



Estrutura do tema Avaliação de Desempenho (IA-32)

1. A avaliação de sistemas de computação
2. Técnicas de otimização de código (IM)
3. Técnicas de otimização de *hardware*
4. Técnicas de otimização de código (DM)
5. Outras técnicas de otimização
6. Medição de tempos



Análise de técnicas de otimização (s/w)

- técnicas de otimização de código (indep. máquina)
 - *já visto...*
- técnicas de otimização de código (dep. máquina)
 - dependentes do processador (*já visto...*)
- **outras técnicas de otimização**
 - na compilação: otimizações efectuadas pelo GCC
 - na identificação dos "gargalos" de desempenho
 - *code profiling*
 - uso dum *profiler* para apoio à otimização
 - **lei de Amdahl**
 - dependentes da hierarquia da memória (*a ver na próxima UC*)
 - a localidade espacial e temporal dum programa
 - influência da *cache* no desempenho

**Code profiling:
análise visual da melhoria de código numa função**



**Antes,
96% na função
ProportionalRegionGrow**

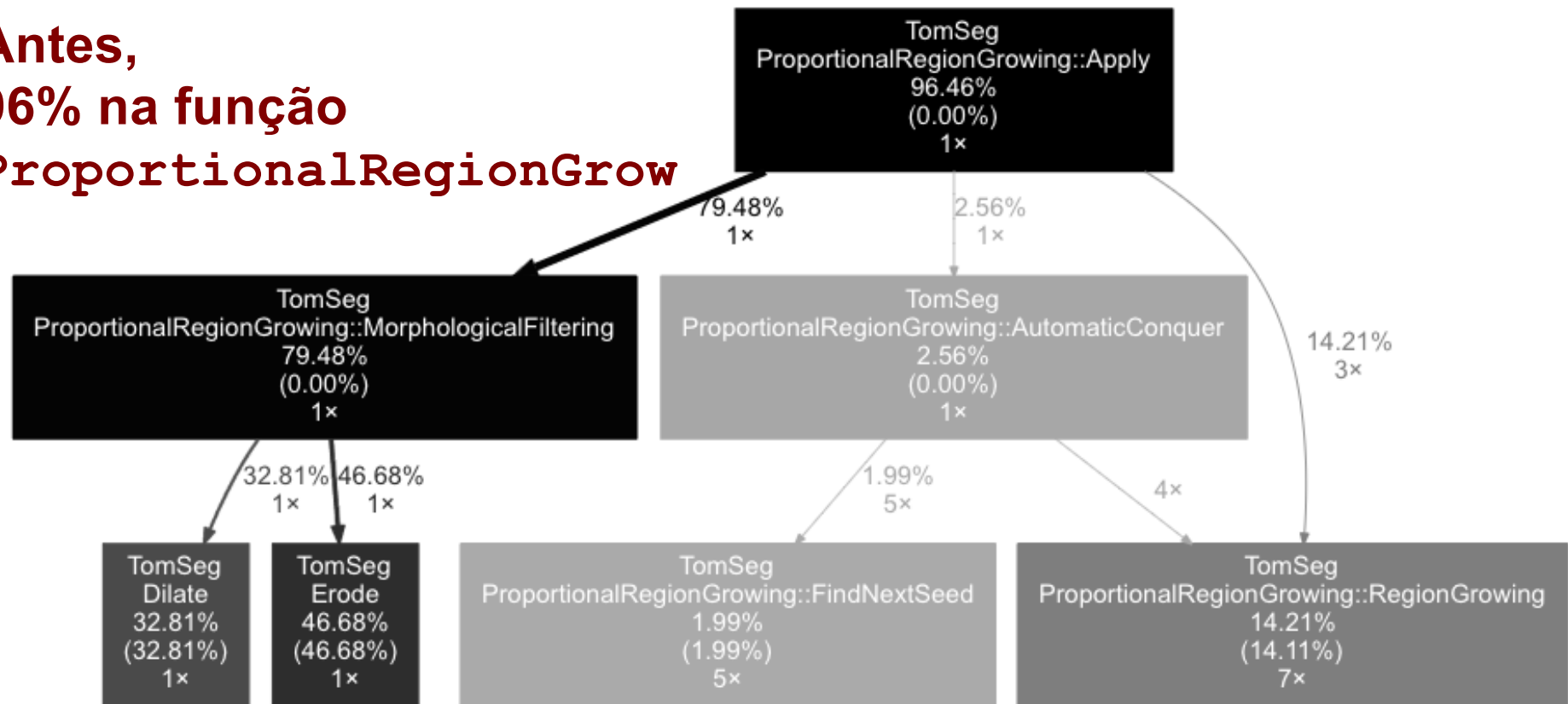


Figure 5.7.: Call-graph of the first version of *Proportional Region Growing* (DS₃)

**Code profiling:
análise visual da melhoria de código numa função**



**Depois,
apenas 36% na
mesma função!**

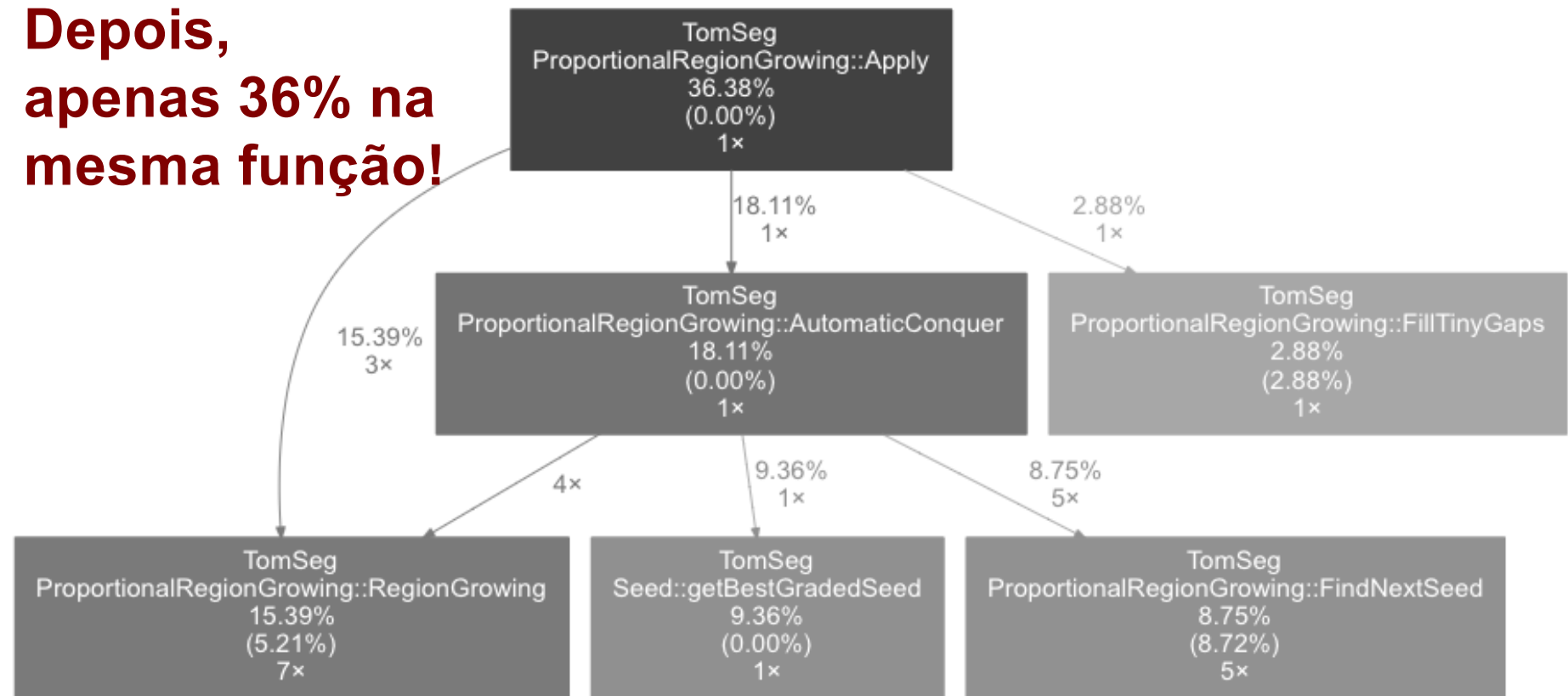


Figure 5.9.: Call-graph from the last version of *Propor. Region Growing* (DS3)

Lei de Amdahl



O ganho no desempenho – *speedup* – obtido com a melhoria do tempo de execução de uma parte do sistema, está limitado pela fração de tempo que essa parte do sistema pode ser usada.

$$\text{Speedup}_{\text{overall}} = \frac{\text{Tempo_exec}_{\text{antigo}}}{\text{Tempo_exec}_{\text{novo}}} = \frac{1}{\sum (f_i / s_i)}$$

f_i - fracções com melhoria s_i
 s_i - *speedup* de cada fracção

Ex.1: Se 10% de um prog executa 90x mais rápido, então

Overall speedup = 1.11

Ex.2: Se 90% de um prog executa 90x mais rápido, então

Overall speedup = 9.09

Paralelismo:
se $N_{\text{proc}} \equiv \text{speedup}$, trocar s_i por N_{proc}

