Abstract
Performance models are mostly simple mathematical formulas. The challenge is to model the performance at an appropriate level. In this presentation we introduce several levels of modeling, labeled zeroth order, second order, et cetera. As illustration we use the performance of MRI reconstruction.
Theory Block: n Order Formulas

0\textsuperscript{th} order
- main function parameters
- order of magnitude relevant for main function

1\textsuperscript{st} order
- add overhead secondary function(s)
- estimation

2\textsuperscript{nd} order
- interference effects circumstances
- main function, overhead and/or secondary functions
- more accurate, understanding
CPU Time Formula Zero Order

\[ t_{\text{cpu total}} = t_{\text{cpu processing}} + t_{\text{UI}} \]

\[ t_{\text{cpu processing}} = n_x \times n_y \times t_{\text{pixel}} \]
CPU Time Formula First Order

\[ t_{\text{cpu total}} = t_{\text{cpu processing}} + t_{\text{UI}} + t_{\text{context switch}} + \text{overhead} \]
CPU Time Formula Second Order

\[ t_{\text{cpu total}} = t_{\text{cpu processing}} + t_{\text{UI}} + t_{\text{context switch overhead}} + \]

\[ t_{\text{stall time due to}} + t_{\text{stall time due to}} \]

cache efficiency context switching

signal processing: high efficiency
control processing: low/medium efficiency
"Test" of performance model on another case
Scope of performance and significance of impact
MR Reconstruction Context

Host

Acquisition
- control
  - magnet
  - gradients
  - RF
  - ADC

Reconstruction

Storage

Viewing & Printing

Formula Based Performance Design
©2006, Embedded Systems Institute
©2006, Gerrit Muller
\[ t_{\text{recon}} = n_{\text{raw-x}} \cdot t_{\text{fft}}(n_{\text{raw-y}}) + n_y \cdot t_{\text{fft}}(n_{\text{raw-x}}) + t_{\text{fft}}(n) = c_{\text{fft}} \cdot n \cdot \log(n) \]
Zero Order Quantitative Example

Typical FFT, 1k points ~ 5 msec
( scales with $2 \times n \times \log(n)$ )

using:

$n_{\text{raw-x}} = 512$

$t_{\text{recon}} = n_{\text{raw-x}} \times t_{\text{fft}}(n_{\text{raw-y}}) + n_{\text{raw-y}} \times t_{\text{fft}}(n_{\text{raw-x}}) +$ 512 * 1.2 + 256 * 2.4

$n_{\text{raw-y}} = 256$

$n_x = 256$

$n_y = 256$

$\approx 1.2$ s
t_{recon} = t_{filter}(n_{raw-x}, n_{raw-y}) + n_{raw-x} \cdot t_{fft}(n_{raw-y}) + n_y \cdot t_{fft}(n_{raw-x}) + t_{corrections}(n_x, n_y)

t_{fft}(n) = c_{fft} \cdot n \cdot \log(n)
First Order Quantitative Example

Typical FFT, 1k points ~ 5 msec
   ( scales with 2 * n * log (n) )

Filter 1k points ~ 2 msec
   ( scales linearly with n )

Correction ~ 2 msec
   ( scales linearly with n )
MR Reconstruction Performance Second Order

\[
\begin{align*}
    t_{\text{recon}} &= t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) \\
    &+ n_{\text{raw-x}} \cdot (t_{\text{fft}}(n_{\text{raw-y}})) \\
    &+ n_y \cdot (t_{\text{fft}}(n_{\text{raw-x}})) \\
    &+ t_{\text{corrections}}(n_x, n_y) \\
    &+ t_{\text{control-overhead}} \\
    t_{\text{fft}}(n) &= c_{\text{fft}} \cdot n \cdot \log(n)
\end{align*}
\]
Second Order Quantitative Example

Typical FFT, 1k points \(~ 5\) msec
\((\text{scales with } 2 \times n \times \log(n))\)

Filter 1k points \(~ 2\) msec
\((\text{scales linearly with } n)\)

Correction \(~ 2\) msec
\((\text{scales linearly with } n)\)

Control overhead = \(n \times t_{\text{row overhead}}\)
\(10 \ldots 100\ \mu\text{s}\)
MR Reconstruction Performance Third Order

\[ t_{\text{recon}} = t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + n_{\text{raw-x}} \cdot (t_{\text{fft}}(n_{\text{raw-y}}) + t_{\text{col-overhead}}) + n_{\text{y}} \cdot (t_{\text{fft}}(n_{\text{raw-x}}) + t_{\text{row-overhead}}) + t_{\text{corrections}}(n_{x}, n_{y}) + t_{\text{read I/O}} + t_{\text{transpose}} + t_{\text{write I/O}} + t_{\text{control-overhead}} \]

Focus on overhead reduction is more important than faster algorithms. This is not an excuse for sloppy algorithms.
MRI reconstruction

System performance may be determined by other than standard facts
E.g. more by overhead I/O rather than optimized core processing

==> Identify & measure what is performance-critical in application
The ASP™ course is partially derived from the EXARCH course developed at Philips CTT by Ton Kostelijk and Gerrit Muller.

Extensions and additional slides have been developed at ESI by Teun Hendriks, Roland Mathijssen and Gerrit Muller.