First introduction to YARR software for pixel testing

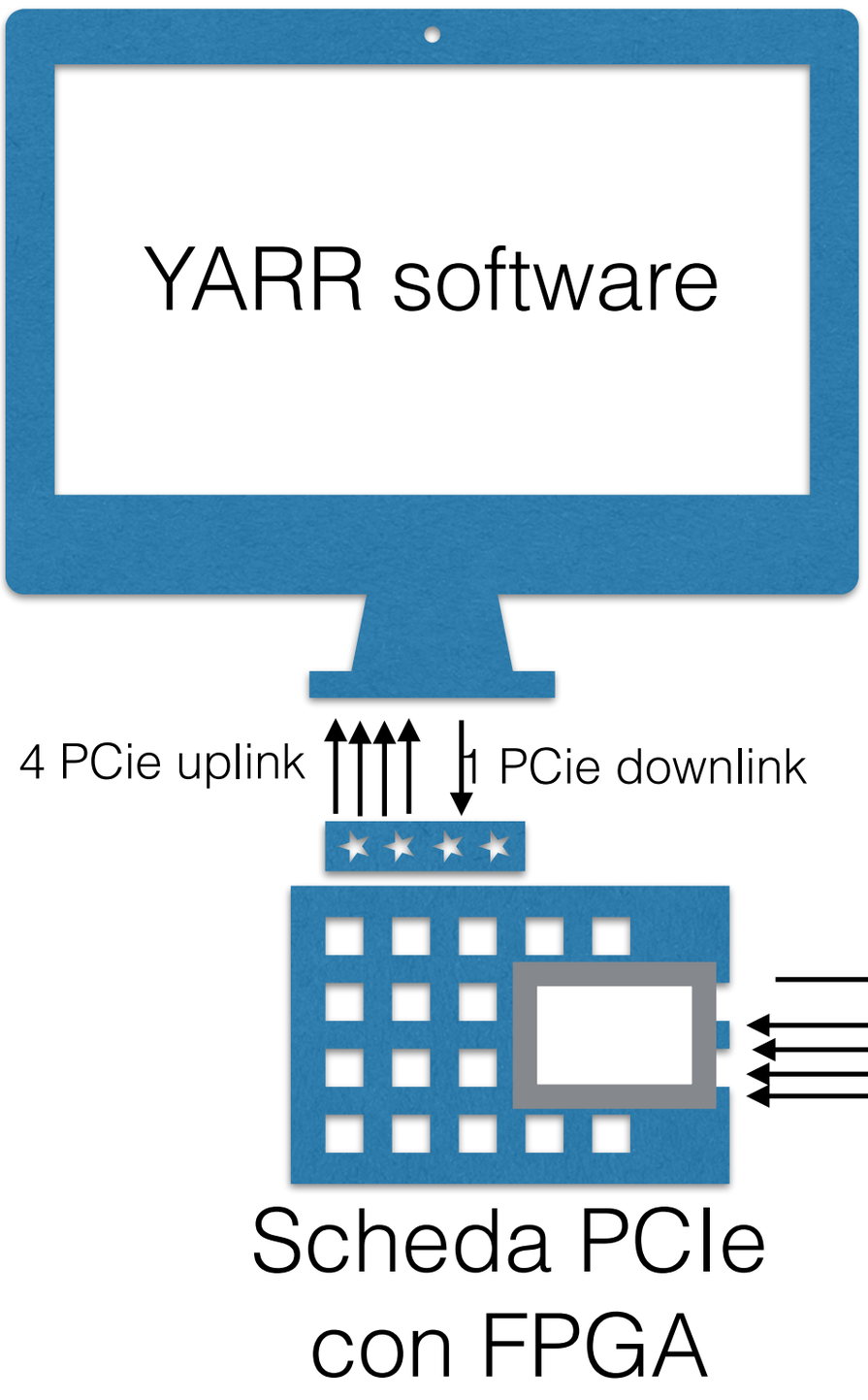
Gabriele Chiodini

INFN Lecce

https://indico.physics.lbl.gov/indico/event/856/contributions/3618/attachments/1912/2386/2019_05_18-yarr_nuthsell-theim.pdf

https://iopscience.iop.org/article/10.1088/1742-6596/898/3/032053/pdf

DAQ blocks



La readout board (Scheda PCIe con FPGA) svolge due compiti minimali:

- scrivere nella memoria del PC i dati di uscita del chip a pixel cosi' come arrivano dal chip (4 bytestream seriali)
- scrivere nei registri del chip a pixel i parametri di controllo

Tutto il processamento dei dati ed i parametri di controllo sono gestiti dal Software C++ detto YARR.

Pixel test e DAQ con YARR SW

I test e e/o calibrazione e/o tuning del chip a pixel sono sostanzialmente due:

1. Iniezione di carica nei canali di elettronica e rivelazione della loro risposta: test di charge injection
2. Rivelazione dei canali soprasoglia a seguito della presenza di radiazione ionizzante

In queste slide ci occuperemo solo di test di charge injection

Le procedure di test e/o calibrazione e/o tuning del chip a pixel mediante charge injection in generale vengono eseguite con SCANSIONI costituite da tre parti:

- iterazione (o loop) su parametro di controllo
- iterazione (o loop) su un pattern (injection mask) di pixel della matrice da abilitare per la iniezione e spegnendo gli altri (kill mask)
- Iterazione (o loop) N volte del trigger, che inietta la carica, seguito dal readout dati dei pixel sopra soglia (SPARSE READOUT).

I DAQ tradizionali eseguivano questi loop mescolando hardware e software .
Il DAQ basato sul software YARR esegue tutti questi loop via software.

Scansione e loop

Il vantaggio di questa implementazione è che una SCANSIONE è semplicemente una raccolta di LOOP che possono essere nidificati in qualsiasi ordine desiderato

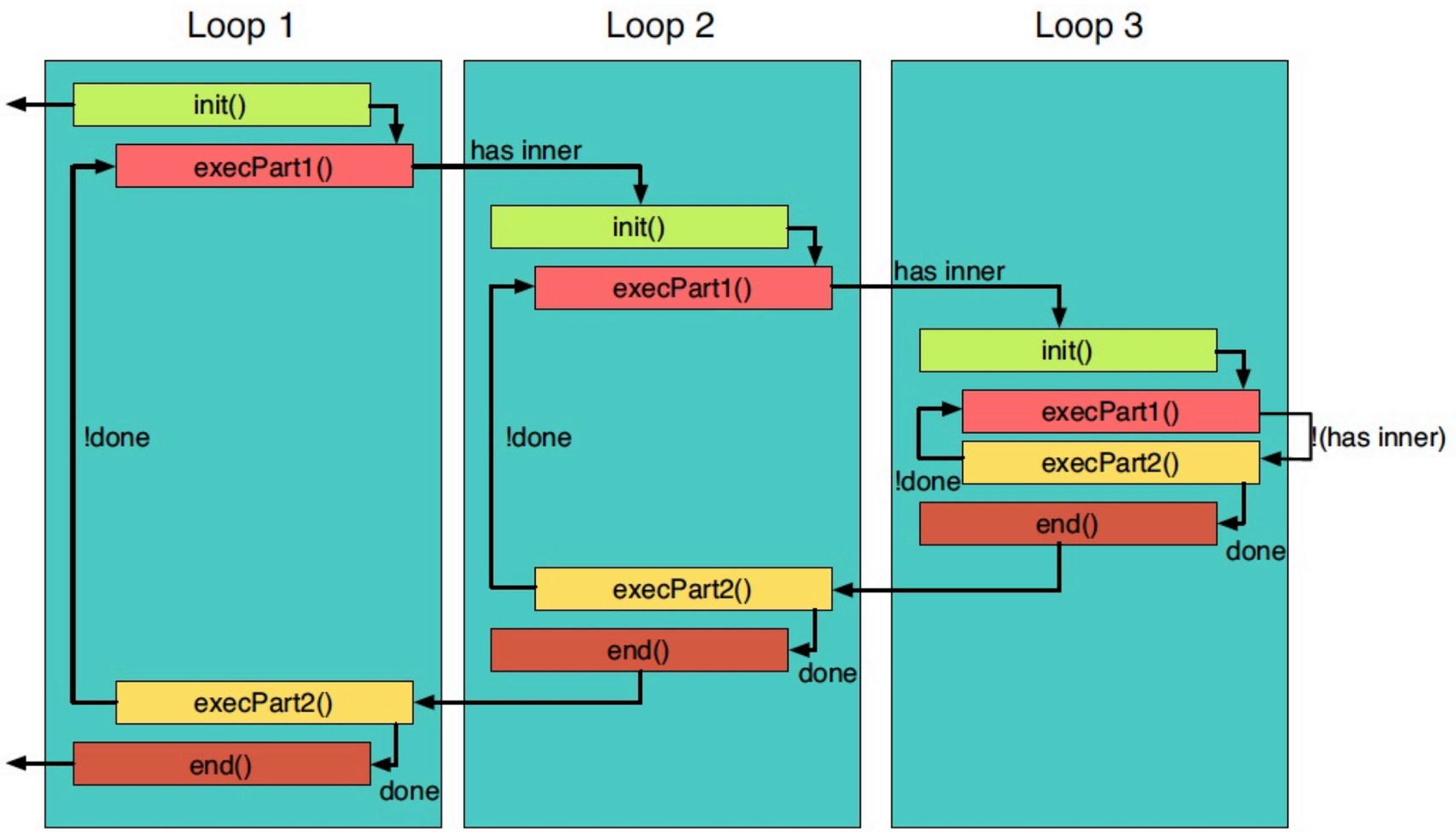
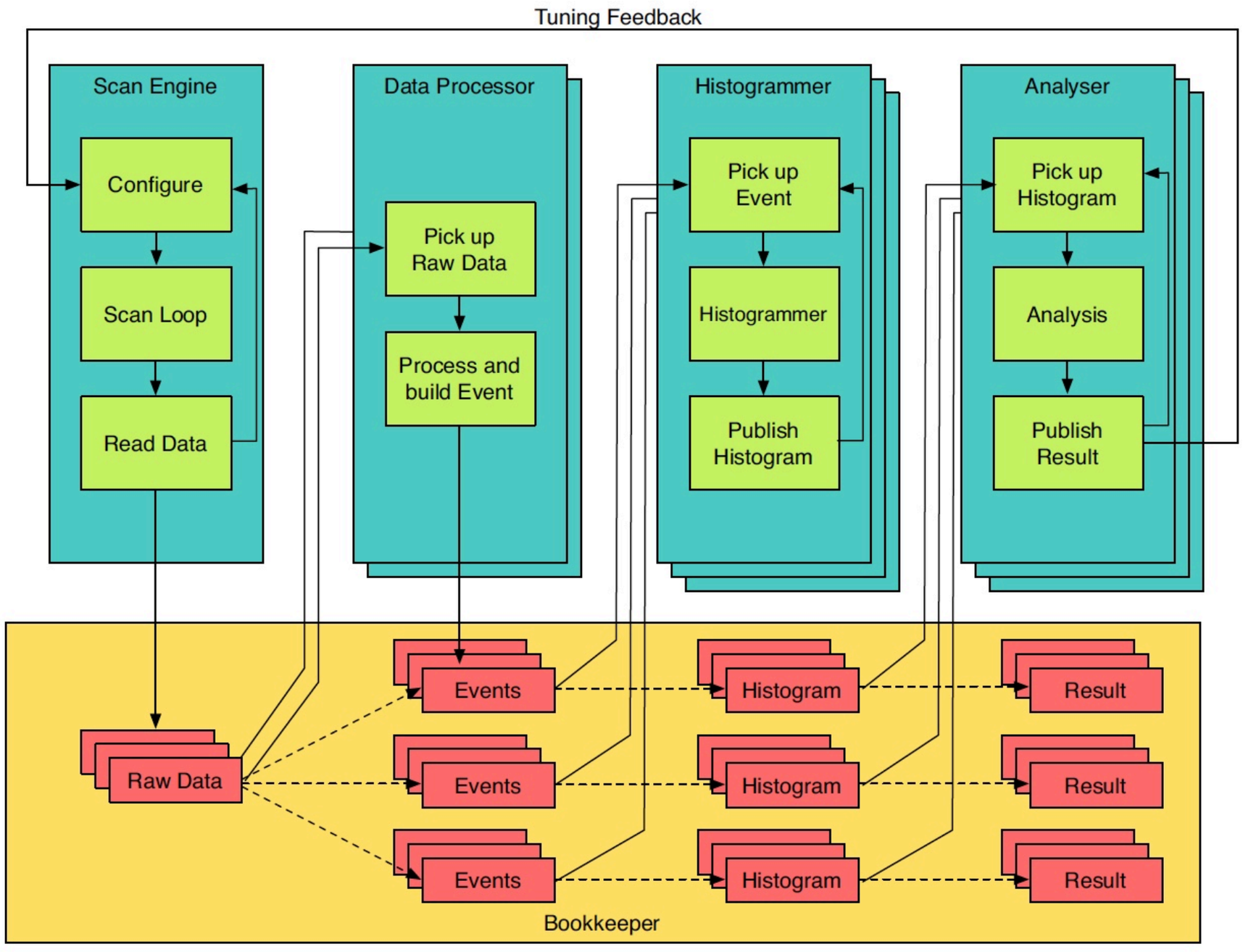


Figure 5: Dynamic nested loop structure used in YARR for the implementation of scans. [4]

Struttura software



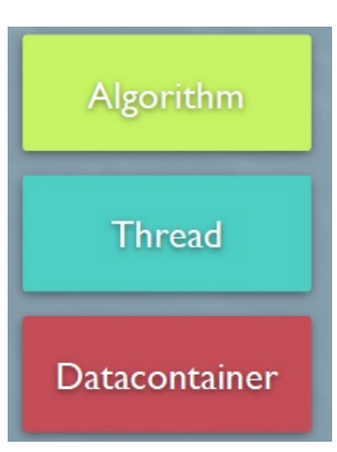
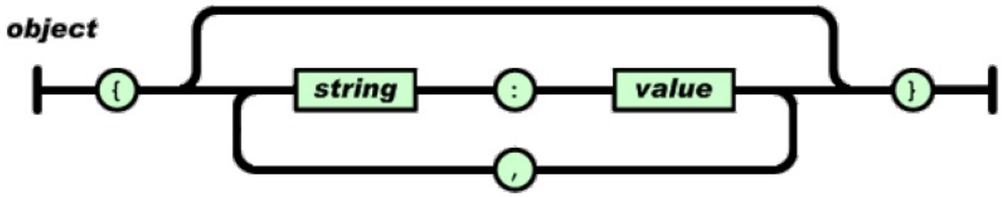


Figure 6: Diagram depicting the different stages of the data processing and showing the usage of multi-threading architectures. Each blue box is a different thread, except for the tuning feedback no inter-thread communication is needed, all the information a thread needs is contained in the data objects in form of metadata.[4]

File di configurazione: .JSON file

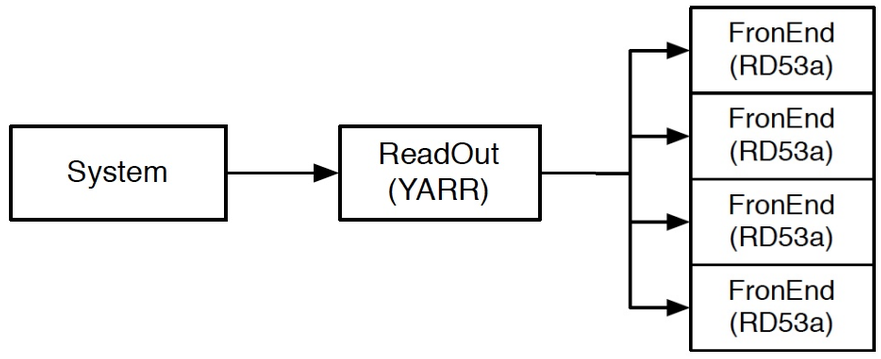



```
{
    "scan": {
        "analysis": {
            "1": {
                "algorithm":
"OccupancyAnalysis",
                "config": {
                    "createMask": true
                }
            },
            "n_count": 1
        },
        "histogrammer": {
            "0": {
                "algorithm": "OccupancyMap"
            },
            "1": {
                "algorithm": "TotMap"
            },
            "2": {
                "algorithm": "Tot2Map"
            },
            "3": {
                "algorithm": "L1Dist"
            },
            "4": {
                "algorithm": "HitDist"
            },
            "n_count": 5
        },
        "loops": {
            "0": {
                "config": {
                    "enable_lcap": true,
                    "enable_scap": true,
                    "mask": 4097,
                    "max": 16,
                    "min": 0,
                    "step": 1
                },
                "loopAction": "Fei4MaskLoop"
            },
            "1": {
                "config": {
                    "max": 4,
                    "min": 0,
                    "mode": 1,
                    "step": 1
                },
                "loopAction": "Fei4DcLoop"
            },
            "2": {
                "config": {
                    "count": 50,
                    "delay": 30,
                    "extTrigger": false,
                    "frequency": 1000.0,
                    "noInject": false,
                    "time": 0
                },
                "loopAction": "Fei4TriggerLoop"
            },
            "3": {
                "loopAction": "StdDataLoop"
            },
            "n_loops": 4
        },
        "name": "DigitalScan",
        "prescan": {
            "FE-I4B": {
                "GlobalConfig": {
                    "DigHitIn_Sel": 1,
                    "Trig_Count": 10,
                    "Trig_Lat": 220,
                    "Vthin_Coarse": 200
                }
            }
        }
    }
}
```


Esempio di JSON file di scan

Lo stato del software si basa su configurazioni di alto livello decise da JSON file:

-Connettività (quanti e quali chip di pixel sono connessi)



-Quale scansione fare (quale test fare):Digital scan(params…)

- Analog scan(params…)
- Threshold scan(params…)
- Threshold tuning(params…)
- ToT scan(params…)
- …

-Configurazione del data taking e.g.

- readout window (consecutiveL1A)
- latency