

Carlos Bederián, Nicolás Wolovick

Objetivos:

- Investigar los problemas de performance de una aplicación no trivial.
- Aplicar conocimientos de optimización secuencial.

El código entregado en `solver.c` implementa una discretización de las ecuaciones de Navier–Stokes para simular dinámicas de flúidos en aplicaciones de tiempo real. Este código es usado por dos front-ends, `demo` (una aplicación OpenGL controlable por el usuario) y `headless` (una aplicación de consola para realizar mediciones). Sin embargo, aunque el algoritmo es veloz la implementación puede mejorarse para brindar mayor performance.

Para ello, deberán:

- Hacer un análisis de las secciones más utilizadas del código utilizando un profiler.
- Optimizar las funciones de mayor utilización utilizando las técnicas vistas en clase, incluyendo vectorización con SSE intrinsics.
- Verificar a través del front-end OpenGL que el código siga funcionando de la manera esperada.

Una vez agotadas las estrategias de optimización secuencial, deberán redactar un pequeño informe (≤ 4 páginas) detallando los cambios realizados (satisfactorios o no), comparando la performance del código compilado con `gcc`, `clang` e `icc` y mostrando una gráfica del escalado del programa para distintos tamaños de matriz entre 32×32 y 1024×1024 .

Extra: Cuando el sistema deja de entrar completo en cache la performance del algoritmo se deteriora rápidamente. Diseñe una versión *cache-aware* o *cache-oblivious* de la función `lin_solve`.