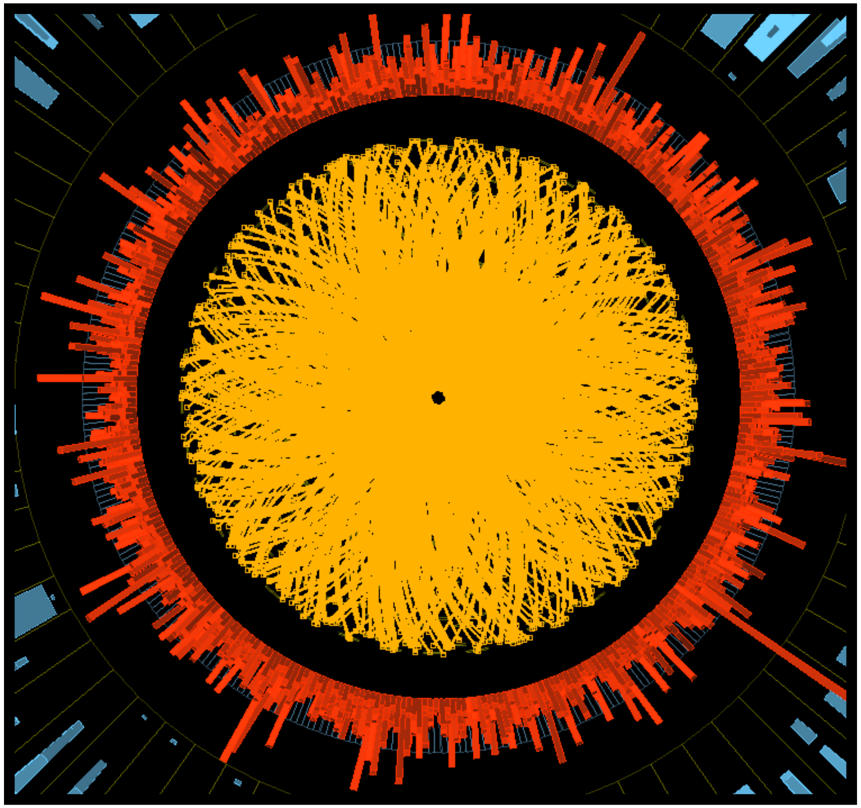
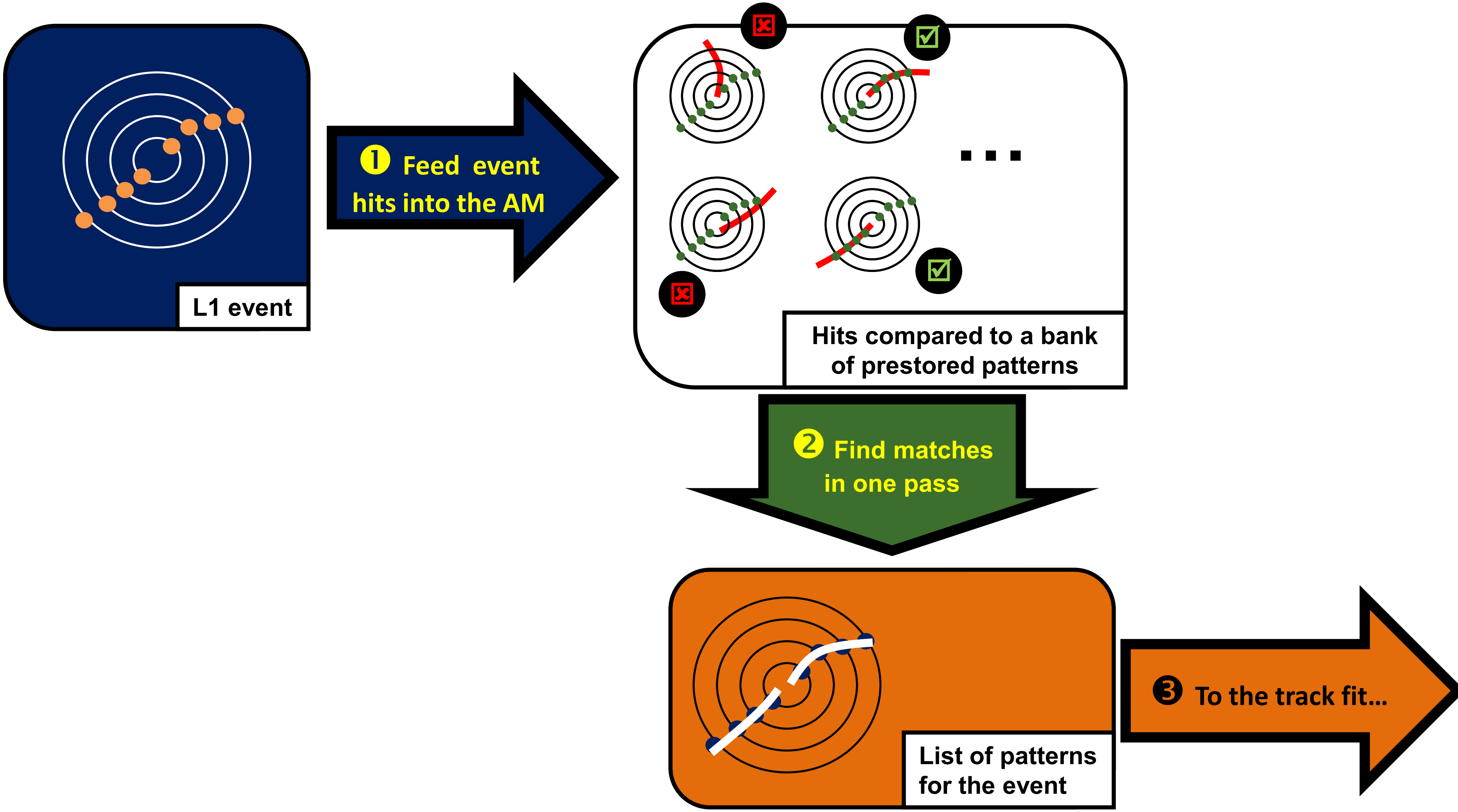
L1 track trigger with Barrel+Endcap geometry



1. Intro
2. Status
3. Plans

G.Baulieu, S.Viret

→ **Tracking trigger using associative memories (AM) in 1 slide:**



→ **Some of the main difficulties:**

❶ Feed event hits into the AM

→ **Which data rates should we expect at 20/40 MHz? Can we extract them towards the AM board?**

→ **Can the AM board input lines sustain such rates?**

❷ Find matches in one pass

→ **Can we do the pattern matching (*well, everything…*) in the available latency (*~3μs*)?**

→ **What is the maximum acceptable size for the pattern bank?**

→ **What is the optimal pattern bank?**

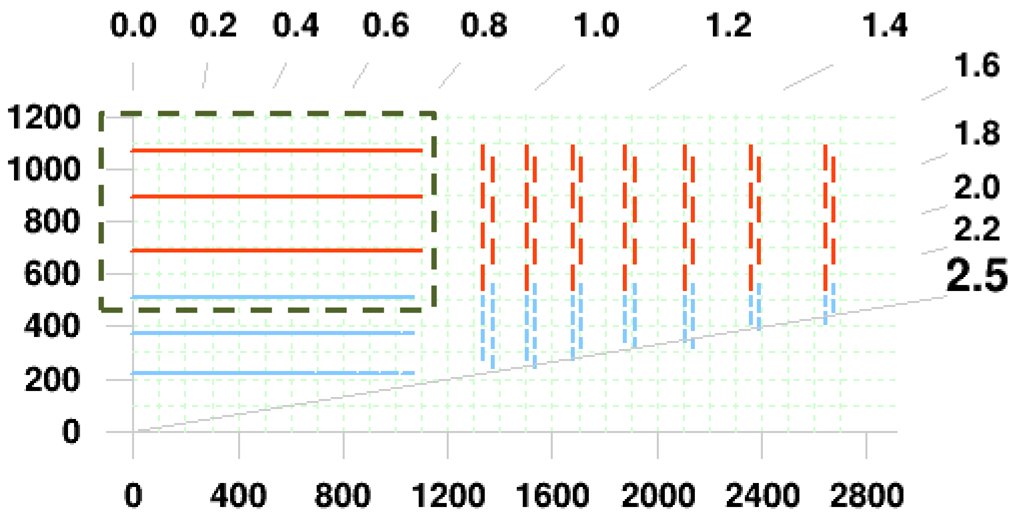
❸ To the track fit…

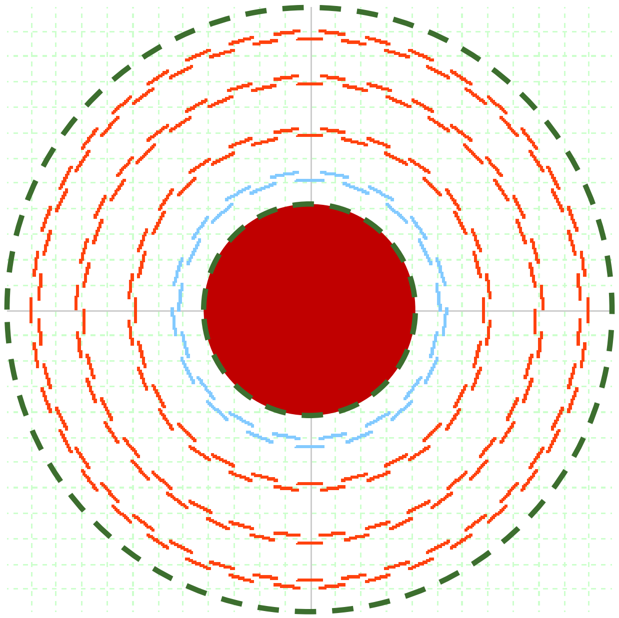
→ **What is maximum acceptable rate of matching pattern after the AM board?**

→ **How to remove efficiently the fake/duplicate pattern?**

→ **How to send the info to other L1 systems?**

→ **Barrel+Endcap geometry:**



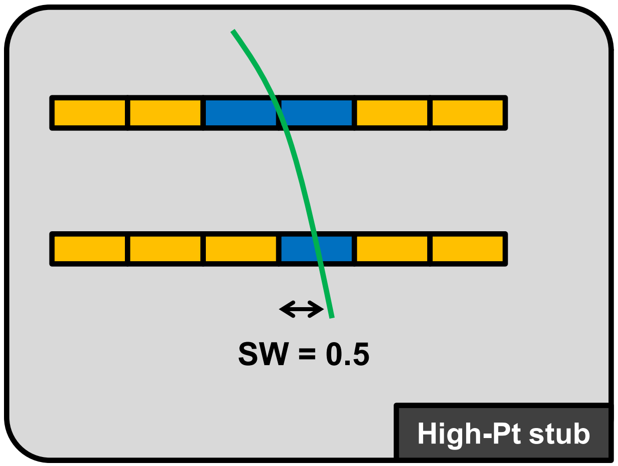


→ For the moment we consider only **the barrel part. Endcap is clearly another story** (*much higher rates*).

→ **Barrel = 6 layers of stacked modules**: **3PS** (*pixel/strips*) and **3SS** (*strips/strips*).

→ In this presentation we looked at the **4 outermost layers** (*but we used the* ***PS*** *layer as an* ***SS*** *one*). It is planned to use the pixel info at some point, but we keep it simple for the moment….

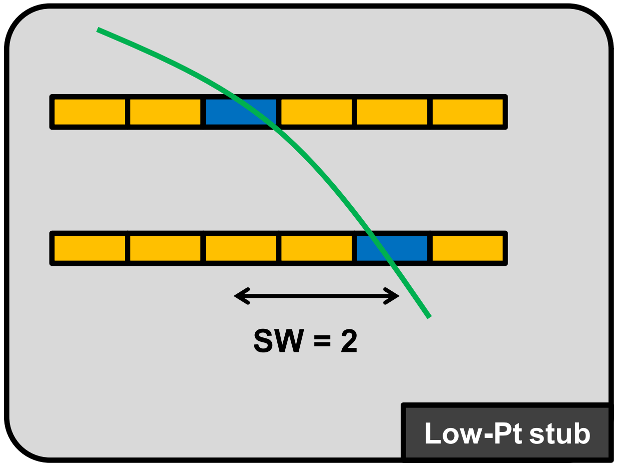
→ **Data reduction via stubs:**



→ Stubs are made of coincident clusters in the stacked modules. **We use them to construct our patterns.**

→ Using stubs instead of clusters **reduces data rate by ~1 order of magnitude** (*in* ***200PU*** *events*)

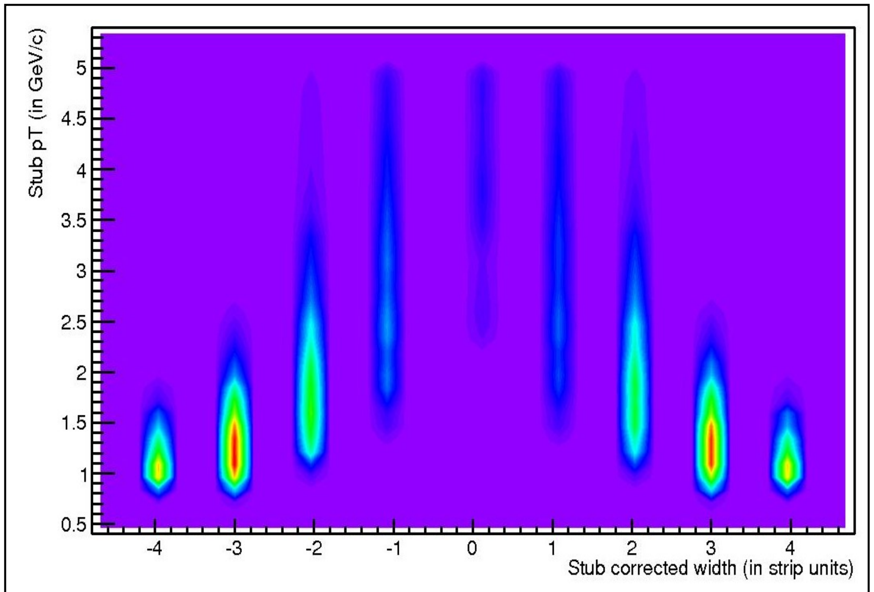
→ This rate reduction could be further enhanced by cuts on stub width (**SW**) and cluster width (**CW**). **Values of these cuts depends on the Pt threshold you're looking for.**



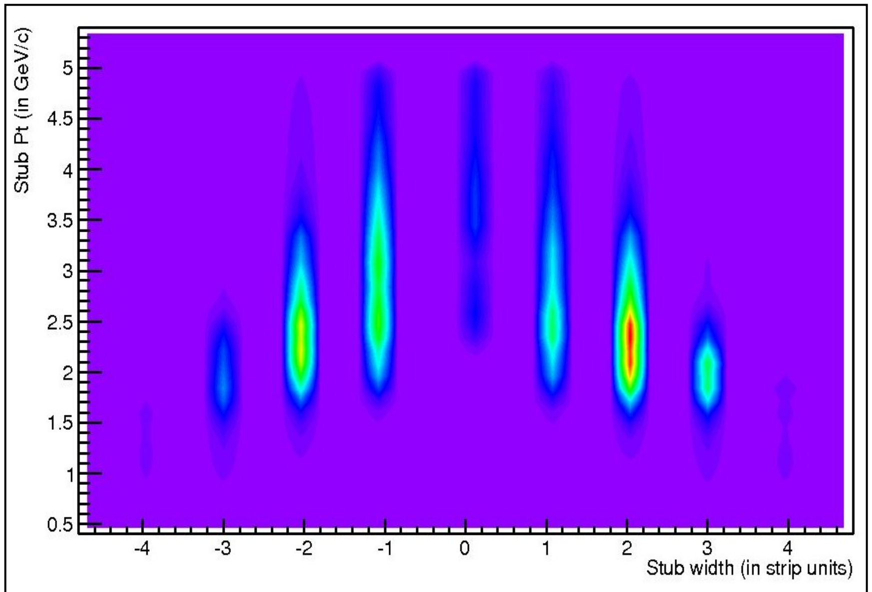
→ **CW and SW cuts can be applied online at the hardware level . Therefore a rough Pt selection can be applied before sending the info to L1 system.**

→ Relation between stub width and Pt of the particle:

→ In our track trigger project, the Pt threshold is currently set to 2 GeV/c



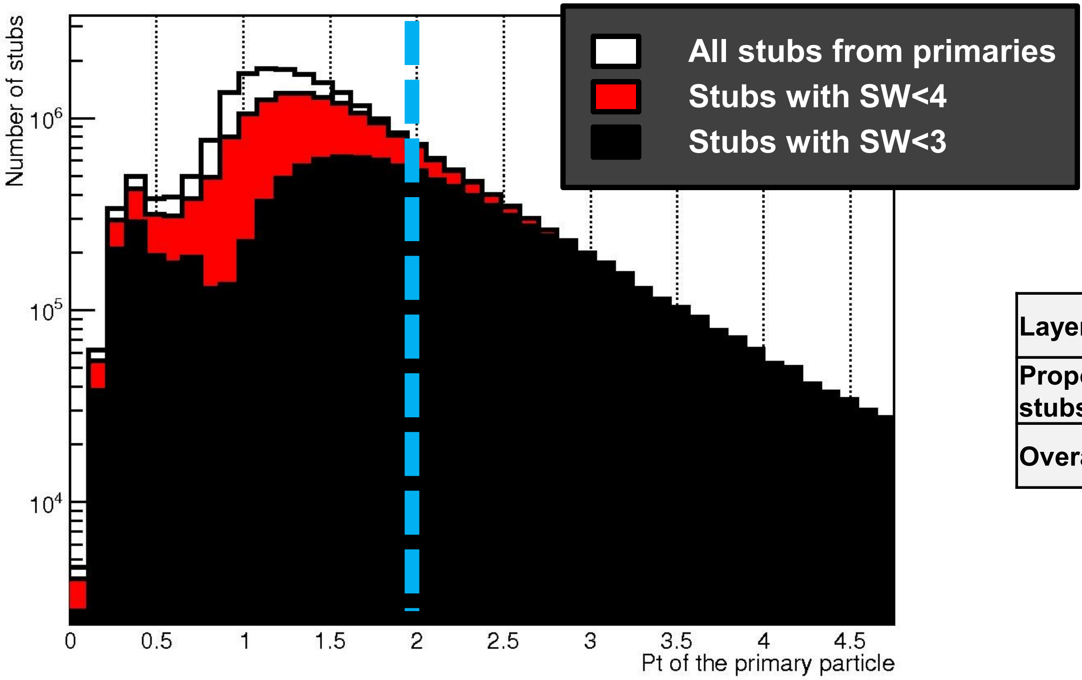
SW vs stub measured Pt for all stubs extracted from CMS PU events (PU200)



Same plot keeping only stubs coming from primary particles with Pt>2GeV/c

→ From these plots, one sees that an SW cut of 2/2.5 could significantly reduce the stub rate, with a relatively small effect on good data.

→ **Data loss estimation when cutting on stub width:**



Good data losses and raw rate with SW<3

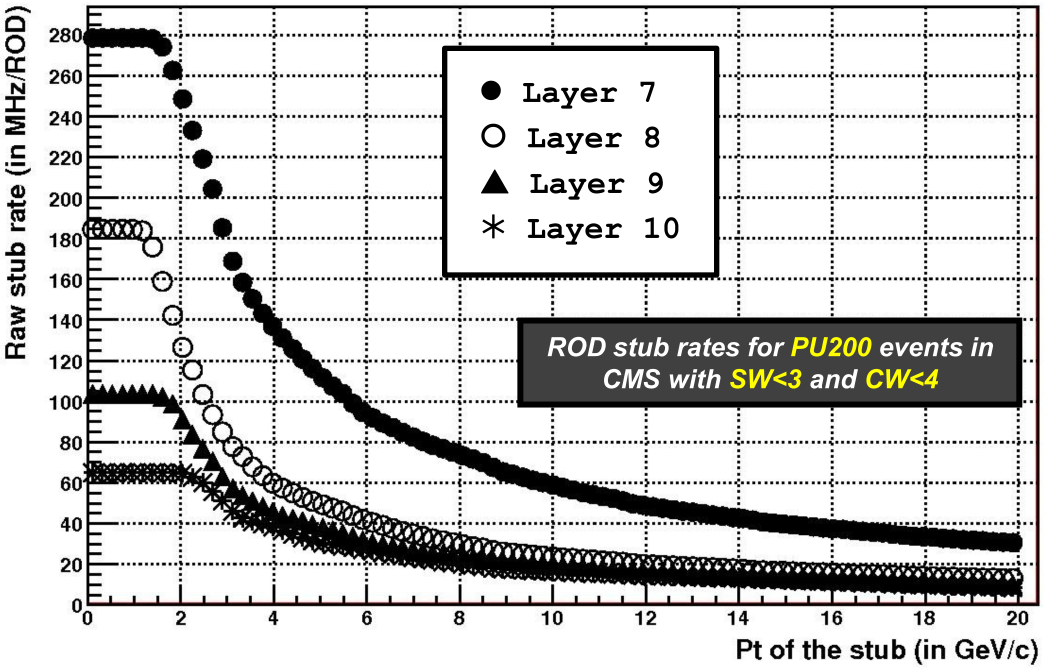
Layer	5	6	7	8	9	10
Proportion of lost stubs (in %)	7,5	6,8	19,9	2,5	7,8	21,9
Overall rate reduction	1,8	1,7	1,7	1,6	1,6	1,5

→ **The gain on raw stub rate is significant.**

→ **The loss is layer dependent (*SW increases with radius*). Cut values can be optimized for each layer.**

→ **We need to study how these cuts may affect the pattern matching process (*for the moment we don't apply strict cuts during stub formation process*).**

→ ROD stub rates using CW3 and SW2 cuts:



→ **For our study, we use stubs made with CW<7 and SW<5 (*conservative approach*).**

→ **The cuts are applied a posteriori for the moment.**

→ **Suppose we need 20 bits to transmit one stub to the AM board. The rates per RODs are:**

Layer	5	6	7	8	9	10
Max rate (in GHz/ROD)	32,3	17,7	9,4	6,0	3,3	2,0
Rate with stub cuts (in GHz/ROD)	17,6	10,4	5,6	3,7	2,1	1,3

→ **Using only the 4 last layers, 10GHz/ROD seems sufficient in all cases (*ie w or w/o stub cuts*).**

→ **Pattern matching: the procedure:**

1. Choose a tracker geometry.

2. **Depending on the constraints**, try to find the best configuration using ***clean*** MC events.

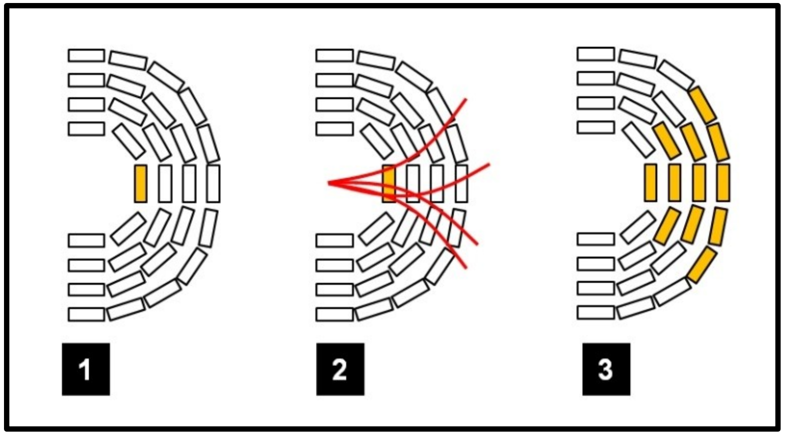
→**The best configuration is the most efficient set of pattern/sectors.**
→**The constraints are:**

- **The minimum pT of the track** you want to trigger ***(constrains the sector size)***
- **The maximum number of words** you could feed into the AM at L1 rate ***(constrains the sector size)***
- **The maximum number of patterns** you could store into one AM chip ***(constrains the pattern granularity)***
- **The maximum number of words per pattern** in order to ensure a correct fit ***(constrains the pattern granularity)***
- **The maximum number of AM chips** you can afford ***(constrains the sector size)***
- …

3. **The best configuration should be reasonably robust against pileup (*efficiency should be robust, not fake rate*)**, so step 2 has to be done also with heavy pileup MC samples.

4. If the results are not satisfying, one should restart at step 1

→ **Sector definition:**



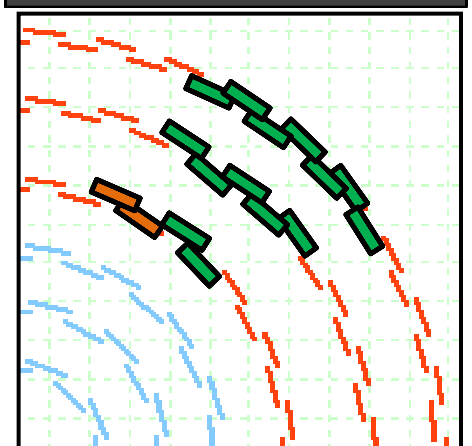
→ **We define sectors using ‘brute force’ method:**

1. **We choose a ladder in the innermost layer**
2. **We look where are going the 2GeV/c $\mu^{+/-}$ crossing this ladder**

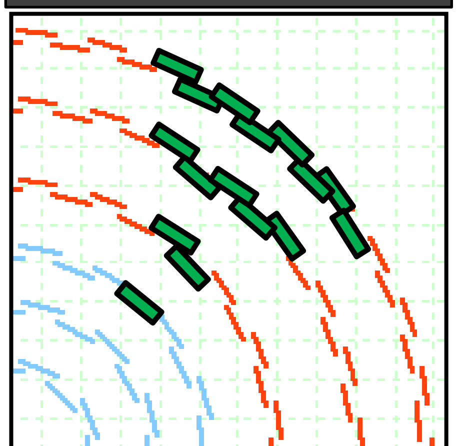
→ **No sector overlap in the innermost layers.**

→ **Of course, if necessary, these sectors can be divided into subsectors using independent pattern banks**

Sectors with 3 layers (*14/15 modules in r/ϕ*)



Sectors with 4 layers (*16 modules in r/ϕ*)



→ **We tested 2 different sector sizes (*3/4 layers*)**

→ **Using the two innermost layers seems challenging (*prohibitive rates*)**

→ Data rate in the AM with the defined sectors

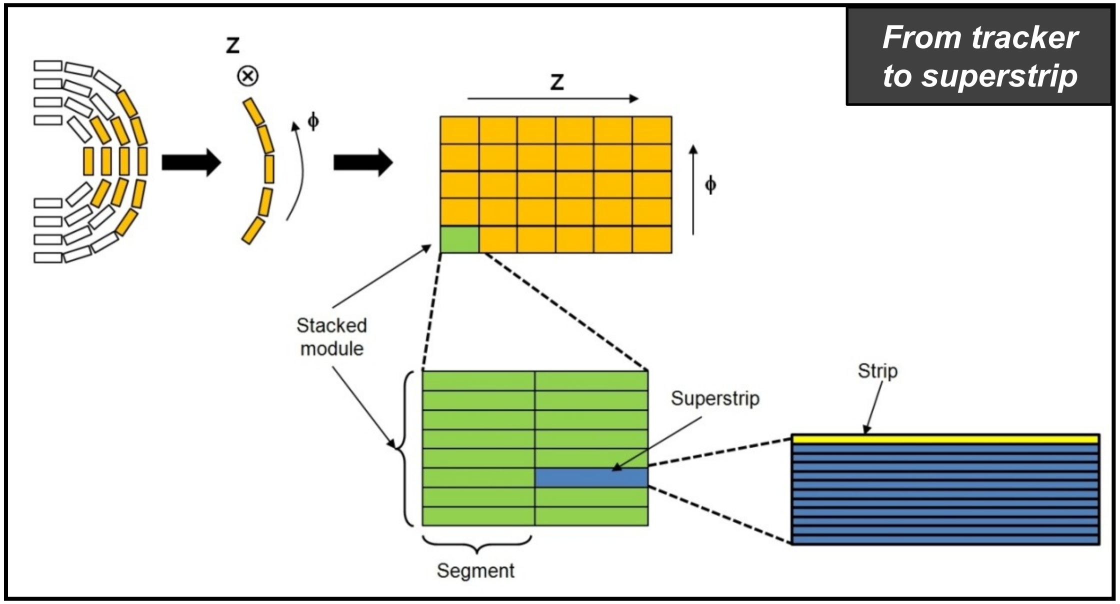
→ **Using ROD rates presented previously (*the ones with stub cuts*), one could extrapolate the input rates in the AM board input buses, for the different configuations**

Layer	5	6	7	8	9	10
Input rate for 3 layers (in GHz)				7,4	10,4	9,1
Input rate for 4 layers (in GHz)			5,6	7,4	10,4	10,4

→ **This can be reduced using smaller sectors, but it gives an idea of the numbers one should expect.**

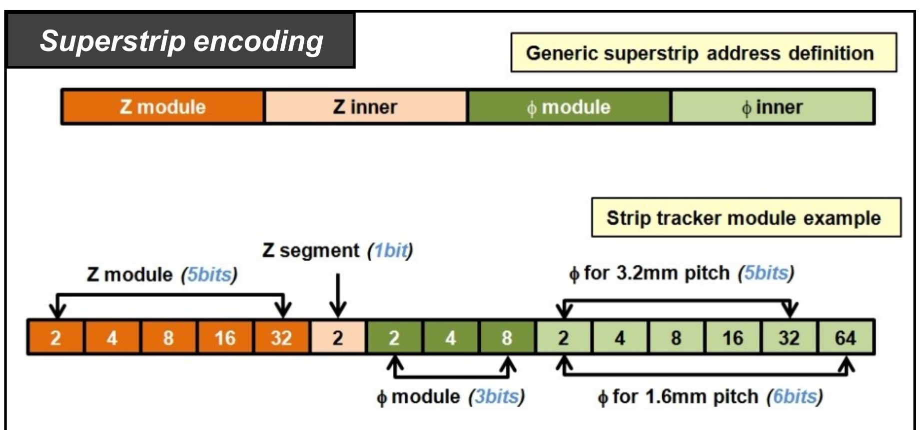
→ Is this realistic or not???

→ **Superstrip definition:**



→ **A superstrip is simply a bunch of strips in one module of the tracking detector.**

→ **The superstrip address is the info sent to the AM board. Is is coded on a certain number of bits, depending on the superstrip resolution.**



→ **The encoding is divided into 4 parts, giving module and intra-module SS position in Z and ϕ direction (*R is not necessary*)**

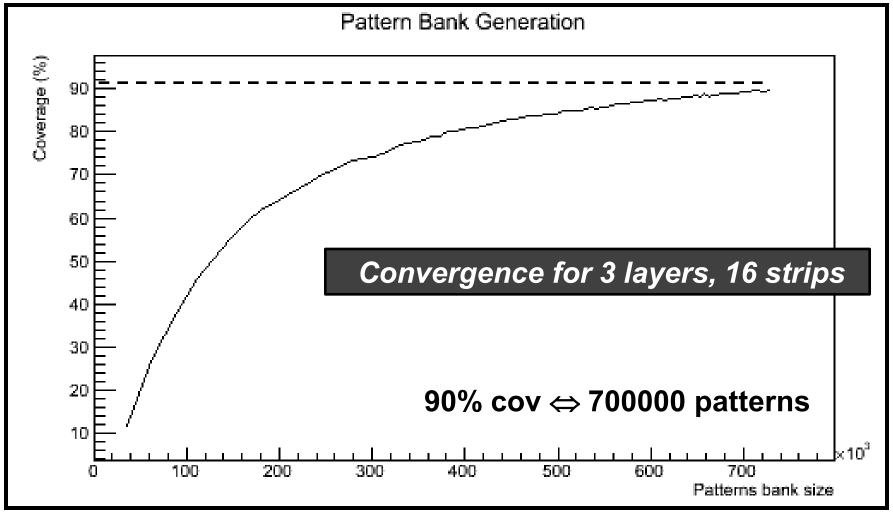
→ **We are not using pixel info yet, so our Z intra-module encoding is very basic for the moment.**

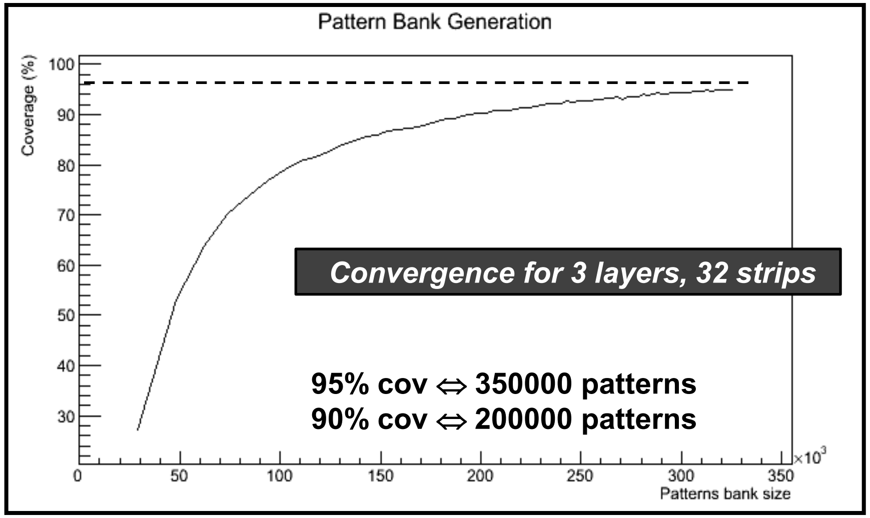
→ **The baseline:**

→ **The bank generation software process iteratively using a set 20M Pgun μ^+/μ^- with p_T bet. 2 and 100 GeV/c**

→ **The coverage is defined as the proportion of tracks in the sector matching a pattern in the bank.**

→ **The coverage you require is one of the parameters driving the size of the bank.**





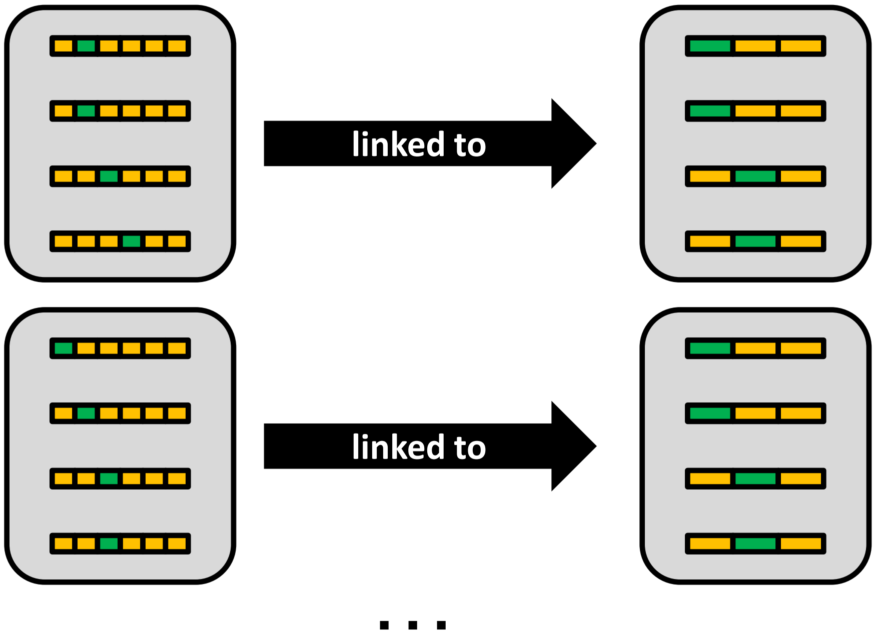
→ **The superstrip size (*see plots*) is significantly affecting the bank size.**

→ **With 32 strips, for an equivalent coverage, 3.5 times less patterns than with 16 strips are necessary**

→ **With 4 layers, one needs 700000 patterns to get a 90% coverage with 32 strips (*not enough stat with 16 strips*)**

→ Reducing bank size using variable resolution patterns:

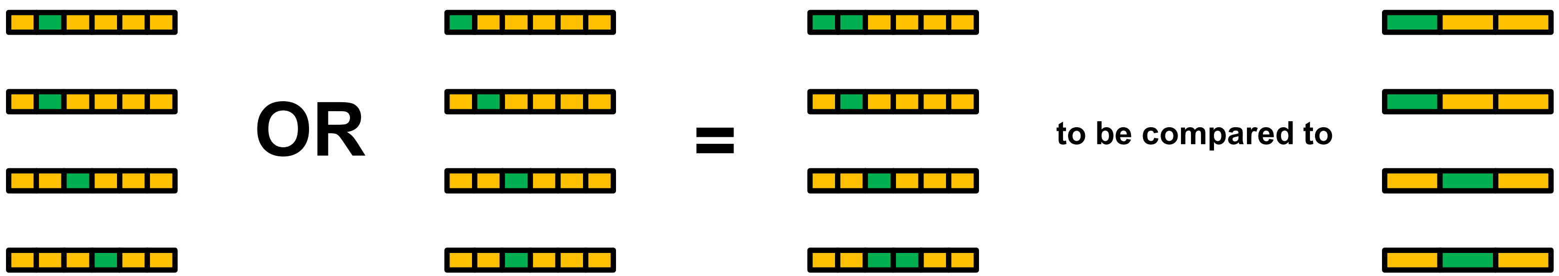
→ We adapted the technique described in ATL-UPGRADE-PROC-2011-004. During the bank generation, we associate the patterns to lower-res patterns:

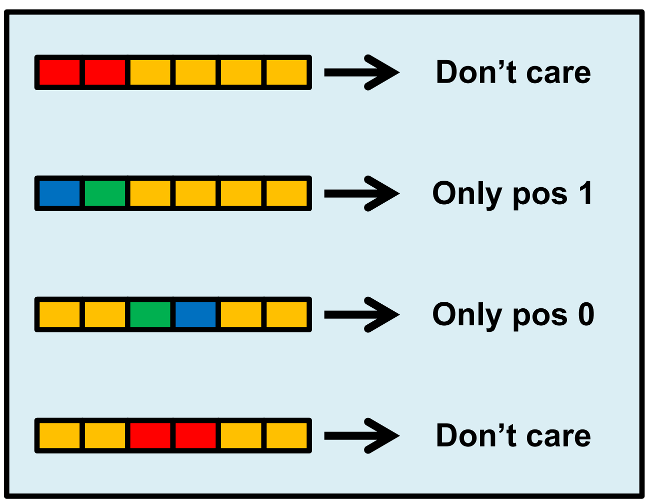


→ **Using this method, you just need to construct one high resolution bank (*this assume that you have enough statistic to do it…*). The low-res patterns and linkings are done on-the-fly.**

→ Reducing bank size using variable resolution patterns:

→ At the end, we loop over the low-res patterns found, and for each of them we retrieve all the corresponding high-res patterns. We **OR** them and compare the result to the parent lo-res pattern:



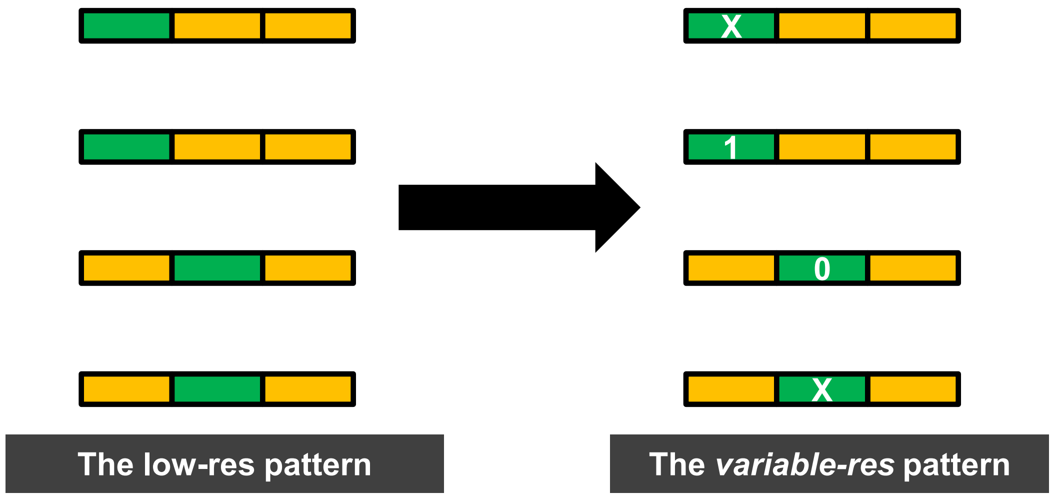


→ **If the two high-res SS are used**, this means that **we can't take advantage of the higher resolution for this pattern in this layer**. Therefore we will set a **don't care bit** (**X**) in the low-res pattern. Don't care because there is nothing to gain here…

→ On the other hand, if we see that **only one of the high-res SS is fired in the merging result**, then **we can use the higher resolution**. We will set a **position bit** (**0 or 1**) for this layer in the low-res pattern.

→ Reducing bank size using variable resolution patterns:

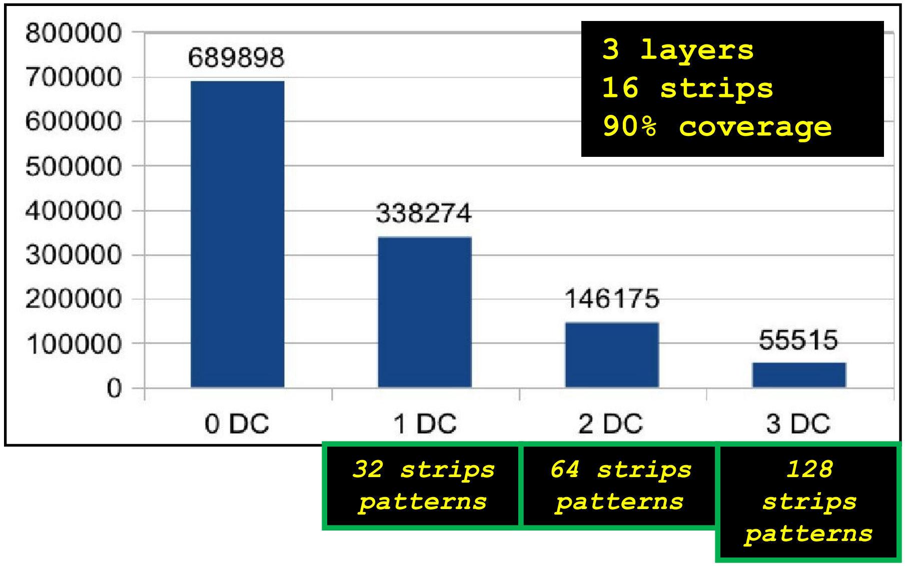
→ In the **pattern bank**, **you keep the low-res pattern**, and you just add **one ternary bit per layer** (*this is possible with ternary AM*). This bit has the three following states: **X (don't care), 0, and 1.**

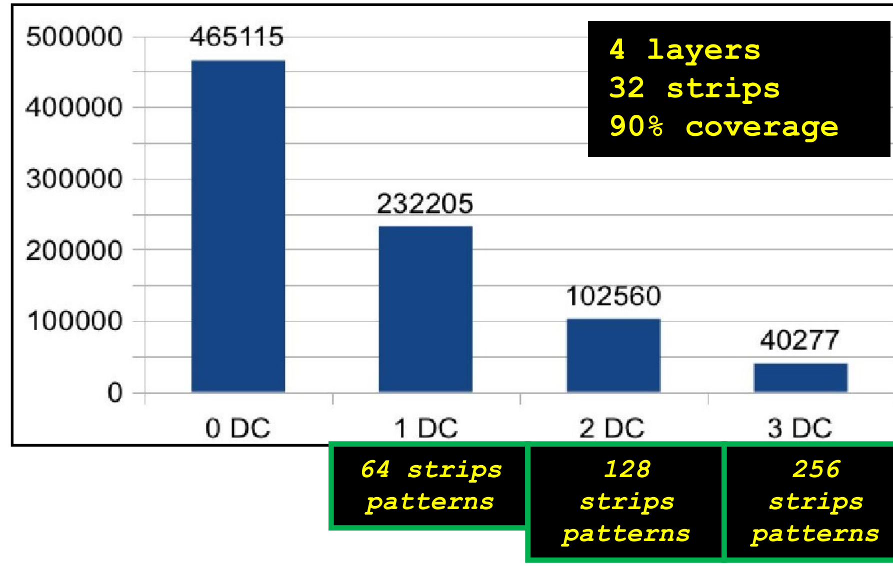


→ So basically **you have the information of two hi-res patterns with one low-res pattern**, you reduce the size of your bank but keep the sensitivity.

→ **You can add up to 3 DC bits in the current AM chip (*is it a fixed number??*), so we produced pattern banks with 1,2, or 3 DC bits, starting from the bank previously shown.**

→ **Effect of DC bits on the pattern bank size:**





→ **90% coverage stands for the 0DC bank, it gets higher by construction when one adds DC bits.**

→ **The reduction of the bank size is significant, but the low res patterns might be a bit large, in particular in the low res case. This could affect the fake rate.**

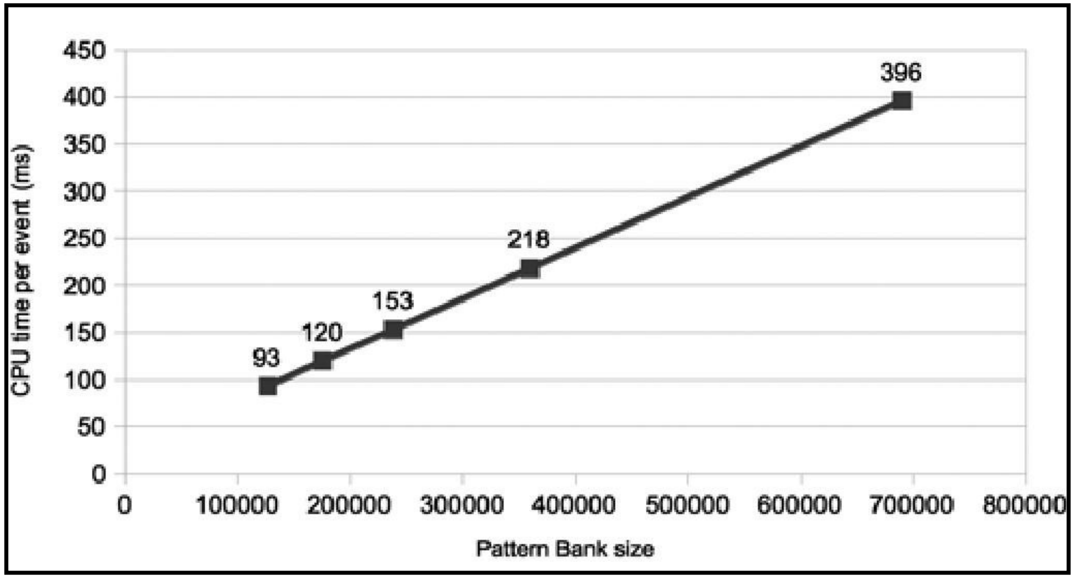
→ **One need to start from higher resolution patterns (*eg 4 layers with 16/8 strips*). Much larger samples have to be produced.**

→ Software emulation principle:

→ In order to test the efficiency of the generated bank, **a software emulation of the AM was written.**

→ Works as the bank generator (***ie fully flexible***). Takes in input the **pattern bank** and the **stubs of a given event**.

→ As in the AM, **the pattern ID is done in one single pass**. Output for each event is the list of patterns activated.



→ CPU consumption of the pattern matching w.r.t. the bank size

→ **The algorithm developed is linear w.r.t. the bank size.**

→ **Few definitions:**

Efficiency = # of primary tracks (*ie coming from IP*) contained in at least one activated pattern / # of primary tracks (*with at least one stub per layer*)

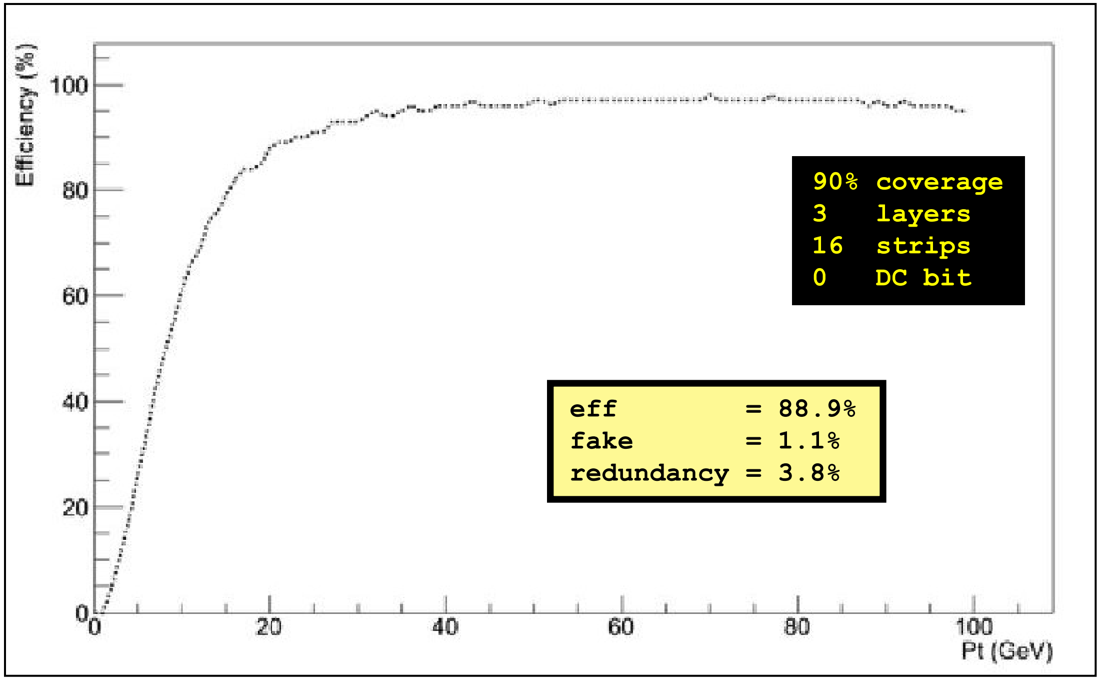
Fake rate = # of activated patterns containing NO primary track / # of activated patterns

Redundancy = # of primary tracks activating more than one pattern / # of primary tracks activating a pattern

→ **We want a good efficiency for particle with Pt>2GeV/c, with the lowest possible fake rate, and keep the redundancy a low as possible.**

→ **Results on mu+/mu-:**

→ **Baseline result for 3 layer sector, using the classic bank, requiring exactly 3 stubs on the pattern**



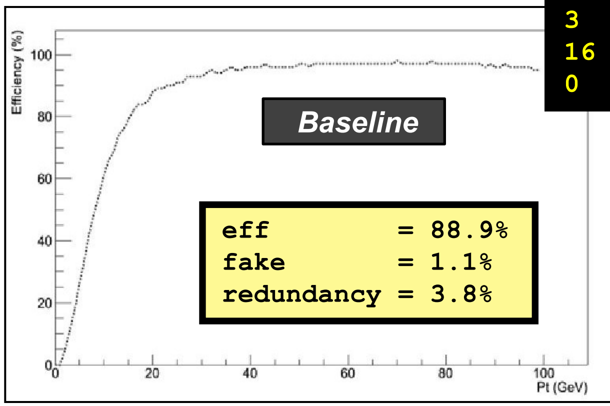
→ **Slow turn-on curve (*we're aiming for 2GeV/c*), but low fake rate and redundancy**

→ **The slow turn on comes from tracks with missing stubs, and from the fact that it's difficult to populate the bank at low Pt.**

→ **First point can be solved by adding missing stubs possibility, second point by adding DC bits.**

→ Effect of missing stubs, DC bits:

90% coverage
3 layers
16 strips
0 DC bit

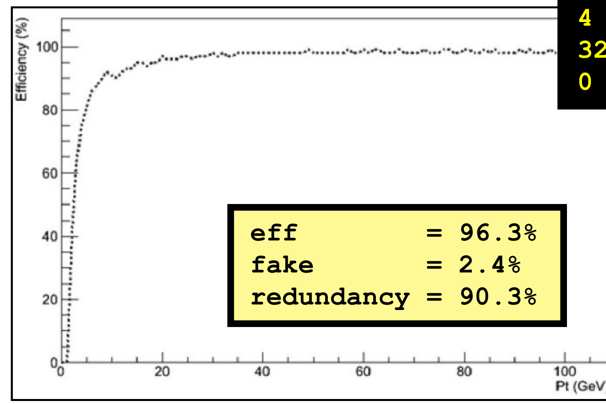


Baseline

eff = 88.9%
fake = 1.1%
redundancy = 3.8%

Go to 4 layers, accept 1 missing stub per pattern

90% coverage
4 layers
32 strips
0 DC bit

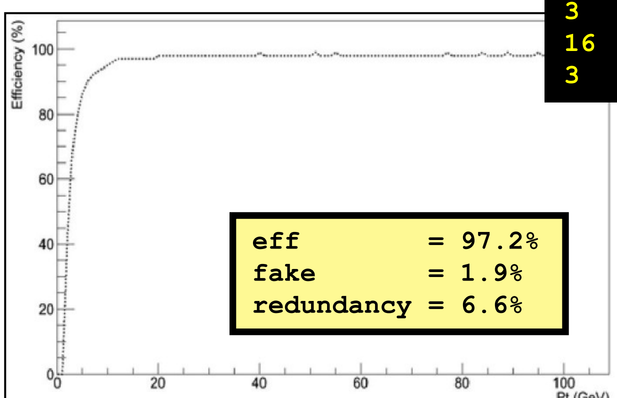


eff = 96.3%
fake = 2.4%
redundancy = 90.3%

Add DC bits

Add DC bits

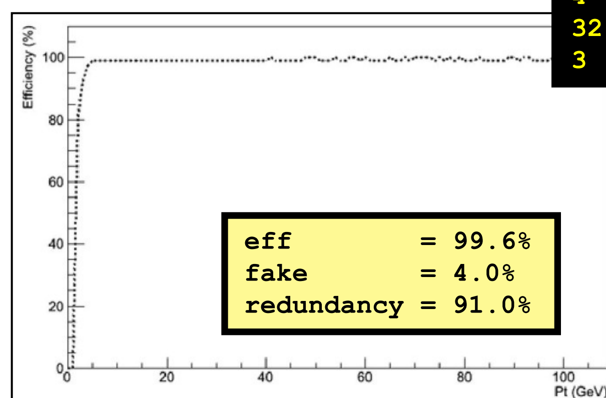
90% coverage
3 layers
16 strips
3 DC bits



eff = 97.2%
fake = 1.9%
redundancy = 6.6%

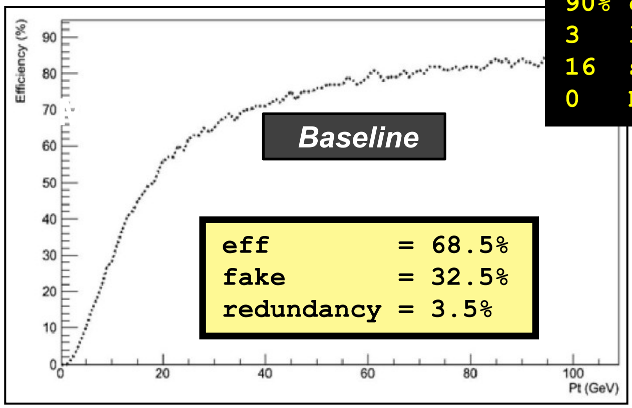
Go to 4 layers, accept 1 missing stub per pattern

90% coverage
4 layers
32 strips
3 DC bits



eff = 99.6%
fake = 4.0%
redundancy = 91.0%

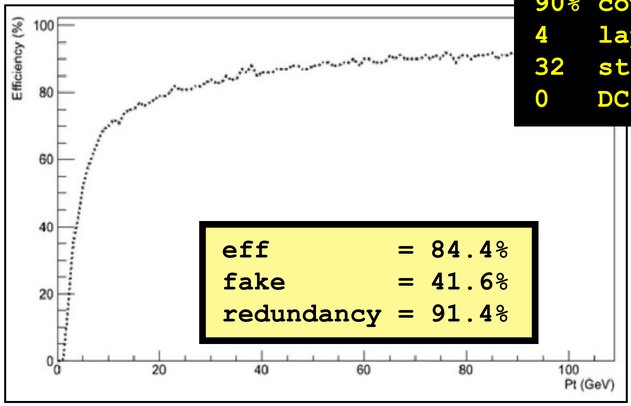
→ Result for electrons (*worst case for single particle*):




```
90%  coverage
3    layers
16   strips
0    DC bit
```

```
eff        = 68.5%
fake       = 32.5%
redundancy = 3.5%
```

Go to 4 layers, accept 1 missing stub per pattern

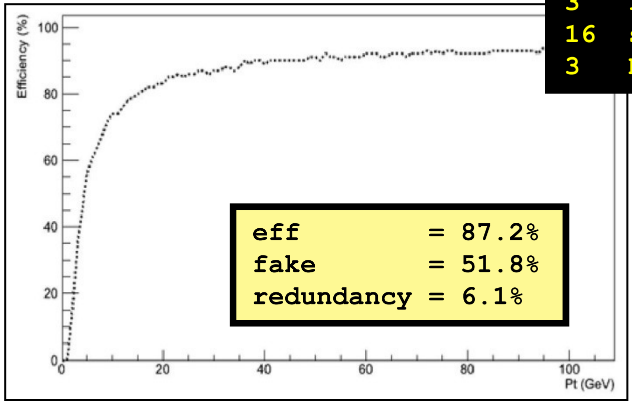



```
90%  coverage
4    layers
32   strips
0    DC bit
```

```
eff        = 84.4%
fake       = 41.6%
redundancy = 91.4%
```

Add DC bits

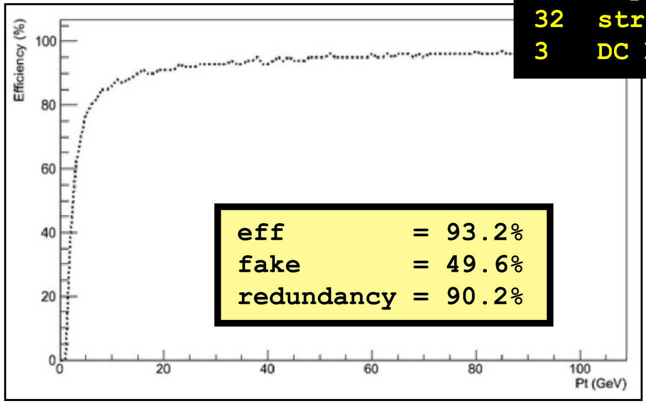
Add DC bits




```
90%  coverage
3    layers
16   strips
3    DC bits
```

```
eff        = 87.2%
fake       = 51.8%
redundancy = 6.1%
```

Go to 4 layers, accept 1 missing stub per pattern




```
90%  coverage
4    layers
32   strips
3    DC bits
```

```
eff        = 93.2%
fake       = 49.6%
redundancy = 90.2%
```

→ **Conclusion on simple events:**

→ **The addition of DC bits and missing stubs is significantly improving the turn on of efficiency curve.**

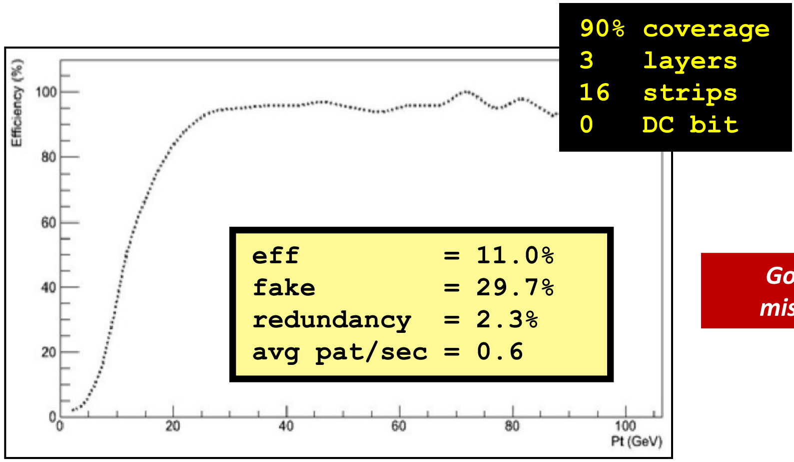
→ **For muons (*and also for pions*), we can be within the requirements (*Pt>2GeV/c with >90% efficiency*).**

→ **The convergence for electrons is also pretty good, but as expected a bit worse.**

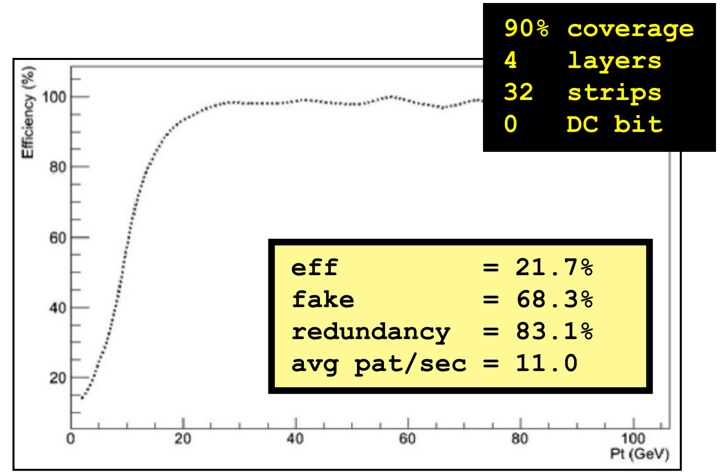
→ **The price to pay is a larger redundancy (*when adding DC bits*), and a larger fake rate (*when adding missing stubs*). Depending on the difficulty to sort this out at the trackfit stage, this could be or not a problem.**

→ **Next step is to look at heavy pile up events (*200PU*).**

→ Results with 200 PU events:

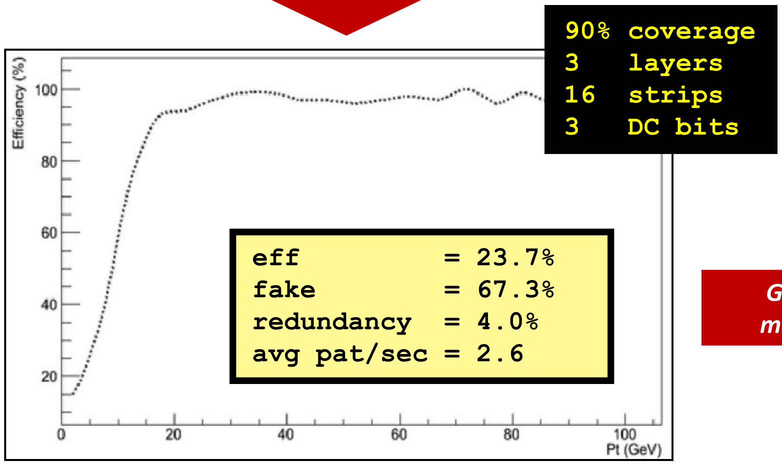


Go to 4 layers, accept 1 missing stub per pattern

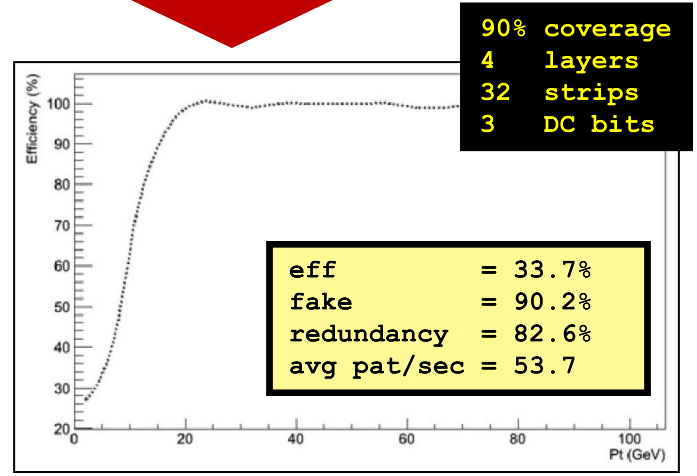


Add DC bits

Add DC bits



Go to 4 layers, accept 1 missing stub per pattern



→ **Rates of matched patterns with 200 PU events at 20MHz:**

	N DC bits	0	1	2	3
Sector with 3 layers	Total pattern rate (in MHz)	11,2	12,4	22,2	52,4
	Good pattern rate (in MHz)	7,9	8,1	10,5	17,1
Sector with 4 layers	Total pattern rate (in MHz)	219,8	214,2	428,6	1073,2
	Good pattern rate (in MHz)	69,7	57,3	70,0	105,2

→ **The fake rate is clearly larger in PU events (*not really a surprise*).**

→ **The efficiency looks small, but is mainly driven by poor efficiency on the huge amount of low Pt particles. At higher Pt the efficiency is close to 1.**

→ **We reach 90% at ~15GeV/c. This is far from 2GeV/c, but this is a very first look, there is plenty of room for improvement (*stub cuts, higher resolution pattern bank,…)***

→ **The 2GeV/c is not out of reach at all…**

→ We have set up a flexible software for studying AM-based tracking in CMS future tracker. This software is not depending on the geometry (*dealing with SS addresses*), and **provides bank generation** (*w or w/o DC bits*) and **pattern matching tools.**

→ Different parameters have been tested, along with different event samples:

- We start to get a better picture of the task, in particular **we have a rough idea of the data rates that the system will have to sustain** (*using stub rates from PU events*).
- **We see the effect of adding missing stubs and DC bits abilities.** This leads to promising results for single particles, but fake rates becomes dominant for high PU events. **But there is plenty of room for improvement here.**
- **The superstrip granularity is strongly affecting the pattern bank size.** Going from 16 to 32 strips doesn't affect the efficiency, but increase fake rate by 50% (*for PU*). One has to find a balance between precision and size of the system. **DC bits can provide that, but we still have to fully understand there impact.**
- Considering the stub rates on the four outermost layers**, including a fourth layer seems a good option,** in particular in view of the track fit which comes after the AM matching. OK, but we definitely need more events to generate the pattern bank.
- This fourth layer is a PS one. We should try to **find a way to add the pixel info without affecting too much the bank size.**