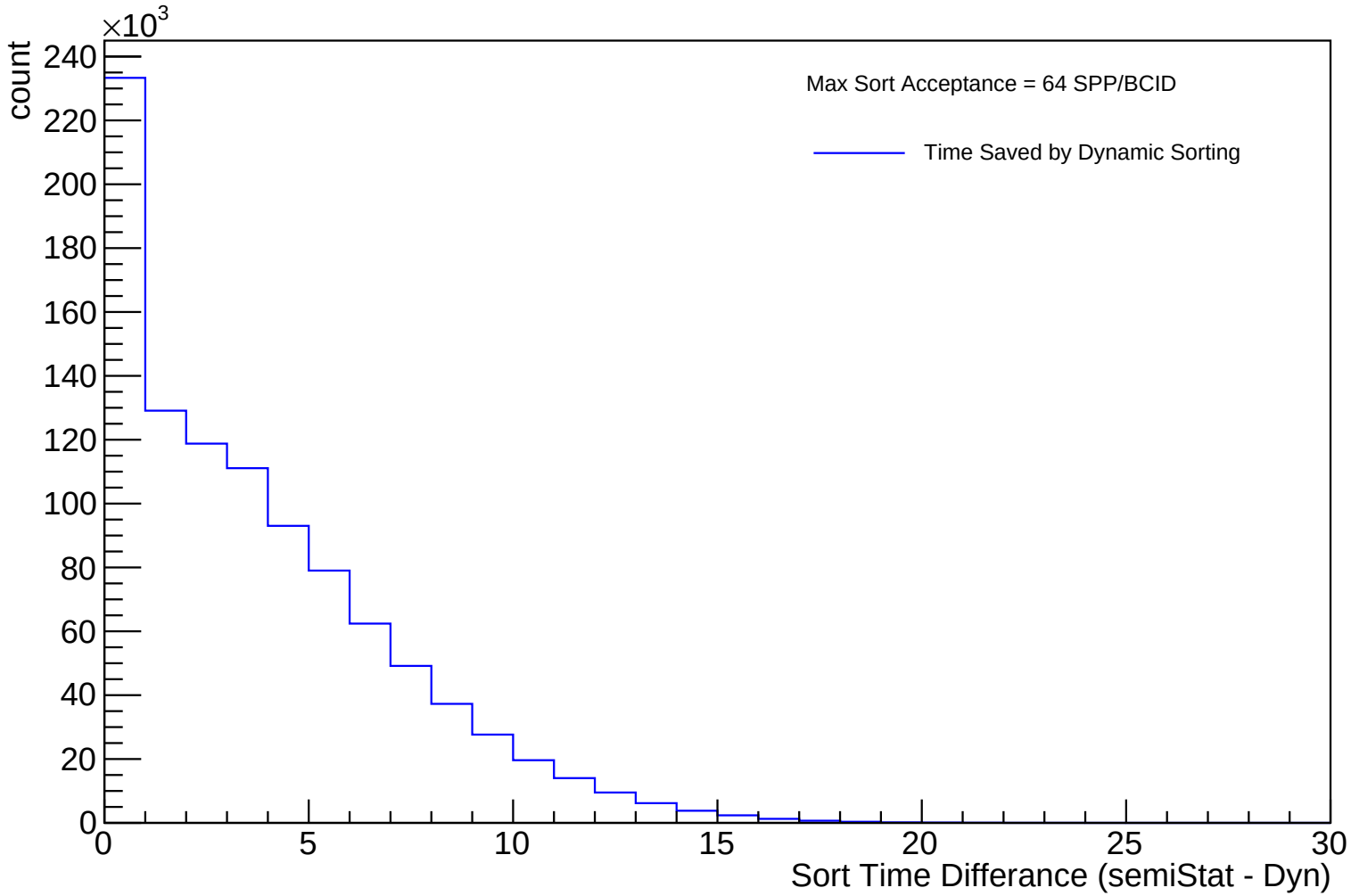
Figure 2: A comparison of the sort time's of the semi static and dynamic sorting methods. Left: Arithmetic scale, Right: Log scale





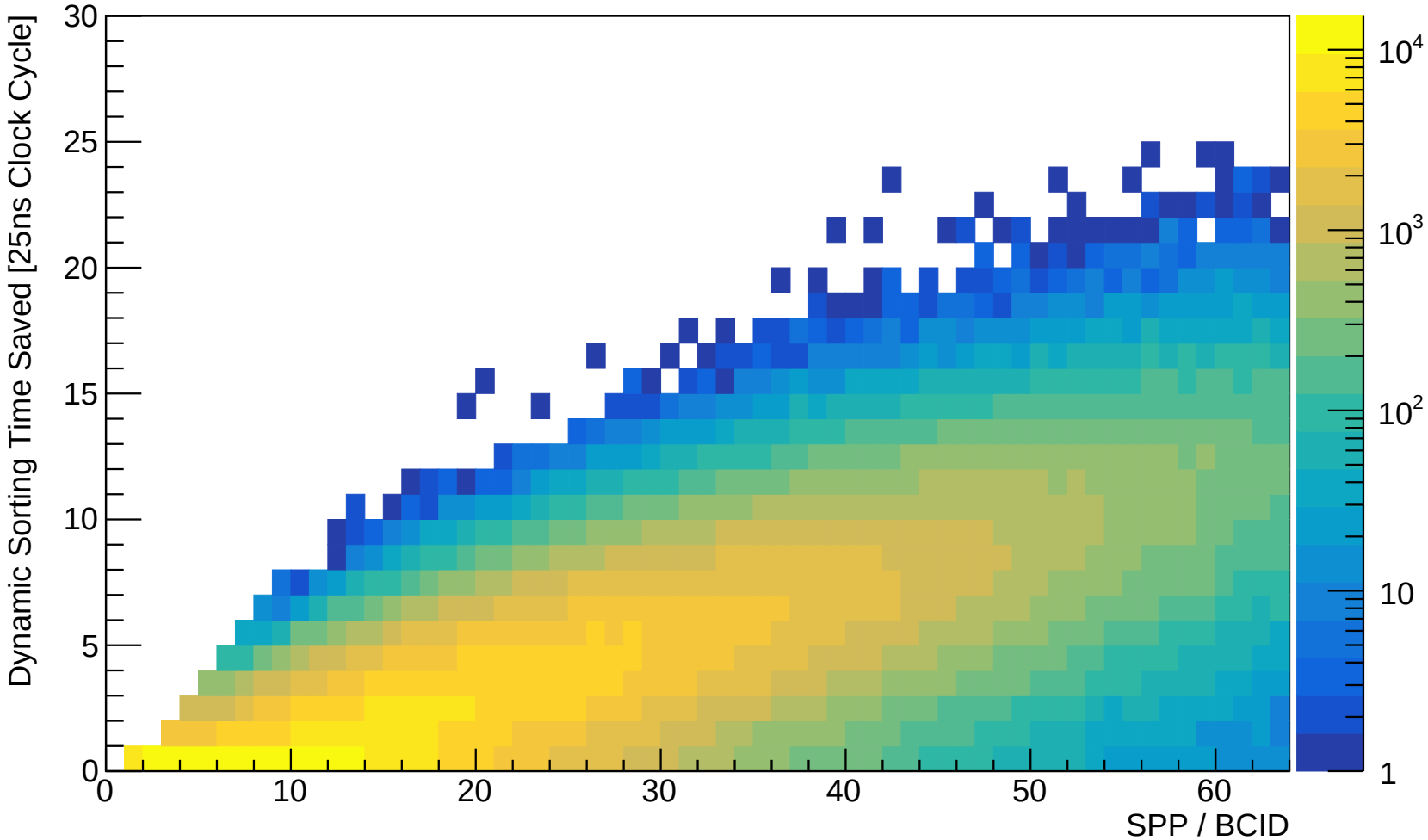
Time Saved by Dynamic Sorting

Figure 3: The time saved by the dynamic strategy compared to the semi static strategy.



Time Saved by Dynamic Sorting

Figure 4: The time saved by dynamic strategy (as before) as a 2D plot against the semi static sort time. **Note: log(Z) scale.**



Conclusion

Conclusion

Dynamic Sorting is only marginally more time effecient globaly

In the majority of cases, very few clock cycles are saved

There is not need implement a Dynamic sorting strategy.