

Chapitre 4 : OpenMP

- 4.1 Un outil de détection de problème de synchronisation vous indique une course dans la section réalisée par votre co-équipier qui affirme que tout est correct. Qui croire?

```
found = 0;

#pragma omp parallel for
for(i = 0 ; i < n ; i++)
{ if(a[i] == key) found = 1; }

if(found) { ... }
```

- 4.2 Votre co-équipier se plaint que votre programme ne profite pas de la performance disponible en raison de la barrière implicite entre les deux boucles. Il veut les mettre `nowait` ou dans des sections. Est-ce correct? Avez-vous mieux à proposer?

```
#pragma omp parallel
{ #pragma omp for
  for(i = 0; i < n; i++)
  { a[i] = f1(i); }

  #pragma omp for
  for(i = 0; i < n; i++)
  { b[i] = a[i] + f2(i); }
}
```

- 4.3 Votre co-équipier a modifié votre programme en ajoutant des `nowait` et obtient un gain de performance appréciable mais ses résultats seront-ils corrects et fiables?

```
#pragma omp parallel
{ #pragma omp for schedule(static) nowait
  for (i=0; i<n; i++) c[i] = (a[i] + b[i]) / 2.0f;

  #pragma omp for schedule(static) nowait
  for (i=0; i<n; i++) z[i] = sqrtf(c[i]);

  #pragma omp for schedule(static) nowait
  for (i=1; i<=n; i++) y[i] = z[i-1] + a[i];
}
```

- 4.4 Votre programme comporte une boucle dont la longueur de chaque itération varie grandement. Discutez de différents mécanismes qui pourraient être utilisés pour assurer néanmoins une bonne parallélisation.