

Trabajo de Investigación: Integración HPC con proyecto Smartender

Juan Cyc, Camila de Jesús, Germán López, Carlos Speranza

Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina
juancyc@gmail.com, camila.agus.dj@gmail.com, germanlopez94@gmail.com,
carlossperanza5@gmail.com

Resumen. En el siguiente escrito se procederá a explicar brevemente los beneficios de la implementación de la tecnología HPC sobre el proyecto de IoT llamado *Smartender*. Tal proyecto persigue la tarea de solucionar un problema típico para cualquier dueño de hogar, consistente en el secado de la ropa en épocas lluviosas o particularmente húmedas comunes en el área de Buenos Aires. El objetivo de esta investigación tratará sobre cómo implementar HPC utilizando MPI. Los nodos a ser considerados serán un servidor junto a los distintos dispositivos Arduinos y Androids de la zona determinada, de manera que se optimice lo mejor posible la toma de decisiones realizadas por nuestro sistema, basándose en una mejor precisión obtenida al promediar todos aquellos dispositivos zonales.

Palabras claves: Arduino, CUDA, Android, paralelismo, MPI, HPC.

1 Introducción

Smartender consiste en un sistema embebido conformado por un tender conectado a un motor bidireccional, el cual es controlado por un Arduino con diversos sensores de reconocimiento ambiental tales como la humedad, presión, temperatura e intensidad de luz. Con ellos, es capaz de determinar si las condiciones climáticas son adecuadas o no para mantener la ropa colgada al exterior o, en su defecto, mantenerlas bajo techo. Asimismo, desde el dispositivo Android del usuario, éste será capaz de decidir por su propia cuenta en base a la información recopilada por el embebido e información del pronóstico zonal brindada por la nube.

Hoy en día, gracias al gran desarrollo del sistema operativo Android y sus dispositivos, cualquier persona puede acceder a un pronóstico del día o extendido para su ciudad en unos minutos. Esto último, da la posibilidad a estos usuarios de tener conocimiento exacto sobre el ambiente de su ciudad, a diferencia del método previo en el cual debía utilizar un periódico o ver algún informativo para saber sobre una zona mucho más abarcativa que su propia ubicación.

A pesar de ello, la eficacia de este pronóstico puede no ser tan exacta como lo sería tener la temperatura y humedad propia de un barrio o incluso una manzana, permitiendo así determinar con mayor exactitud si las condiciones del perímetro aledaño al usuario son favorables para diversos usos.

Debido a esto, diversas aplicaciones Android acceden a información brindada por APIs climáticas o servicios meteorológicos en línea con formato XML¹ a los cuales el desarrollador de dicha aplicación podrá acceder para utilizar aquello que le sea útil.

Los sistemas actuales de predicción climática sufren la terrible necesidad de estar constantemente en falta de hardware de última generación para poder aumentar la precisión de sus pronósticos. Una potencial solución a este problema radicaría en la utilización no de un supercomputador con altas prestaciones para medir las distintas variaciones de cierta zona sino, utilizar múltiples y pequeños procesadores con la capacidad de generar cómputo en paralelo en el interior de la misma zona que abarcaría el sistema especializado a un mucho mayor costo.

Se buscará en esta investigación optimizar el reconocimiento al crear redes zonales comunicadas entre sí para obtener resultados más precisos y minimizar no sólo el tiempo de respuesta sino la variabilidad de la predicción.

2 Desarrollo

Hoy en día, se pueden obtener proyecciones a 7 días con el 95% de eficacia gracias al conocido *Big Data*, los satélites y las computación en paralelo. A pesar de todas las mejoras obtenidas gracias al crecimiento de la tecnología, Vijay Tallapragada de la Administración Nacional Oceánica y Atmosférica de los Estados Unidos ha explicado que la dependencia hacia los supercomputadores de altas prestaciones con miles de CPUs y una extensa memoria interna se encuentran limitados físicamente por la necesidad de aún más recursos de cómputo².

Utilizando HPC junto a *Smartender*, se podría solucionar inicialmente el problema de las imperfecciones relacionadas a la medición del clima con herramientas básicas, al sumarle la exactitud y capacidad de procesamiento de conjuntos de datos simultáneos y de múltiples variables. Luego, como segundo punto a tener en cuenta, se podría destacar de las supercomputadoras utilizadas por agencias nacionales de meteorología al poner a trabajar no un procesador de alta capacidad sino muchos de ellos pequeños con tareas en paralelo detectando así hasta la más mínima variación en un barrio o zona cerrada de, por ejemplo, 300 metros. Por último, el sistema podría ser mejorado una vez más sumando el uso de inteligencia artificial, encargada de estudiar los patrones asociados a las malas condiciones climáticas y poder tomar la mejor acción para el usuario según sea el caso.

Como primer paso, la investigación tomará la información expuesta en la XX WICC 2018³, cuyo tema a desarrollar consistía en el aumento de la precisión a la hora de simulación de aguas para evitar pérdida de recursos económicos debido a inundaciones inesperadas, utilizando HPC para confrontar dicha problemática. Otro enfoque a destacar sobre aquel proyecto, es que permitiría adicionalmente a la capacidad de utilizar paralelismo para aumentar la precisión, un ahorro energético considerable al minimizar los recursos computacionales.

Sabiendo gracias a esto que el uso de HPC puede brindar mayor precisión en la lectura de valores ambientales, la investigación propuesta para *Smartender* continuará su rumbo sobre cómo dividir los clúster⁴ (o sea, el conjunto de computadoras unidas mediante una red de alta velocidad funcionando como una única computadora) quienes son fundamentales a la hora de maximizar la eficiencia de ejecución en paralelo.

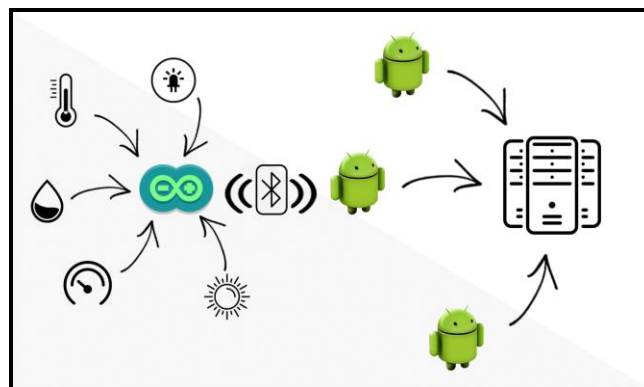
Al momento de programar nuestro clúster zonal, conformado por los dispositivos Android, será asignado un servidor al cual cada celular del sector enviará la información que ellos recopian por la API del clima de su ubicación. De esta manera, el servidor operará como dispositivo anfitrión de nuestro sistema distribuido mientras que los celulares serán los dispositivos secundarios.

Para mejorar la precisión, sería conveniente que el celular Android comunique adicionalmente la información obtenida por su Arduino dependiente. Sin embargo, esto ralentizaría la ejecución al tener que comunicarse constantemente y recomunicando nuevamente lo obtenido al servidor. Una forma de optimizar esto sería utilizar el *middleware* HPC Java, o también llamado JCL⁵, el cual permite trabajar con dispositivos tanto Android como Java de manera asíncrona específica para IoT.

Para la implementación de MPI en dispositivos móviles Android utilizaremos la biblioteca MPICH2 que permitirá armar los clúster portables con dispositivos que usen el sistema operativo Android⁶.

3 Explicación del algoritmo.

Para desarrollar la solución propuesta debemos implementar los distintos clústeres locales formados por los dispositivos móviles Android que recopilarán los datos climatológicos de sus respectivos tenderos conectados y a través de MPI procesarán los datos leídos para obtener información más precisa del estado del tiempo que a su vez será reportada al servidor central de nuestra implementación, programado a través de CUDA para mejorar su performance. El sector secuencial del proyecto, por otro lado, se encontraría en la comunicación tanto entre el servidor y los celulares, como con éstos y los embebidos. Esto queda representado en el siguiente diagrama:



El servidor, con toda la información recopilada procederá a un análisis de dicha zona para poder brindar posteriormente a los usuarios de *Smartender* los valores precisos del clima en su ubicación.

```
1  --Cluster Local
2
3  Nodo de entrada (Android)
4  (1)//Recibir y mostrar informacion del servidor
5  //Leer informacion del Arduino
6  //Procesar string de entrada
7  //Guardar temperatura, humedad y presion atmosferica
8
9  mpi_gather(); //datos de los otros nodos
10 mpi_scatter(); //informacion a refinar
11 mpi_gather(); //informacion refinada
12 return a (1);
13
14 Nodos de procesamiento (Android)
15 //Recibir y mostrar informacion del servidor
16 //Leer informacion del Arduino
17 //Procesar string de entrada
18 //Guardar temperatura, humedad y presion atmosferica
19 mpi_recv(); //datos a refinar
20 refinarInformacion(); //Aqui hace un calculo para el refinamiento
21 mpi_send(); //envio del resultado
22 return a (1);
23
24 --Nodo central (PC con GPU)
25
26 //Leer informacion de clusters locales
27 //Procesar informacion y obtener informacion refinada de los clusters
28
29 cuda_iniciar();//hilos para obtener lectura precisa del estado del tiempo en la zona
30 cuda_leerResultado();
31 enviarInfoPrecisaASmartender();
```

4 Pruebas que pueden realizarse

Para probar si la mejora propuesta al sistema original *Smartender* funcionase, la prueba fundamental a la hora de evaluar dicha mejoría consistiría en ver si la temperatura obtenida como resultado del cálculo realizado por el servidor, es considerablemente distinta a la provista por las APIs de clima utilizadas por alguno de los celulares partícipes de la prueba. Luego, utilizar alguna herramienta de medición climática tales como termómetros de ambiente, barómetros, sensores de luz o higrómetros y comparar efectivamente que no haya algún error de reconocimiento por parte de los sensores conectados a los Arduinos. De ser exitosa, se puede afirmar que el proyecto de investigación con base a *Smartender* y HPC cumplió con su finalidad.

5 Conclusiones

En base a lo investigado y desarrollado, podemos afirmar que la propuesta para ampliar *Smartender* es no sólo factible y oportuna, sino incluso una mejora considerable al estilo de vida de cualquier persona encargada de su hogar, especialmente para aquellos residentes de lugares donde la mayoría de las APIs climáticas no reconocen ni poseen en la lista disponible de pronósticos.

A futuro, la idea de implementar inteligencia artificial en el nodo servidor podría agregar la funcionalidad de comunicarle a dichas personas, desde proyecciones del clima en base a históricos de su barrio particular hasta recomendaciones particulares en base a dicho historial y los hábitos semanales o mensuales de manera de buscar alguna optimización sobre cuándo potencialmente debería colgar su ropa o cuando sacarla. Por ejemplo, si un día de la semana un usuario comienza a trabajar más temprano que lo normal, recomendar en base variables climáticas previas, cuál sería el mejor horario para que se encargase de sus prendas.

6 Referencias

1. How do weather apps work? What is their source of information?
<https://www.quora.com/How-do-weather-apps-work-What-is-their-source-of-information>
2. HPC, weather prediction, and how you know it's going to rain.
<https://www.hpe.com/us/en/insights/articles/hpc-weather-prediction-and-how-you-know-its-going-to-rain-1801.html>
3. Trigila, Mariano;Gaudiani, Adriana Angélica; Luque, Emilio; Naiouf, Marcelo: Simulación computacional, ciencia de los datos, cómputo de alto rendimiento y optimización aplicados a mejorar la predicción de modelos de simulación que representan la evolución de sistemas complejos. XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste). UNLP. (2018)
4. Guamán Rivera; Brayme Lino: Análisis de rendimiento de un clúster HPC y arquitecturas manycore y multicorer. Tesis de grado. Cuenca. (2017)
5. Instituto de Ciencias Exatas e Biologicas; Universidade Federal de Ouro Preto ; Instituto Federal de Minas Gerais; Universidade Federal de Alagoas. JCL Android: Uma Extensao IoT para o Middleware HPC JCL. Investigación compartida entre dichas universidades. Brasil. (2017).
6. Zachary Yannes, Portable MPICH2 clusters with Android devices. Tesis de grado. Florida State University, Estados Unidos (2015).