

MagicBox generación de imagen térmica

Cristian Flores, Maximiliano Mercado, Maciel Gabriel,
Marcelo Lorenz, Lautaro Lorenz

¹Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina
xcrisflores12@gmail.com, maximilianomercado.n@gmail.com,
maciel.gabriel@gmail.com, marcelo.lorenz@hotmail.com,
lautaro.lorenz@gmail.com

Resumen. En el presente documento se desarrolla la funcionalidad de generación de imagen térmica a emplear en el proyecto MagicBox. Para esto se empleará un modelo de procesamiento basado en MPI en donde se genera una red conformada por todas las MagicBox que se encuentren en el mercado. A mayor cantidad de MagicBox disponibles, mayor será el poder de procesamiento conjunto, necesario para resolver la generación de la imagen térmica que muestra el mapa de temperatura dentro de una única MagicBox. Se pretende con el mecanismo descrito, ahorrar en los costos debidos a soluciones basadas en adquirir nuevo hardware, aprovechando los tiempos ociosos de los procesadores que ya están disponibles por el simple hecho de haber adquirido una unidad de MagicBox.

Palabras claves: MagicBox, HPC, MPI, peer-to-peer, temperatura.

1 Introducción

MagicBox es un proyecto de código abierto basado en la administración de su contenido, saber si tengo lo suficiente de un producto o debo adquirir más, y mantener el producto en condiciones adecuadas para favorecer el alargamiento de su vida útil el mayor tiempo que sea posible. Para cumplir su objetivo principal, cuenta con sensores que cumplen la función de medir el peso, el volumen ocupados y la temperatura. En este documento ahondaremos en la capacidad de generar una imagen térmica del producto contenido, basados en una matriz de sensores infrarrojos.

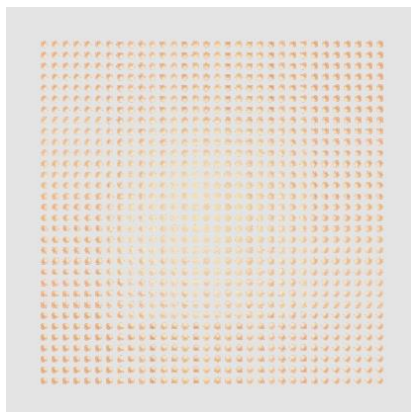
Generar una imagen térmica de objetos resulta ser una característica que aporta alto valor agregado; Sin embargo es algo complejo que requiere de un poder de procesamiento considerable, el cual no es posible de obtener con un único Arduino R1. Para resolver este inconveniente ofrecemos una solución que abarque el problema desde la raíz y para ello comenzaremos explicando como se obtendrán las mediciones de temperatura, luego explicaremos como serán procesadas, y finalmente como llegará esto a satisfacer los requerimientos del usuario.

2 Desarrollo

Obtención de la temperatura

La intensidad de la radiación emitida por un cuerpo negro con una temperatura T viene dada por la Ley de Planck ^[2]. La aplicación de la ley de Planck nos sirve para obtener la temperatura superficial midiendo las longitudes de onda infrarrojas.

La obtención de mediciones de temperatura se hará por medio una matriz de 32x32x32 sensores infrarrojos conectados en cada cara interna del recipiente de almacenamiento, al ser un recipiente cuadrado, mostraremos una imagen de una única cara ya que las 5 caras restantes son idénticas.



1 Matriz de 32x32 sensores infrarrojos

Al haber 1024 sensores infrarrojos por cada cara interna del recipiente y al ser 6 caras en total, se necesita procesar la información de 6144 sensores infrarrojos por cada mapa de temperatura que se realice.

Cada sensor tomará la longitud de onda infrarroja medida con una resolución de 12bit la cual será almacenada en un registro de una memoria flash de 6144x12 bit es decir 9.216kb de tamaño total. Dicha memoria es capaz de guardar la última medición de cada sensor, para esto cada sensor tiene asociado un registro de la memoria, es por esto, que un mismo sensor almacena el valor censado, siempre en la misma posición de memoria.

Procesamiento de los valores obtenidos

Para conseguir el poder necesario, se utilizará un sistema de comunicación punto a punto, en donde cada nodo de la red estará conformado por un procesador de MagicBox. El nodo que haga una solicitud de procesamiento al resto de los nodos mediante mensajes MPI (de ahora en adelante “nodo solicitante”) recibirá como respuesta la cantidad de sensores que puede procesar cada nodo que responda la solicitud (desde ahora llamados “nodos receptor”).

El nodo solicitante entonces, informa a los nodos receptor los registros de su memoria interna a la cual deben acceder para leer los valores obtenidos por los sensores de temperatura. Es decir, por ejemplo, si el nodo solicitante “A”, solicita poder de procesamiento en la red, y el nodo receptor “B” le responde que puede procesar 5 sensores, entonces el nodo “A”, le responderá que lea los registros en las posiciones 0, 1, 2, 3 y 4, recordando que cada posición es un registro de 12bits ^[3] con la medida de temperatura obtenida por los sensores 1, 2, 3, 4 y 5 de la matriz de sensores. Pudiendo estos sensores estar ubicados en cualquier punto de la matriz.

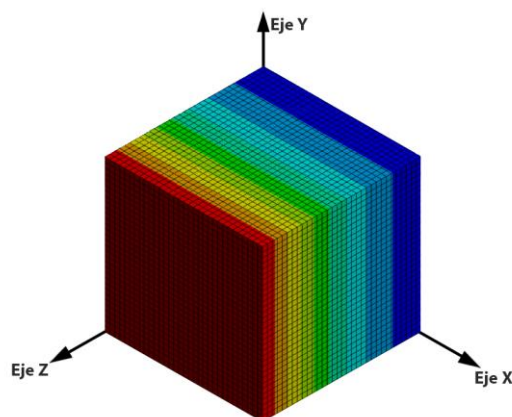
Una vez que los valores son leídos por los nodos receptor, estos generarán una porción de la imagen final del mapa de temperatura, para lograr esto, cada valor en el registro de 12bit tendrá asociado un color (en una escala de colores predeterminada) donde el azul estará asociado a las temperaturas más frías (es decir con longitudes de onda más grandes) mientras que en contra parte el rojo estará asociado a las temperaturas más calientes, es decir, con longitudes de onda más chicas.

Luego de que el nodo receptor haya generado la imagen en forma de matriz de colores, la misma es devuelta al nodo solicitante.

Es importante no confundir al “nodo solicitante” con un “nodo maestro” del protocolo OpenMP, en la configuración que estamos presentando, todos los nodos pueden ser solicitantes o receptores según sea el caso, por lo que todos ejecutan un mismo código.

Interpretación de los valores procesados

El nodo solicitante recibe de los nodos receptores una matriz de colores, donde cada color representa la temperatura que fue obtenida en cada punto de la matriz de sensores infrarrojos, será su trabajo unir las matrices de colores recibidas para presentarlas al usuario de MagicBox en forma de una imagen que pueda interpretar, cabe destacar que como la matriz de sensores es discreta a la imagen que se le muestra al usuario se le puede aplicar un algoritmo de suavizado para que le sea más armónica a la vista, siendo este un tema que se escapa del alcance del presente documento.



2 Mapa de temperatura de 32x32x32, interpretado

3 Explicación del algoritmo.

En un momento dado, un nodo (MagicBox) decide generar un mapa térmico de su recipiente de almacenamiento, entonces se dispara la ejecución del siguiente algoritmo.

Para interpretar el algoritmo adecuadamente, tenga en cuenta que las sentencias de más arriba ocurren antes en el tiempo que las sentencias que le sigan más abajo, independientemente del sentido de izquierda a derecha.

NODO SOLICITANTE

```
inicializarSolicitudDeMapaTermico();
solicitarProcesamientoExterno();

indicarRegistrosDeMemoria();

unirMapasDeColores();
mostrarMapaDeTemperatura();
```

NODO RECEPTOR

```
MPI_INIT();
escucharSolicitudesProcesamiento();

responderPoderDeProcesamiento();

leerRegistrosDeMemoria();
generarMapaDeColores();
responderMapaDeColores();
MPI_Finalize();
```

4 Pruebas que pueden realizarse

Para poder llevar a cabo la propuesta debe haber en el mercado suficientes unidades de MagicBox como poder de procesamiento sea necesario, a mayor cantidad de unidades, mayor poder de procesamiento. Cuando el usuario solicite un mapa de temperatura interna en el recipiente de almacenamiento. Se realizará la solicitud a todas las unidades de MagicBox que se encuentren disponibles en la red y puedan ayudar a resolver la petición de la solicitante.

Finalizado el procesamiento el usuario debe conectar la aplicación móvil al sistema embebido para poder observar el mapa de temperatura generado, el cual deberá presentar 6144 lecturas de temperatura, como si cada una hubiese sido tomada, procesada y mostrada en el mismo instante de tiempo que las demás.

5 Conclusiones

En un contexto económico desfavorable donde el mercado de venta de electrdomesticos atraviesa un período de caída constante y prolongada, los bienes tienden a ser cada vez más caros y proveernos de menos características o de menor calidad. Es por este motivo, que la solución planteada está basada en el aprovechamiento de los recursos ya disponibles en el mercado, y en una mejora del producto, sin elevar el costo final significativamente.

Como el mapa de temperatura obtenido está basado en una matriz de sensores discreta es lógico que el resultado sea un mapa de temperatura también discreto, sin embargo podría aplicarse al mismo un algoritmo de suavizado utilizando el GPU del dispositivo Android desde el cual se visualiza la imagen final.

Para finalizar cabe mencionar que la lógica aplicada puede ser utilizada para generar diferentes tipos de censado de magnitudes, cambiando los sensores utilizados, por ejemplo se podría medir radiación ultravioleta, electromagnética u ionizante solo por poner algunos ejemplos.

6 Referencias

1. WaterMaster: Reconocimiento del usuario mediante huellas dactilares <https://github.com/ilandaburu/WaterMaster/blob/master/HPC/WaterMaster%20Reconocimiento%20del%20usuario%20mediante%20huellas%20dactilares.pdf>
2. Ley de Planck https://www.ecured.cu/Ley_de_Planck
3. Characteristic Analysis of DS18B20, Song Nie, Yang-chun Cheng, Yuan Dai, Scientific Research, Beijing Key Laboratory of High Voltage & EMC
4. Configuración y puesta en marcha de una red MPI, Electricidad y electrónica I.E.S Himilce – Linares
5. Programación Distribuida y Paralela, Universidad de Granada, Daniel Guerrero Martínez y Sergio Rodríguez Lumley (2010)