Cahier des charges d'un corrélateur pour NenuFAR-Imager: NARCOSYS V2

Cedric Viou

2018/09/24

$Rev:: 5528 $:

https://svn.obs-nancay.fr/svn/NenuFar/trunk/NRI/pre-studies/FPGA_correlator/CDC_noFPGA.md

1 Historique

Action	Auteur	Date
Cahier des charges	Philippe Zarka	2018/05/17
V0	Cedric Viou	2018/05/27
Relecture	Philippe Zarka	2018/06/21
Relecture	Laurent Denis	2018/06/22
Relecture (fringe stopping)	Julien Girard	2018/07/05
V1	Cedric Viou	2018/07/17
Supprime infos FPGA	Cedric Viou	2018/09/24
V2	Cedric Viou	2018/09/24

2 Description générale

L'instrument NenuFAR disposera de 96 Mini-Réseaux + 6 Mini-Réseaux-Distants. Néanmoins pour des considérations d'implémentation plus efficace des calculs, on traitera dans un premier temps seulement 90 Mini-Réseaux + 6 Mini-Réseaux-Distants pour rester proche d'une puissance entière de 2 ($96 = 3 \times 2^5$). Lors de la complétion de l'instrument, on upgradera éventuellement le corrélateur pour effectuer les 6 corrélations manquantes.

Parameter	Description	Value
N_{stat}	# transmitting stations	96, puis 102 max
$N_{MR\_per\_HPAPB}$	# of MR processed by a HPAPB	0 à 4
N_{band}	# sub-bands (coarse-grain channels)	512
N_{sel_band}	# sub-bands selected for correlation	384 max
N_{pol}	# polarizations	2
f_{stat}	Station samples frequency (Mega samples/s)	200
N_{ch}	# channels for frequency sub-band	64

Parameter	Description	Value
N_{taps}	# PFB FIR filter taps	8 ???
N_{dump}	# samples integrated at the correlation step	3052 @ 1 Hz (= $f_{stat}/(2N_{band}N_{ch})$

3 NenuFAR-Imager requirements

De chacune des 384 sous-bandes de 195.3125 kHz (=$f_{stat}/(2N_{band})$ = 200 MHz/1024), le corrélateur reçoit 96 x 2 (polarisations) flux de données complexes à 5.12 μsec de résolution temporelle (=195312.5 valeurs complexes / sec), des 90 Mini-Réseaux + 6 Mini-Réseaux-Distants, pour les 2 polarisations + le numéro de sub-array de chaque MR (éventuellement identique pour tous les MR si on considère 1 seul sub-array formé de tous les MR).

Le corrélateur doit effectuer en temps réel, à flux continu, les opérations suivantes:

- Sub-channelisation des 96×2 signaux dans N_{ch} canaux spectraux, avec $N_{ch} \in (1, 64)$, par FFT sur N_{ch} échantillons consécutifs, réduisant la largeur spectrale de chaque canal à 3.051 kHz et la résolution temporelle à 327.68 μsec. Le nombre d'échantillons n'est pas modifié, mais la dynamique des signaux doit augmenter si on veut rester conservatif.
- Phasage relatif des signaux des MR de chaque sub-array dans la direction (θ, ϕ) par application d'une rampe de phases ($2\pi f_c \delta t_i$ avec $i \in [1, nMR_sub-array]$, hypothèse bande étroite). **Pas sûr qu'on ait besoin... Coût et impact sur la synchro négligeable. Voir section *Compensation de retard et Fringe stopping***
- Calcul des cross- et auto-corrélations (produits complexes) par lattices 2x2 C_{ij} tel que

$$C_{ij} = \begin{bmatrix} V_{i1}V_{j1}^* & V_{i1}V_{j2}^* \\ V_{i2}V_{j1}^* & V_{i2}V_{j2}^* \end{bmatrix}$$

avec i et j, les numéro de MR tels que $i \in [1, 96+6], j \in [1, 96+6]$ et polar = 1 (NE), 2 (NW)

On note que $C_{ji} = C_{ij}^*$, On peut se limiter à calculer les C_{ij} pour $i \in [1, 96+6], j \in [1, i]$ pour réduire le coût de calcul d'un facteur 2.

- Moyenne des produits calculés sur 0.25 à 1 sec => écriture des résultats.

Le traitement en flux continu est un peu difficile ici pour le PFB à cause de l'implémentation qui sera choisie (besoin de mémoriser l'état du filtre PFB pour chacune des ssb de chaque MRs pour repartir correctement après un dump). Il est acceptable d'éliminer systématiquement les N_{taps} premières FFT avant de corréler/accumuler chaque dump? Cela correspond à une perte de donnée de $N_{taps} \times N_{ch}$ échantillons par dump, soient typiquement ~512 sur 195312/s, ce qui est négligeable pour l'application.

Il est souhaité que le nombre de voies à corréler, ainsi que le nombre de sous-bandes par unité de corrélation soit configuré par des paramêtres génériques dans le code (statique après compilation) afin d'ajuster la puissance de calcul du corrélateur à l'état de déploiement de NenuFAR.

Le controle-commande du corrélateur disposera des valeurs des fréquences centrales fc et la direction de pointage analogique (θ, ϕ) du sub-array afin de calculer les paramètres de *fringe stopping*.

4 Structure générale du calcul

De 0 (sic) à 4 MRs peuvent être connectés sur chaque cartes HPAPB (MRs du cœur et MRDs). Les formes d'onde ADC transitent vers un PFB+FFT, puis des modules sont configurés pour effectuer le phasage, puis la somme numérique de ces signaux à travers toutes les cartes du système pour construire des beams numériques. (voir Figure 1)

En parallèle de ça, en sortie de la FFT, nous disposons des 512 sous-bandes calculées sur chaque paires d'entrées ADC. L'instanciation de plusieurs modules CEP_V2 va nous permettre de sélectionner une partie des sous-bandes de tout ou partie des MRs connectés aux HPAPB. Chaque carte pourra ainsi créer un paquet réseau (UDP) qui contiendra l'information d'une partie des antennes pour une portion de la bande de 75 MHz (soit 384 sous-bandes). Les cartes enverront vers la même Unité de Corrélation (CU) tous les paquets construits pour la même portion de bande. De cette façon, un CU, après agrégation en mémoire de toutes les trames, disposera, pour un jeu de sous-bandes donnée, de l'information de tous les MRs. En agrégeant les résultats de calcul de tous les CU (qui auront tous traité une portion de bande différente), on reconstruit la bande de l'instrument dans son ensemble.

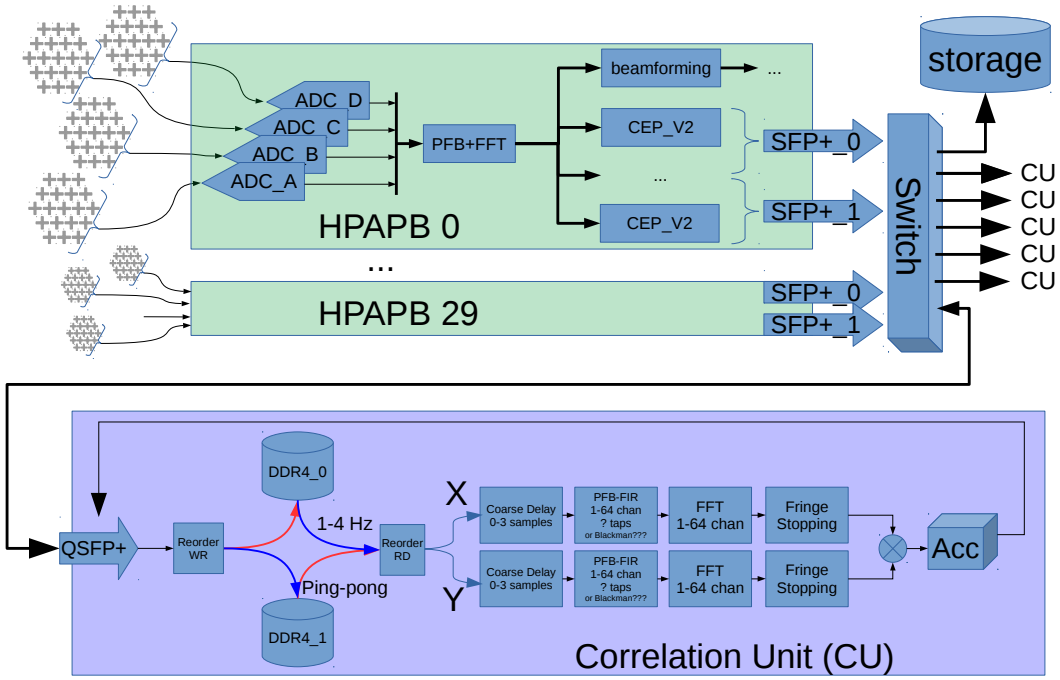

Figure 1: Structure générale du calcul

Nous allons maintenant nous concentrer à détailler chaque sous-éléments de ce calcul.

5 Description et dimensionnement des sous-éléments

Mis à part l'aspect très calculatoire de l'application (5 TFLOPS pour la corrélation seule), le point dur est la réorganisation des données pour alimenter efficacement ce corrélateur. Nous détaillerons cette partie avec attention dans la section *Échange mémoire DDR4*.

5.1 Définition de CEP_V2

Ce module ne fait pas à proprement parler du corrélateur. Mais ses instances contribuent à aider les serveurs de calcul à réordonner les flux pour nourrir efficacement les GPU de COBALT, ou les unités de corrélation de cette étude.

Il prend en entrée l'intégralité des sous-bandes des 4 MRs d'une carte HPAPB via le port *sdo_arr_arr* du module *processing*.

Sa configuration via une interface AVM permettra de:

- sélectionner les ssb à exporter
- sélectionner les MRs à exporter
- configurer un gain commun pour les ssb et les MRs
- configurer les limites de la plage temporelle d'émission admise
- configurer des champs d'entête facilitant la transposition des données dans le corrélateur:
  - MAC/IP/UDP
  - NOF_SSB, NOF_BLOCK, ...
  - F_c, (θ, ϕ), ... pas souhaité, mais OK si nécessaire...

Le module produira au "bon" moment des trames UDP d'un MTU < 9000 qui seront routées vers le bon CU. Les infos contenues dans l'entête devront permettre au CU de transposer correctement la payload pour alimenter son corrélateur.

L'entête contiendra du datage pour permettre aux CU de démarrer le traitement en synchronisme sur une date donnée et pour permettre de supprimer les paquets trop anciens.

TODO:

- profondeur FIFO
- Formatage de l'entête
- nombre d'instances requise par le corrélateur
- ressources par instances
- ressources totales

Il sera compatible avec l'envoi de beamlets (somme phasée des sous-bandes de plusieurs MRs) au format BHR.

5.2 Type et format des données entrantes

Une carte HPAPB dispose des données de toute la bande de quelques antennes, tandis qu'une unité de calcul du corrélateur doit disposer des données de toutes les antennes mais pour une bande de fréquence restreinte seulement.

Les HPAPB disposeront du framer CEP_V2 qui construira des paquets jumbo-frame contenant les sous-bandes complexes d'une partie de la bande à destination d'un IP/UDP configurable. Une carte maintiendra en FIFO assez de données pour construire ces paquets puis les enverra. Le nombre de FIFO à implémenter dépendra du nombre d'unité de calcul de corrélation à employer et ce nombre dépendra de la quantité de sous-bandes qui pourra être traitées par une unité.

L'instant d'envoi des données par rapport au PPS sera aussi programmable afin de ne pas saturer un des ports réseau du switch à cause de la confluence sur celui-ci de multiples flux de cartes différentes.

Les paquets réseau contiendront un échantillon complexe (x2) polarisé (x2) sur 8 bits, soit 4x8 bits pour une succession de sous-bandes à un instant donné, puis une nouvelle succession d'échantillons pour l'instant suivant ($+5.12\mu$s). Le nombre d'échantillons successifs $N_{sample\_per\_bloc}$ sera choisi pour maximiser la taille des paquets, mais pourra être fixé à la première puissance entière de 2 pour faciliter l'adressage. Le contenu d'un paquet aura la forme d'un hypercube de dimensions $(N_{sample\_per\_bloc}, N_{sel\_band_i}, N_{MR\_per\_HPAPB}, N_{pol}, N_{cpx})$ en row-major order.

Le débit par sous-bande pour les données de toutes les antennes est:

$R_{bit} = N_{stat}N_{cpx}N_{pol}N_{bits}F_e/(2N_{band})$

avec $N_{stat} = 102, N_{cpx} = 2, N_{pol} = 2, N_{bits} = 8, F_e = 200e6, N_{band} = 512$

Soit 76 Mo/s par sous-bande.

En considérant que les interfaces réseau utilisées pour le corrélateur seront des QSFP+ (40GbE) (bien qu'il soit encore possible d'utiliser du QSFP28-100GbE), on ne pourra transmettre sur une

interface qu'un grand maximum de 64 sous-bandes ($64 \times 76MB/s \times 8bits = 39Gbps$), soit au moins 6 unités de corrélation, donc 6 instances minimum de CEP_V2 dans les HPAPB.

Ce débit moyen est constitués de paquets provenant de :

- 24 HPAPB connectées à 4 MRs maximum chacune
- 6 HPAPB connectées à 1 MR chacune

On pourra choisir dans CEP_V2 d'obtenir de trames de même taille pour les 2 types de flux, ou d'obtenir des trames contenant le même nombre d'échantillons temporel, mais pour un nombre d'antenne différent.

5.3 Reorder

Les données produites par les instances de CEP_V2 ont partiellement été transposée (Antennes <-> sous-bandes) par le switch. Mais une dernière étape est encore nécessaire pour mettre les données en forme de manière optimale en mémoire DDR pour optimiser leur relecture multiple pour nourrir les corrélateurs.

L'accès aux DDR est plus intense en relecture qu'en écriture (voir plus loin). On s'efforcera donc d'optimiser l'agencement mémoire pour faciliter la lecture (Reorder RD) au détriment de l'écriture (Reorder WR) qui s'effectuera de façon plus fragmentée.

5.4 PFB et FFT

Le PFB utilisé pour rechanneliser les sous-bandes peut se contenter de 8 taps + FFT 64 canaux. Le filtre prototype choisit est une fenêtre de Kaiser.

5.5 Compensation de retard (Coarse Delay) et Fringe stopping

Afin de limiter la décorrélation, il est nécessaire de corriger des retards géométriques de l'instrument sur les signaux des différentes stations. (Perley 2014) et (Thompson, Moran, and Swenson 2017, sec. 7.7 et 9.7.1)

Le temps de vol du signal au dessus de NenuFAR-Imager pour une source à l'horizon vaut $\frac{4 \text{ km}}{c} = 13\mu$s, soit moins de 3 échantillons dans une sous-bande de 195 kHz.

Cette correction sera appliquée via un paramètre positif (0, 1, 2, 3) configurable station par station qui indiquera à quel offset mémoire DDR4 commencera la lecture des données.

Cette partie demande un peu de logique de contrôle, mais les ressources sont mineures.

Il reste à corriger la partie fractionnaire au cours de l'accumulation si elle évolue trop rapidement: $\nu_f = u\omega_e cos\delta$, avec $u = B/\lambda$ et $\omega_e = 2\pi/86164.1 = 7.3 \times 10^{-5}$ rad/s.

Soit un fringe rate maximum de $\frac{3000}{3} 7.3 \times 10^{-5} = 0.062$ Hz @ 85 MHz et 0.007 Hz @ 10MHz pour NenuFAR.

Et $\frac{1500000}{1.2} 7.3 \times 10^{-5} = 91$Hz pour LOFAR (HBA).

Le besoin de faire du fringe stopping pour NenuFAR si le dump rate vaut 1-4 Hz est discutable.

Il subsistera une rampe de phase dans les données, mais à mon avis, ce sont des phases relatives qui n'ont pas à être corrigée en temps réel pendant la corrélation et qui peuvent être soustraites après corrélation.

Conclusion:

- OK pour un coarse delay compensation.
- Besoin de fringe stopping pour COBALT dans le réseau LOFAR.

- Pas besoin forcément besoin de fringe stopping pour NenuFAR.

5.6 Type et taille des opérandes

La problématique se pose pour une implémentation virgule fixe en FPGA. Le calcul en float64 sur les MPPA facilitent cet aspect.

5.7 Échange mémoire DDR4

5.7.1 Solution naïve:

Dans une version antérieure de l'étude, nous avions choisi de rapatrier pour chaque sous-bande, un échantillon de chacune des 102 antennes, d'effectuer l'autocorrélation pour produire une matrice complexe 102x102, puis d'accumuler cette matrice aux matrice précédemment calculées. Le pseudo code est donné ici:

```
for (ch=0; ch<nrChannels; ch++)
  for (time=0; time < integrationTime; time++)
    for (station2=0; station2<nrStations; station2++)
      for (station1=0; station1<=station2; station1++)
        for (pol1 = 0; pol1 < nrPolarizations; pol1++)
          for (pol2 = 0; pol2 < nrPolarizations; pol2++) {
            baseline = computeBaseline(station1, station2);
            FROM_RAM_STA_1 = samples[ch][station1][time][pol1]
            FROM_RAM_STA_2 = samples[ch][station2][time][pol2]
            FROM_RAM_ACC = correlation[baseline][ch][pol1][pol2]
            correlation[baseline][ch][pol1][pol2] = FROM_RAM_ACC + ( FROM_RAM_STA_1 * ~FROM_RAM_ST
          }
```

La taille de ces matrices impliquait des miss de cache fréquents et les accès multiples nécessaires à l'accumulation (lecture + accumulation + réécriture) produisait des débits bien trop excessifs pour être réalistes.

Le descriptif est tout de même réalisé ici.

5.7.1.1 Données à relire en mémoire

On charge en mémoire $N_{stat} \times N_{pol}$ échantillons complexes, on calcule une matrice de covariance "instantanée" de taille $N_{stat}N_{pol} \times N_{stat}N_{pol}$ que l'on accumule $N_{integrationTime}$ fois par seconde.

Le débit par sous-bande vaut donc:

$Débit = N_{stat}N_{pol}N_{integrationTime}sample_{Size}$

Soit 75 Mo/s/ssb pour notre cas ($sample_{Size} = N_{cpx} \times 1$ octet).

5.7.1.2 Ressources pour la corrélation

Pour chaque matrice de covariance "instantanée" calculées précédemment, on lit la mémoire d'accumulation, on met à jour le contenu, puis on réécrit en mémoire externe. Le volume à déplacer correspond à la diagonale inférieur de la matrice de covariance, soit:

$CovMat_{size} = \frac{N_{stat}(N_{stat}+1)}{2}N_{pol}^2 N_{cpx} \times \text{sizeof(int40)}$

Soit 145 ko par sous-bande, et 54 Mo pour la bande totale de l'instrument. Le double buffer nécessaire pour permettre l'export de ces données nécessite donc au moins 110 Mo de stockage qui seraient à implémenter en mémoire externe au FPGA.

Le débit en lecture ET en écriture vers cette mémoire vaut donc:

$D\acute{e}bit = CovMat_{size} N_{integrationTime}$

Soit 31 Go/s/ssb de bande passante en full duplex, ce qui est déjà supérieur à la bande passante théorique d'une DDR4 1200MHz (19 Go/s). Donc cette méthode n'est pas implémentable.

La sub-channelisation N_{ch} ne modifie pas les débits (il y a N_{ch} plus de canaux, mais N_{ch} moins d'échantillons temporels), mais augmente la taille mémoire d'une facteur N_{ch}.

5.7.1.3 Données à exporter pour stockage

Il faut exporter d'un coup la matrice accumulée car la matrice accumulée complète n'est disponible qu'à la fin de la période de dump. Si on veux exporter le cube de corrélation avant le début de la prochaine accumulation, il faut être capable de transmettre $N_{ch} \times 54Mo$ en moins de 5.12 µs, soit un débit pic irréaliste de 10 To/s.

On peut lisser le flux en utilisant un double buffer, soit 2 fois plus de mémoire.

5.7.2 Solution à implémenter

Après bibliographie, il s'avère que l'approche naïve précédente n'est utilisée que pour des corrélateurs disposant d'un nombre réduit d'antennes. Quand le nombre d'antenne augmente (van Nieuwpoort and Romein 2017; Hargreaves and Verkouter 2010), le choix est fait de stocker en mémoire les données brutes, puis pour chaque sous-bande, d'extraire les formes d'onde d'une paire d'antennes pour toute la durée enregistrée, effectuer la sub-channelization sur ces 2 flux continus et de cross-corréler/intégrer en mémoire locale avant de stocker le résultat en RAM externe (ou de transférer le résultat vers un serveur de stockage accessible à travers le réseau).

L'inconvénient ici est de devoir relire plusieurs fois les données brutes de certaines voies, mais cela reste moins coûteux que de lire et réécrire $N_{integrationTime}$ fois par seconde la mémoire d'accumulation.

Le pseudo-code (sans la sub-channelization) est présenté ici:

```
for (ch=0; ch<nrChannels; ch++)
  for (station2=0; station2<nrStations; station2++)
    for (station1=0; station1<=station2; station1++)
      for (pol1 = 0; pol1 < nrPolarizations; pol1++)
        for (pol2 = 0; pol2 < nrPolarizations; pol2++) {
          complex float sum = 0 + i*0;
          for (time=0; time < integrationTime; time++) {
            FROM_RAM_STA_1 = samples[ch][station1][time][pol1]
            FROM_RAM_STA_2 = samples[ch][station2][time][pol2]
            sum += FROM_RAM_STA_1 * ~FROM_RAM_STA_2;
          }
          baseline = computeBaseline(station1, station2);
          correlation[baseline][ch][pol1][pol2] = sum;
        }
```

5.7.2.1 Sans parallélisation d'accès aux données Stations

5.7.2.1.1 Données à relire en mémoire

Le pire cas se produit quand les paires de stations sont lues une par une (on n'envisagera pas le cas où les polarisations seraient aussi lues une par une). Dans le cas d'un réseau de 6 stations $(0, 1, 2, 3, 4, 5)$, on doit lire/corréler/accumuler/dumper les couples $[(0,0), (1,0), (1,1), (2,0), (2,1), (2,2), \cdots, (5,4), (5,5)]$ (voir Figure 2).

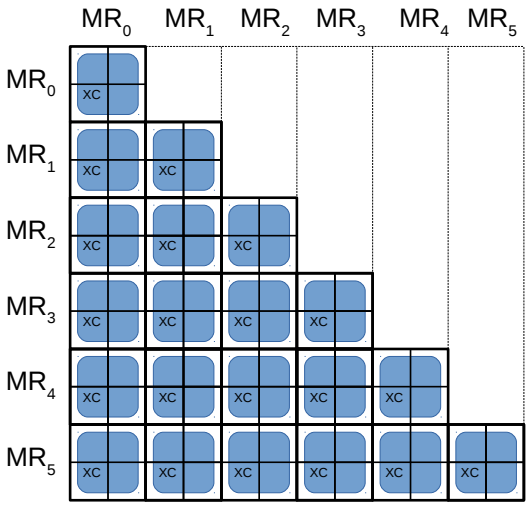

Figure 2: corrélation

Le nombre de lecture vaut donc $2 \times \frac{N_{stat}(N_{stat}+1)}{2} = N_{stat}(N_{stat}+1)$, soit un débit de :

$2 \times \frac{N_{stat}(N_{stat}+1)}{2}(\frac{f_{stat}}{N_{band}} N_{pol} N_{cpx} \times \text{sizeof(int8)}) = 6.8$ Go/s/ssb

La paire de flux à extraire de la mémoire a pour dimension $2 \times (t, N_{\|band}, N_{\|stat}, N_{pol}, N_{cpx})$ en row-major order avec $N_{\|band}$, le nombre de sous-bande qu'un CU pourra traiter en parallèle et $N_{\|stat} = 1, N_{pol} = 2, N_{cpx} = 2$.

5.7.2.1.2 Données à exporter pour stockage

Il suffit de transmettre au fil de l'eau le résultats de l'accumulation des corrélations successives. Le transfert est plus lissé et le volume à déplacer correspond toujours à la même diagonale inférieur de la matrice de covariance, soit:

$CovMat_{size} = \frac{N_{stat}(N_{stat}+1)}{2} N_{pol}^2 \times \text{sizeof(complex64)}$

Soit 145 ko/ssb et 54 Mo pour la bande visée lors de chaque dump. Avec une période de dump de 1 Hz, cela mène à 54 Mo/s, et 216 Mo/s pour 4 Hz.

5.7.2.2 Avec parallélisation d'accès aux données Stations

5.7.2.2.1 Données à relire en mémoire

On peut réduire le nombre de lectures redondantes si l'on est capable de corréler/accumuler en parallèle ($N_{\|stat}$) plusieurs produits de stations. Il y a donc $N_{\|stat}$ fois moins de lignes (elles-même $N_{\|stat}$ fois plus courtes) à lire qui calculeront chacune des trémis de taille $N_{\|stat} \times N_{\|stat}$. (Voir Figure 3)

Le nombre de calcul à lancer vaut donc $\frac{\frac{N_{stat}}{N_{\|stat}}\left(\frac{N_{stat}}{N_{\|stat}}+1\right)}{2}$. Et à chaque fois, il faut lire simultanément les données de $2 \times N_{\|stat}$ stations, donc le nombre total de lecture vaut $N_{stat}\left(\frac{N_{stat}}{N_{\|stat}}+1\right)$

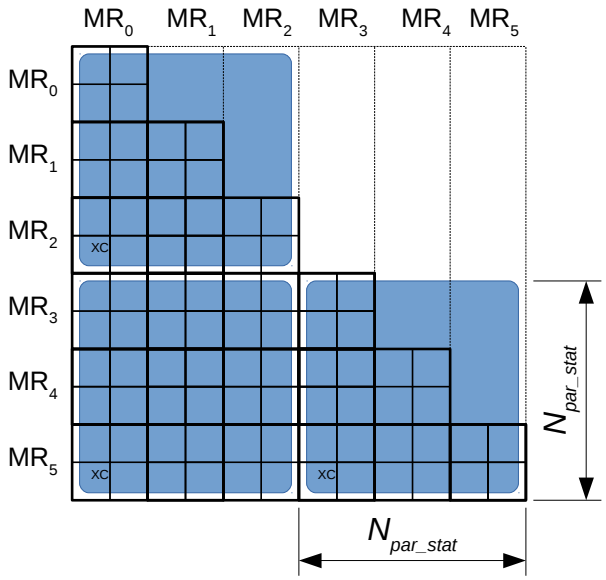

Figure 3: corrélation

La réduction de débit par rapport à une lecture paire par paire vaut $\frac{N_{stat}(N_{stat}+1)}{N_{stat}\left(\frac{N_{stat}}{N_{\|stat}}+1\right)} = \frac{N_{stat}+1}{\frac{N_{stat}}{N_{\|stat}}+1}$.

Le débit pour relire et traiter les données des 96 stations vaut:

- 6.8/ 7.4 = 0.91 Go/s/ssb pour $N_{\|stat} = 8$
- 6.8/13.8 = 0.49 Go/s/ssb pour $N_{\|stat} = 16$

La paire de flux à extraire de la mémoire a pour dimension $2 \times (t, N_{\|band}, N_{\|stat}, N_{pol}, N_{cpx})$ en row-major order avec $N_{\|band}$, le nombre de sous-bande qu'un CU pourra traiter en parallèle et $N_{pol} = 2, N_{cpx} = 2$.

5.7.2.2.2 Données à exporter pour stockage

La parallélisation rajoute un peu de calcul inutile au niveau des trémis situés sur la diagonale et un peu de complexité si on ne souhaite pas exporter les résultats.

Si on décide d'exporter les trémies brutes, en incluant les infos redondantes, le débit augmente mais reste sensiblement le même:

- 58.5 Mo/s pour un dump rate de 1 Hz et 234 Mo/s pour 4 Hz pour $N_{\|stat} = 8$
- 63 Mo/s pour un dump rate de 1 Hz et 252 Mo/s pour 4 Hz pour $N_{\|stat} = 16$

6 Tests

Les données à traiter, fortement bruitées, sont peu utilisables pour valider facilement l'intégrité du calcul. On prévoit d'implémenter dès le début du développement des générateurs de trames.

6.1 Test du modèle

La partie sensible du calcul est le réordering des données en DDR4 qui conditionne la synchronisation des données en entrée de la corrélation/accumulation.

On pourra ignorer/by-passer, ou ne pas implémenter dans un premier temps, les modules de Coarse Delay, PFB, FFT et fringe stopping. Ils apportent de la latence, utilisent des ressources, mais leurs comportements très prédictifs n'affecteront pas la synchronisation.

Un générateur de trames placé en amont du réordering WR simulera l'arrivée de paquet sur l'interface QSFP+, et permettra de valider le réordering DDR4:

- directement en sortie du réordering RD (on peut prédire le contenu du flux à ce niveau),
- et après corrélation/accumulation (on peut prédire le contenu de l'accumulateur).

6.2 Test de l'implémentation

Un générateur de trames configurable sera ajouté aux instances du module CEP_V2 afin de tester l'intégralité de la chaîne HPAPB –> switch –> CU –> switch –> Stockage.

Références

Hargreaves, Jonathan, and Harro Verkouter. 2010. "EVN Correlator Design." Version 1.0. JIVE.

Perley, Rick. 2014. "Fundamentals of Radio Interferometry Ii." *14th Synthesis Imaging Workshop*. Socorro, New Mexico. https://science.nrao.edu/science/meetings/2014/14th-synthesis-imaging-workshop/archive/SISS14Advanced.pdf.

Thompson, A. R., J. M. Moran, and G. W. Swenson Jr. 2017. *Interferometry and Synthesis in Radio Astronomy, 3rd Edition*. doi:10.1007/978-3-319-44431-4.

van Nieuwpoort, R. V., and J. W. Romein. 2017. "Correlating Radio Astronomy Signals with Many-Core Hardware." *ArXiv E-Prints*, February.