

# Programació Paral·lela amb Múltiples Fluxos en un Entorn de Memòria Compartida (Many-Core)

## Introducció

Enguany, farem us d'una iniciativa nascuda en el workshop EduPar de la conferència IPDPS [1] i el workshop EduHPC de la conferència SuperComputing [2] (probablement dues de les més prestigioses del món en HPC). Es tracta d'un conjunt de problemes que poden ser utilitzats en pràctiques d'assignatures de Còmput d'Altes Prestacions o Sistemes Distribuïts i que s'han inclòs dins d'un projecte més ambiciós de la NSF/IEEE\_TCPP pel disseny curricular i l'elaboració de continguts d'assignatures de l'àmbit [3].

De totes les possibilitats existents en aquest projecte, hem escollit un sistema de simulació basat en agents d'extinció d'incendis [4]. Us proporcionarem un codi seqüencial en C que simula l'extinció d'un incendi, amb diferents focus d'ignició, per part d'un o més equip de bombers.

Aquest codi es farà servir durant tot el curs per anar generant versions paral·leles fent servir els diferents paradigmes que estudiarem. Així, per aquesta segona pràctica la idea és desenvolupar una versió paral·lela d'aquest programa fent servir múltiples fluxos d'execució en una arquitectura many-core utilitzant OpenACC o CUDA i fer un estudi dels guanys de rendiment que obteniu.

A continuació es descriu el problema, els requeriments i les condicions del lliurament d'aquesta pràctica.

## Descripció del programa

El codi proporcionat per aquesta pràctica fa una simulació simplificada d'un incendi amb un o més focus. Tindrem una matriu que representarà la superfície rectangular on es durà a terme la simulació (*surface*) i un conjunt d'estructures per emmagatzemar els paràmetres dels equips de bombers i focus de l'incendi.

La Figura 1 mostra un estat possible de la simulació per a una superfície de 30x30 punts després d'un cert nombre d'iteracions. Els següents símbols s'utilitzen per a representar l'estat de cadascun dels punts:

**Arguments del programa.** El programa rep una descripció del escenari de la simulació com a fitxer de text o mitjançant els següents arguments :

**size1, size2:** Mida de l'array bidimensional que guarda la informació de la temperatura a la superfície.

**maxIter:** Màxim nombre d'iteracions que durà a terme la simulació.

**numTeams:** Nombre d'equips de bombers presents a la simulació.

**teamDataList:** Tupla de tres números separats per espais que indica la posició inicial de cada equip (x, y) i la seva capacitat d'extinció d'incendis (1, 2 i 3).

**numFocalPoints:** Número de focus de l'incendi.

**FocalPointsDataList:** Tupla de 4 números separats per espais que indiquen per cada focus; posició (x, y), pas en el qual s'activarà i quantitat de calor constant que emetrà el foc un cop actiu.

*Format dels fitxers de superfície.* Els arguments del problema (introduïts a la secció anterior) es poden passar mitjançant un fitxer de text. La primera línia del fitxer ens indica els valors de *size1*, *size2* i *maxIter*. La segona conté el número d'equips de bombers presents a la simulació. Després tindrem una línia per equip on s'indicarà la seva posició inicial (x, y) i el seu tipus (1,2,3). Un cop acabem de descriure els equips de bombers tindrem una línia que ens indicarà el nombre de focus que hi haurà a l'incendi, seguida d'una línia per focus on es descriurà la seva posició (x, y) i el pas de la simulació on s'activarà i la quantitat de calor que emetrà un cop actiu. Podeu veure exemples de fitxers d'input per a la simulació al directori */test\_files* dins de la carpeta de la pràctica.

*Descripció funcional.* Al començament el programa inicialitza tots els punts de la superfície a temperatura ambient (0). Un cop inicialitzats començarà la simulació de l'incendi que compta amb diferents fases (numerades de la mateixa manera que als comentaris del codi del programa):

**4.1. Activació dels focus.** S'activen tots els focus que han s'encenen per aquest pas de la simulació. Es compten el nombre de focus que han estat desactivats durant la simulació.

**4.2. Propagació de calor.** Aquesta fase realitza 10 passos de propagació del calor per la superfície abans de que els bombers comencin a actuar.

**4.2.1. Actualització de la temperatura als focus actius.** S'actualitza la temperatura dels punts amb focus actius, si un focus encara no s'ha activat no canvia la seva temperatura.

**4.2.2. Còpia a la superfície auxiliar.** Es copien els valors de la superfície a una superfície auxiliar.

**4.2.3. Actualització dels valors de la superfície.** Es calcula el valor de cada punt de la superfície com el valor mitjà dels seus veïns directes al pas anterior de la simulació.

**4.2.4. Càlcul del calor residual.** Es calcula el calor residual com el màxim (de entre tots els punts de la simulació) de la diferència entre la temperatura actual d'un punt i la temperatura al pas anterior de aquest mateix punt.

Després de la fase de propagació de calor es decideix si s'han estabilitzat l'incendi (s'han apagat tots els focus i el calor residual és per sota d'un llindar determinat).

**4.3. Moviment dels equips de bombers.** Un cop el calor s'ha propagat els equips de bombers entren en acció.

**4.3.1. Elecció del focus.** Cada equip de bombers escull el focus encès que es troba més proper a la seva posició.

**4.3.2. Elecció de l'objectiu.** El focus escollit a 4.3.1 es marca com objectiu de l'equip.

**4.3.3. Comprovació objectiu.** Si un equip no té objectiu no es mourà.

**4.3.4. Moviment cap al focus.** Si l'equip té un objectiu es mourà una posició en direcció a aquest. Cada equip es mou d'una forma diferent, els equips de tipus 1 es mouren en diagonal, els de tipus 2 primer horitzontalment i després en vertical i per últim els de tipus 3 es mouren primer verticalment i després en horitzontal.

**4.4. Acció dels bombers.** Un cop els bombers s'han mogut realitzaran les accions pertinents.

**4.4.1. Desactivació del focus.** Si un equip ha arribat a un focus actiu el desactivarà.

**4.4.2. Reducció del calor a la zona.** Es disminueix la temperatura a un radi fixe al voltant de cada equip. El radi d'actuació dels equips de tipus 1 es menor al dels de tipus 2 i 3.

La simulació acabarà un cop s'apaguin tots els focus o quan el calor residual sigui per sota el llindar d'estabilitat. Al acabar el programa mostra el temps que ha transcorregut durant la simulació i alguns valor per verificar que s'ha executat correctament (posició final dels equips de bombers y calor residual dels focus).

**Mode debug.** Si compilem el codi amb el flag **-DDEBUG** es mostrarà una representació gràfica de la simulació com la que hem vist anteriorment. Això pot ser útil per detectar bugs o veure com evoluciona la simulació.

## Descripció del problema.

Amb aquest enunciat us proporcionem la solució seqüencial programada en C pel problema descrit en l'apartat anterior. A vosaltres us toca fer el següent pas, és a dir, afegir les directives OpenACC que considereu adients per tal de que aquest programa s'executi fent servir els recursos de còmput (cores) d'un accelerador. Com a alternativa per a la implementació del la pràctica us oferim utilitzar CUDA enlloc de OpenACC.

Els requeriments que ha de complir la solució que proposeu (tant per a la versió d'OpenACC com per a la de CUDA) són els següents:

- a. La solució ha de continuar sent correcta. El resultat final de la simulació per a la versió paral·lela no s'ha de desviar del resultat per a la versió seqüencial del programa.
- b. Minimitzar sobrecàrregues causades per l'ús de les directives OpenACC. Per simplicitat podeu generar una primera versió paral·lela amb la menor quantitat de canvis possibles, però posteriorment l'analitzareu i la depurareu per minimitzar el nombre de bytes que es mouen entre el host i el dispositiu, el nombre de sincronitzacions necessàries, etc.
- c. Maximitzar el rendiment, definint com a millora de rendiment la reducció del temps d'execució de l'aplicació. Quin és el millor nombre de threads per executar l'aplicació? Val la pena explorar diferents configuracions del grid? Val la pena utilitzar directives per controlar el moviment de les dades? Etc.

**Heu de tenir en compte que l'estratègia a seguir no serà la mateixa que a la pràctica anterior, aquest cop haureu de raonar que parts del codi seria adient que s'executessin dins la GPU.**

## Material

Amb aquest enunciat us proporcionem el codi seqüencial del programa `extinguishing.c` i un arxiu de llicència `LICENSE` per donar crèdit als autors d'aquest programa. Aquests arxius es penjaran al discord i també els trobareu en la carpeta `/home/alumnos/capmc/alumnos/p-OpenACC`.

La recomanació és que creeu una còpia de l'arxiu proporcionat per a cada versió a desenvolupar (ex. `extinguishing_acc.c` / `extinguishing_cuda.cu`) i feu les modificacions que considereu oportunes.

## Lliurament

Pel lliurament d'aquesta pràctica s'haurà d'enviar un correu a `cristina.peralta@uab.cat` que contingui mínim dos arxius per cada grup de pràctiques.

El primer arxiu serà un informe de dos o tres pàgines en el que incloureu:

1. Noms i NIUs dels integrants del grup
2. Breu resum del plantejament del problema (paral·lelització del codi donat fent servir OpenACC i/o CUDA)
3. Anàlisi del problema
4. Disseny de la solució
5. Resultat (s'han assolit els objectius?, proves realitzades, comparació de les versions implementades, errors coneguts, etc.)
6. Principals problemes (què us ha costat més?)

El segon arxiu serà el codi font de la versió desenvolupada (en un arxiu de text, és a dir, que pugui ser compilat i provat) comentat de forma apropiada (comentaris aclaridors d'aquells punts més complexes i generals de les funcions). Si heu fet ambdues versions podeu incloure-les al correu de la pràctica.

Per tal que els temps mesurats siguin vàlids, recordeu que heu de fer les execucions fent servir el gestor de treballs SLURM.

## Referències

- [1] <https://www.ipdps.org/ipdps2022/>
- [2] <https://sc21.supercomputing.org/>
- [3] <https://tcpp.cs.gsu.edu/curriculum/?q=peachy>
- [4] [https://tcpp.cs.gsu.edu/curriculum/?q=system/files/Peachy\\_Eduhpc\\_19.pdf](https://tcpp.cs.gsu.edu/curriculum/?q=system/files/Peachy_Eduhpc_19.pdf)