

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

LENGUAJES Y AUTOMATAS II

Propuesta para el desarrollo de proyecto final:

ELABORACION DE PROYECTO DE OPENMP

Presenta:

ORTEGA BUSTAMANTE ANTONIO

No. De Control:

171080148

Maestro:

Parra Hernández Abiel Tomas

CIUDAD DE MEXICO

ENERO/2021

Resumen

OpenMP proporciona un tiempo de ejecución usada para la implementación de una interfaz de programación para aplicaciones (API), esto nos permite añadir concurrencias a los programas escritos en C, C++, etc. Esto se encuentra disponible en muchas arquitecturas, incluidas para plataformas como Unix y de Microsoft Windows, y se compone de un conjunto de directivas de compilador, rutinas de biblioteca y sus variables de entorno que incluyen un comportamiento en su tiempo de ejecución. OpenMP es un modelo de programación portable y escalable que nos proporciona a los programadores una interfaz simple y flexible para el desarrollo de aplicaciones paralelas que van desde las computadoras de escritorio hasta supermercados.

Cronograma de actividades

INSTITUTO TECNOLÓGICO DE IZTAPALAPA
CRONOGRAMA DE RESIDENCIAS PROFESIONALES

Alumno:	ORTEGA BUSTAMANTE ANTONIO	No. de Control	171080148
Carrera:	ING. EN SISTEMAS COMPUTACIONALES		
Nombre de Proyecto:	Subproyecto openMP		
Asesor Interno:	Parra Hernandez Abelil Tomas		
Fecha de inicio:	21 de septiembre del 2020	Fecha de terminación:	20 de enero del 2021
objetivo de proyecto:	entendimiento de subproyecto VLLM		

Actividad		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
RECOLECCION DE INFORMACION	p																
	n																
ELABORACION DE DOCUMENTO	p																
	n																
ELABORACION DE JUSTIFICACION	p																
	n																
ELABORACION DE SUBPROYECTO	p																
	n																
ENTREGA DE SUBPROYECTO EN GITLAB	p																
	n																

Observaciones:

EN LA ELABORACION DE ESTE PROYECTO DOS INTEGRANTES DEL EQUIPO DINAMITA DEJARON DE PRESENTARSE A LA CLASE DEJANDO A ANTONIO ORTEGA REALIZANDO SOLO EL PROYECTO

Lista de soluciones y riesgos

Lista de riesgos	Lista de soluciones
1) No conocer bien el entorno de desarrollo	1) Capacitación sobre el entorno de desarrollo
2) No comprender el funcionamiento del proyecto	2) Conocer el comportamiento y funcionamiento del proyecto
3) No capacitar al personal sobre el funcionamiento del proyecto	3) Capacitación básica para el personal
4) Ingresar datos erróneos	4) Conocer los datos que pueden ingresarse
5) No Conocer la eficiencia y funcionalidad del proyecto	5) Conocer eficiencia o límites del proyecto

El subproyecto OpenMP de LLVM contiene los componentes necesarios para construir un programa OpenMP ejecutable que está fuera del compilador.

Aquí puedes encontrar: –

- el código de la biblioteca en tiempo de ejecución con el que se debe vincular el código compilado por `clang -fopenmp` antes de que pueda ejecutarse.
- la biblioteca que admite la descarga a los dispositivos de destino (en "libomptarget")

El soporte para las partes del lenguaje OpenMP 4.0 (y posteriores) que no están asociadas con las construcciones "objetivo" se encuentran en el directorio "runtime". El soporte para descargar cálculos a través de la directiva "target" se encuentra en el directorio "libomptarget" separado.

Todo el código aquí tiene [doble licencia](#) bajo la licencia MIT y la licencia UIUC (una licencia similar a BSD). El archivo LICENSE.txt en la parte superior del proyecto OpenMP contiene el texto de la licencia y las concesiones de patentes asociadas.

Estado

Con el lanzamiento de Clang 3.8.0, el soporte de OpenMP 3.1 está habilitado en Clang de forma predeterminada y, por lo tanto, el tiempo de ejecución de OpenMP se construye como una parte normal de la compilación de Clang y se distribuye con las distribuciones binarias. Por lo tanto, no es necesario explícitamente para verificar este código, o construirlo a partir de un árbol; una comprobación y compilación normal de Clang incluirá automáticamente la compilación de estas bibliotecas en tiempo de ejecución.

Funciones y objetivos

- La compatibilidad con el se ha logrado en la versión Clang 3.8.0.
- Ahora se está implementando la compatibilidad con el . (Algunas funciones de OpenMP 4.0 y 4.5 ya están disponibles).
- Alto rendimiento.
- Compatibilidad ABI con [los compiladores OpenMP existentes de Gcc](#) e [Intel](#). Actualmente tenemos compatibilidad binaria con el código OpenMP 3.1 compilado por gcc 4.9, sin embargo, no tenemos soporte para el código OpenMP 4.0 que usa la cancelación de tareas cuando se compila con gcc 4.9. Cómo apoyaremos dicho código sigue siendo un tema de investigación.

JUSTIFICACION

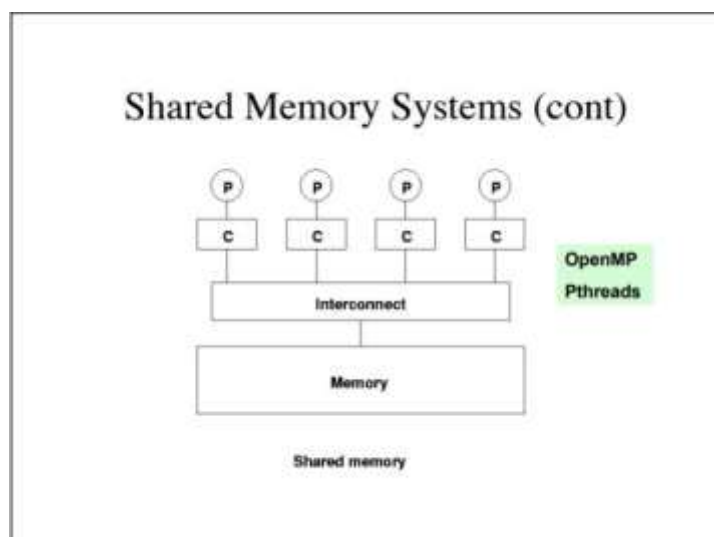
El subproyecto OpenMP de LLVM posee la compatibilidad con el estándar OpenMP 3.1 (PDF), debido a que proporciona un tiempo de ejecución usada para la implementación de una interfaz de programación para aplicaciones (API), esto nos permite añadir concurrencias a los programas escritos en C, C++, etc.

se compone de un conjunto de directivas de compilador, rutinas de biblioteca y sus variables de entorno que incluyen un comportamiento en su tiempo de ejecución. OpenMP es un modelo de programación portable y escalable que nos proporciona a los programadores una interfaz simple y flexible para el desarrollo de aplicaciones.

Sintaxis de OpenMP

- La mayoría de las construcciones en OpenMP son directivas de compilación o pragmas. – En C y C++, los pragmas tienen la forma:
 - #pragma omp construct [clause [clause]...] – En Fortran, las directivas tienen la forma:
 - C\$OMP construct [clause [clause]...]
 - !\$OMP construct [clause [clause]...]
 - *\$OMP construct [clause [clause]...]
- Como las construcciones son directivas, un programa en OpenMP puede ser compilado por compiladores que no soportan OpenMP.

Por lo que este programa contine una gran importancia en la manera de compiladores debido a su alta funcionalidad en diferntes industrias que se manejan o dedican al uso de este subproyecto añadiéndolos a otros diferentes proyectos que dependen de este tipo de subproyecto proporcionado por llvm.



Metodología:

Desarrollo Rápido de Aplicaciones (RAD)

Con el objetivo de otorgar resultados rápidos, se trata de un enfoque que está destinado a proporcionar un excelente proceso de desarrollo con la ayuda de otros enfoques, pero, además, está diseñado para aumentar la viabilidad de todo el procedimiento de desarrollo de software para resaltar la participación de un usuario activo. Dicho esto, algunas de las ventajas a destacar de este tipo de desarrollo son las siguientes:

- Hace todo el proceso de desarrollo sin esfuerzo.
- Asiste al cliente en la realización de revisiones rápidas.
- Alienta la retroalimentación de los clientes para su mejora.

se trata de un modelo de desarrollo de aplicaciones ágil. Es decir, hablamos del proceso de desarrollo de software. Este método abarca el desarrollo interactivo, la creación de prototipos y el empleo de utilidades CASE (Computer Aided Software Engineering). Además, la metodología RAD suele englobar también la usabilidad, utilidad y la rapidez de ejecución.

Estas son algunas de las ventajas que ofrece el desarrollo rápido de aplicaciones:

- Aumenta la versatilidad y la adaptabilidad, dado que los desarrolladores pueden hacer los ajustes necesarios de forma inmediata durante el proceso de desarrollo.
- Las iteraciones rápidas reducen el periodo de desarrollo y agilizan la entrega.

- Se fomenta la reutilización del código, por lo que se reduce la programación manual y, en consecuencia, disminuyen tanto la posibilidad de cometer errores como los periodos de prueba.

proponemos una lista de comprobaciones que te ayudará a saber si tu equipo está preparado para adoptar esta metodología:

- ¿Necesitas desarrollar productos de software dentro de un lapso corto de tiempo (de dos a tres meses)?
- ¿Cuentas con un equipo experimentado de desarrolladores, programadores y diseñadores que pueda llevar a cabo el trabajo en los plazos establecidos?
- ¿Tu cliente está dispuesto a adoptar el desarrollo rápido de aplicaciones?
O lo que es lo mismo, ¿crees que el cliente estará disponible para colaborar durante todo el proceso de desarrollo del software?

5 pasos del desarrollo rápido de aplicaciones

Paso 1: Definir y concretar los requisitos del proyecto

Paso 2: Comenzar a diseñar los prototipos

Paso 3: Recopilación de las opiniones del usuario

Paso 4: Pruebas, pruebas y más pruebas

Paso 5: Presentación del sistema