# Sistemas Distribuidos
# Departamento de Ingeniería en Informática
# LAB: Kafka - Spark Streaming

## 1 Objetivo

El objetivo de este laboratorio es diseñar e implementar una aplicación distribuida usando MPI. La aplicación a construir consiste del proceso llamado *gridding*, que se utiliza en síntesis de imágenes interferométricas.

## 2 Interferometría

### 2.1 El observatotio ALMA

Las antenas del observatorio ALMA se ubican en el llano de Chajnantor a 4800 metros sobre el nivel del mar, en el Desierto de Atacama. Estas antenas miden señales de radio frecuencia en longitudes de onda milimétricas y submilimétricas. La señal capturada por las antenas, entonces, no es la luz visible emitida por el objeto, sino la radiación que esa luz produce al interactuar con el gas y polvo que se encuentra alrededor del objeto. Luego, es necesario transformar esta señal de RF a una señal de intensidad. Este proceso se llama síntesis de imágenes.

Para ser más específicos, la señal capturada por las antenas corresponde a mediciones en el plano de Fourier. La Figura 1 muestra una típica distribución de posiciones de muestreo en el plano $uv$ (plano Fourier) de una cierto objeto. Cada punto representa una medición, llamada **visibilidad**. Aunque no es relevante para este lab, diremos que cada par de antenas produce una visibilidad, y a medida que pasa el tiempo y gira la tierra, el punto se mueve en el plano $uv$.

Como se puede observar, los puntos no cubren completamente el plano $uv$. Más aún, el muestreo es irregular, en el sentido que no están regularmente espaciados en ambas direcciones.

### 2.2 Conceptos básicos de Fourier

Sea $I(x, y)$ una imagen bidimensional de $N \times N$ píxeles, donde cada pixel representa el nivel de intensidad en la posición $(x, y)$. Luego, la transformada discreta de Fourier de esta imagen es:

$$V(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I(x, y) \exp(-2\pi j(ux + vy)/N) \tag{1}$$

y su transformada inversa

$$I(x, y) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} V(u, v) \exp(+2\pi j(ux + vy)/N) \tag{2}$$

ALMA mide $V(u, v)$ , pero la imagen que deseamos es $I(x, y)$. Si tuvieramos el plano $uv$ completo, es decir los valores $V(u, v)$ para todos los puntos $uv$ y éstos correspondiesen a una grilla regular,

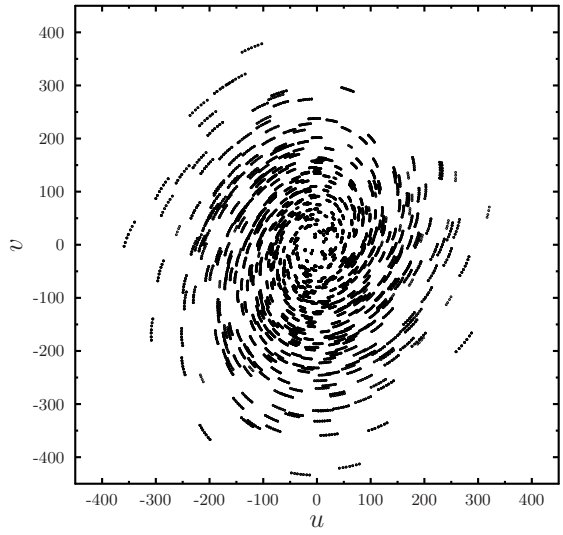


Figure 1: Típico muestreo del plano $uv$ por ALMA.

entonces la síntesis de imagen sería trivial, pues bastaría aplicar la transformada inversa a los datos $V(u, v)$ y obtendríamos $I(x, y)$.

La Figura 2 muestra una plano $uv$ grideado y su correspondiente imagen de intensidad. Note que la regularidad de muestreo también significa que tanto $I$ como $V$ pueden ser almacenados en matrices cuadradas. Tanto la matriz $V(u, v)$ como $I(x, y)$ son matrices complejas, es decir cada punto posee un componente real y otro imaginario, y lo que se visualiza es la parte real de este valor. Las imágenes fueron creadas con Python con las siguientes instrucciones:

```
import numpy as np
import matplotlib.pyplot as plt

# Suponga que el array complejo grid contiene los datos griddeados
image = np.fft.ifftshift(np.fft.fft2(np.fft.fftshift(grid)))

fig, ax = plt.subplots((1,2))

ax[0].imshow(np.abs(grid))
ax[1].imshow(image.real, origin="lower")
```

Listing 1: Ejemplo en Python

Asumiendo que los datos fueron grideados en un un numpy array, en donde cada valor en la grilla $uv$ es un número complejo. Se aplica la transformada inversa de Fourier 2D para obtener la imagen de intensidad. La grilla compleja se muestra usando el valor absoluto y la imagen se muestra usando su parte real.

La imagen de intensidad creada de esta forma se llama ***dirty* image** y es un imagen de calidad pobre. Existen otros algoritmos que, a partir de los mismos datos grideados, sintetizan imágenes de mejor calidad.

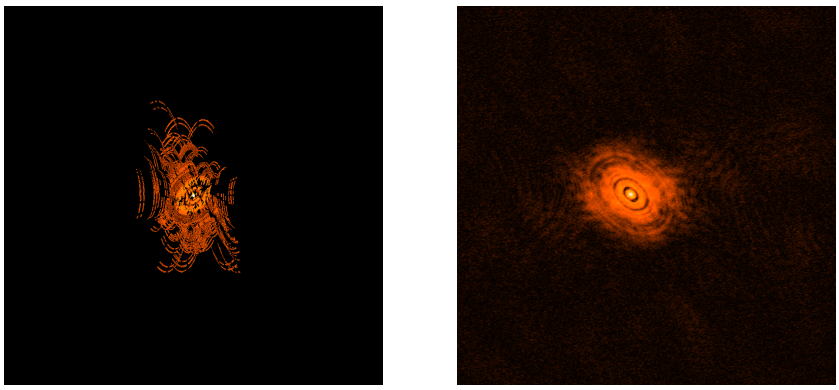

Figure 2: Datos grideados y su transformada inversa de Fourier, la imagen sucia.

# 3 Gridding - primera parte

El objetivo de este lab es implementar un sistema que transforme el plano $uv$ no grideado a uno grideado.

Las ecuaciones anteriores son válidas cuando la imagen y su transformada de Fourier están muestrados en una grilla regular. Es decir, las visibilidades y los píxeles están equidistantemente espaciados en las direcciones $u$ y $v$, y $x$ e $y$, respectivamente. Suponga que la distancia entre los píxeles de la imagen $I(x, y)$ es $\Delta x$ y $\Delta y$ en unidades de radianes. Entonces, la distancia en los puntos en su transformada $V(u, v)$ es:

$$\Delta u_\lambda = \frac{1}{N\Delta x_{\text{rad}}} \quad \Delta v_\lambda = \frac{1}{N\Delta y_{\text{rad}}} \tag{3}$$

Lamentablemente, los datos que ALMA captura no son completos ni regulares. Como se aprecia en la Figura 1, los puntos $(u, v)$ pueden estar en cualquier posición y no necesariamente están equiespaciados uno del otro. La Figura 3 muestra un zoom del plano $uv$ junto con una grilla regular. Por lo tanto los puntos $uv$ no pueden ser usados directamente para aplicar la inversa de Fourier y obtener la imagen sucia.

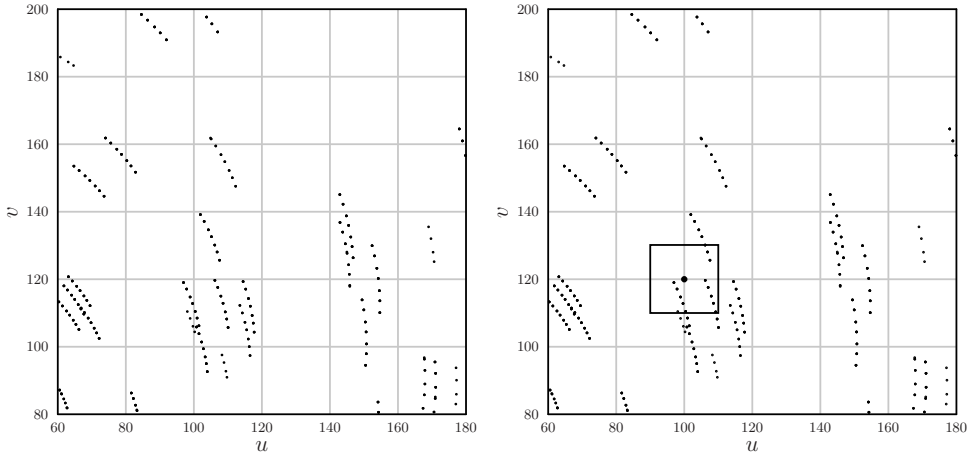


Figure 3: Zoom en el plano $uv$.

El proceso de gridding consiste en construir datos regularmente espaciados en el plano Fourier, a partir de los datos que entrega ALMA. Hay muchas formas de hacer esto, pero la más sencilla es sumar todos los valores de las visibilidades que se encuentran en una vecindad de cada punto de la grilla regular.

A continuación, explicamos con detalle esta operación.

Asuma que $\Delta u = \Delta v$ (y por lo tanto $\Delta x = \Delta y$ ). Estos valores se pueden considerar como la distancia entre dos celdas de la grilla, o la tasa de muestreo (sampling rate). Cada celda (cuadrado) en la figura 3 tiene un centro $(u_i, v_j)$. Suponga además que la grilla regular es de tamaño $N \times N$. Luego, los centros de las celdas son:

$$(u_i, v_j) = (i\Delta u, j\Delta v) \quad i, j = -(N/2-1), -(N/2-2), \ldots, -1, 0, 1, \ldots . N/2-1$$

Esta relación implica que:

1. Los puntos $(u_i, v_j)$ son coordenadas equiespaciadas en el plano $uv$ y no corresponden a los índices de la matriz.
2. El plano $uv$ está centrado en la grilla.
3. El eje $u$ comienza en valores negativos a la izquierda de la grilla y va hacia los positivos hacia la derecha
4. El eje $v$ comienza con valores negativos arriba de la grilla y va hacia los positivos hacia abajo.

El proceso de gridding consiste en acumular todos los valores de los puntos $uv$ que caen en una celda, en su centro. En realidad hay muchas formas de realizar esta suma, pero lo más simple es la suma.

# 4 Datos ALMA

Los datos de ALMA corresponden a una lista de $Z$ visibilidades del tipo

$$L = \{(u_k, v_k), V(u_k, v_k).r, V(u_k, v_k).i; k = 1, \ldots Z\}$$

donde

1. $(u_k, v_k)$ corresponde a la coordenada en el plano $uv$ de la visibilidad $k$
2. $V(u_k, v_k).r$ es el componente real del valor de la visibilidad $k$
3. $V(u_k, v_k).i$ es el componente imaginario del valor de la visibilidad $k$

La diferencia entre $(u_i, v_j)$ y $(u_k, v_k)$ es que $u_k$ y $v_k$ pueden ser cualquier valor, en cambio $u_i$ y $v_j$ son solo valores que corresponden a la grilla regular. El siguiente es un estracto de un archivo con visibilidades:

```
-2245.512935,-625.275579,-404.967700,0.019848,-0.009888,12065070.000000,224749993984.000000,0
-2245.512935,-625.275579,-404.967700,-0.000239,-0.023323,12001271.000000,224749993984.000000,0
-2245.512935,-625.275579,-404.967700,0.005969,-0.014794,12065070.000000,224250003456.000000,1
-2245.512935,-625.275579,-404.967700,-0.003359,-0.009349,12001271.000000,224250003456.000000,1
-2245.512935,-625.275579,-404.967700,-0.011006,-0.024791,12065070.000000,223749996544.000000,2
-2245.512935,-625.275579,-404.967700,-0.003334,-0.012453,12001271.000000,223749996544.000000,2
-2245.512935,-625.275579,-404.967700,-0.011025,-0.030222,12065070.000000,223250006016.000000,3
-2245.512935,-625.275579,-404.967700,-0.013623,-0.028793,12001271.000000,223250006016.000000,3
2532.096378,-2918.324430,-2675.440413,-0.010072,-0.004509,13042176.000000,224749993984.000000,0
2532.096378,-2918.324430,-2675.440413,-0.012755,0.006682,13798244.000000,224749993984.000000,0
```

A continuación se describe con detalle cada columna:

1. Coordenada $u$, en metros
2. Coordenada $v$, en metros
3. Coordenada $w$ en metros. No se usa en este lab
4. Parte real de la visibilidad
5. Parte imaginaria de la visibilidad
6. Peso $W$. Este es un valor de confianza en la medición
7. Frecuencia de observación $\nu$ en Hz.
8. Canal espectral. Cada canal tiene una frecuencia de observación. En este caso hay 3 canales de observación

Este conjunto de datos corresponde a HLTau, que es justamente el mostrado en la Figura 2. Para aumentar la cobertura en el plano $uv$ ALMA realiza observaciones en varias frecuencias. En este caso, cada punto $uv$ se mide en tres frecuencia, 224.75 GHz (canal 0), 224.25 GHz (canal 1), 223.75 GHz (canal 2). Además, cada canal se mide en dos correlaciones para muestrear la dirección del campo eléctrico de las ondas electromagnéticas, pero por ahora esto no es relevante.

Antes de realizar el gridding, se debe transformar las coordenadas $uv$ a número de longitudes de onda, de la siguiente forma:

$$u_\lambda = u \times \frac{1}{\lambda} = u \times \frac{\nu}{c}$$

donde $c$ es la velocidad de la luz en metros por segundo. Lo mismo se aplica para $v$.

Cada visibilidad tiene una estimación del error en la medición, el peso $W$. Mientras más grande este valor, menor la incertidumbre. Durante el gridding, los valores de las visibilidades deben ser ajustados según el peso, mediante una simple normalización, que se verá más adelante.

# 5 Gridding - segunda parte

En la Figura 3 las posiciones en las intersecciones de las líneas son las posiciones en la grilla regular, y las posiciones marcadas con puntos son posiciones de las visibilidades de la lista $L$. Entonces, a partir de los valores en los puntos deseamos estimar el valor en las intersecciones. En la Figura 3 se muestra la vecindad del punto de la grilla regular $(100, 120)$. Todos los puntos $uv$ que están dentro de esa vecindad cuadrada deben sumarse para formar el valor del punto en la grilla. Note que para este ejemplo $\Delta u_\lambda = \Delta v_\lambda = 20$.

Recordar que antes de comenzar el gridding, las coordenadas se deben transformar número de longitudes de onda. De ahora en adelante, se asume que ese proceso ya se realizó.

Sea $Fr$ y $Fi$ dos matrices de $N \times N$, que es el resultado del gridding para la parte real y para la parte imaginaria, respectivamente. Sea $Wt$ una matriz real de $N \times N$ que acumula los pesos correspondientes. Para determinar a que posición de la matriz corresponde una visibilidad $(u_k, v_k)$ basta calcular:

$$i_k = \left\lfloor \frac{u_{\lambda k}}{\Delta u_\lambda} \right\rceil + \frac{N}{2}, \quad j_k = \left\lfloor \frac{v_{\lambda k}}{\Delta v_\lambda} \right\rceil + \frac{N}{2}$$

donde el operador es el redondeo al entero más cercano. Por ejemplo, supongamos que $N = 8$, $\Delta u_\lambda = 0.5$, y que $u_{\lambda k} = -0.24$. Entonces $i_k = 4$. Si $u_{\lambda k} = 0.3$, $i = 5$.

Una vez determinado a que entrada de la matriz pertenece la visibilidad, acumulamos

$$\begin{aligned} Fr[i_k, j_k] &= Fr[i_k, j_k] + W_k \times V(u_k, v_k).r \\ Fi[i_k, j_k] &= Fi[i_k, j_k] + W_k \times V(u_k, v_k).i \\ Wt[i_k, j_k] &= Wt[i_k, j_k] + W_k \end{aligned}$$

Al finalizar de gridear todas la visibilidades, se realiza la normalización con los pesos acumulados:

$$Fr[i, j] = Fr[i, j]/Wt[i, j]$$

$$Fi[i, j] = Fi[i, j]/Wt[i, j]$$

# 6 Kafka

El sistema debe leer un archivo `csv` y crear productores de evento Kafka en donde los topicos sean los diferentes **canales espectrales**. Los productores deben enviar los datos $u, v, w, V_\text{o}, \omega$ y $\nu$ a Spark. Donde $V_\text{o}$ es la visibilidad de la muestra y tiene una parte real y una imaginaria.

# 7 Spark Streaming

Spark actuará como consumidor y consumirá los datos provenientes de distintos productores Kafka. Al consumir los datos, debe calcular el pixel en donde cae la visibilidad y además crear las grillas de visibilidades y pesos para ir colocándolas en los píxeles que corresponda. Finalmente, al terminar el consumo de datos, Spark debe realizar la IFFT para obtener la *dirty image* y almacenarla en un archivo HDF5.

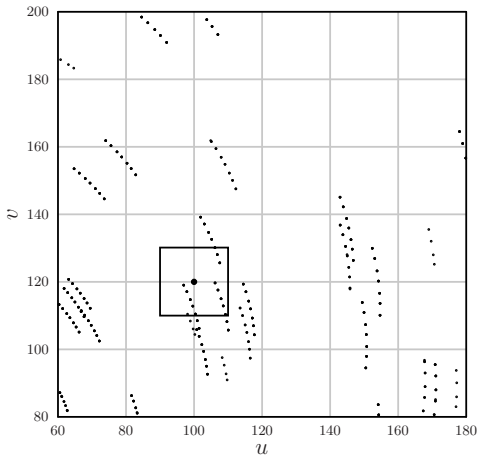


Figure 4: Ejemplo de vecindad que define los puntos que se suman para el punto de la grilla regular.

# 8 Entregables

El envío de los archivos se debe realizar a través de classroom.

1. Archivos python y/o un jupyter notebook.

**Fecha de entrega (hard-deadline):**
**18 de diciembre, 2023 antes de las 23:59 hrs.**

.