A simple Concept for the Performance Analysis of Cluster-Computing

H. Kredel[1], S. Richling[2], J.P. Kruse[3], E. Strohmaier[4], H.G. Kruse[1]

[1] IT-Center, University of Mannheim, Germany
[2] IT-Center, University of Heidelberg, Germany
[3] Institute of Geosciences, Goethe University Frankfurt, Germany
[4] Future Technology Group, LBNL, Berkeley, USA

ISC'13, Leipzig, 18. June 2013

Outline

Introduction

Motivation

- Sophisticated mathematical models for performance analysis cannot keep up with rapid hardware development.
- There is a lack of reliable rules of thumb to estimate the size and performance of clusters.

Goals

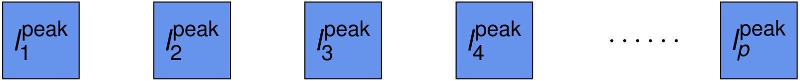
- Development of a simple and transparent model.
- Restriction to few parameters describing hardware and software.
- Using speed-up as a dimensionless metric.
- Finding the optimal size of a cluster for a given application.
- Validation of the results by modeling of standard kernels.

Related Work

- Roofline model for multi-cores (Williams et al. 2009)
- Performance models by Hockney:
  - Model with few hardware and software parameters, focus on benchmark runtimes and performance (Hockney 1987, Hockney & Jesshope 1988)
  - Model based on similarities to fluid dynamics (Hockney 1995)
- Performance models by Numrich:
  - Based on Newtons classical mechanics (Numrich 2007)
  - Based on dimension analysis (Numrich 2008)
  - Based on the Pi theorem (Numrich 2010)
- Linpack performance model (Luszczek & Dongarra 2011)
- Performance model based on a stochastic approach (Kruse 2009, Kredel et al. 2010)
- Performance model for interconnected clusters (Kredel et al. 2012)

Model Parameters

Hardware Parameters



p	number of processing units (PUs)
$l_{k=1,p}^{peak}$	theoretical peak performance of each PU
b_c	bandwidth of the network

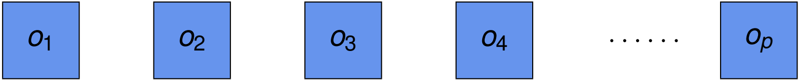
Software Parameters

$\#op$	total number of arithmetic operations
$\#b$	total number of bytes involved
$\#x$	total number of bytes communicated between the PUs

Distribution of the work load ($\#op$,$\#b$)

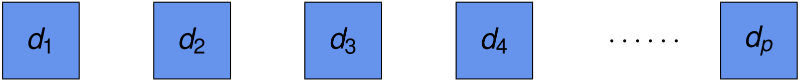
Homogeneous case

- Distribution of operations $\#op$



$$o_k = \#op/p \qquad (\text{or} \quad \omega_k = 1/p)$$

- Distribution of data $\#b$



$$d_k = \#b/p \qquad (\text{or} \quad \delta_k = 1/p)$$

Distribution of the work load ($\#op$,$\#b$)

Heterogeneous case $\rightarrow$ additional parameters (ω_k,δ_k)

- Distribution of operations $\#op$

o_1 o_2 o_3 o_4 $\cdots\cdots$ o_p

$$o_k = \omega_k \cdot \#op \quad \text{with} \quad \sum_{k=1}^{p} \omega_k = 1$$

- Distribution of data $\#b$

d_1 d_2 d_3 d_4 $\cdots\cdots$ d_p

$$d_k = \delta_k \cdot \#b \quad \text{with} \quad \sum_{k=1}^{p} \delta_k = 1$$

Performance Indicators

Primary performance measure

t Total time to process the work load $(\#op, \#b)$

Derived performance measures

$I(p) = \frac{\#op}{t}$ Performance

$S = \frac{I(p)}{I(1)}$ Speed-up (dimensionless)

Goal: Speed-up as a function of

- total work load $(\#op, \#b)$ $[Flop, Byte]$
- work distribution (ω_k, δ_k)
- communication requirements $\#x$ [Byte]
- hardware parameters $(p, I_k^{\text{peak}}, b_c)$ [-,Flop/s, Byte]

Total execution time

Computation time

$$t^r = \max\{\, t_1(o_1, d_1), \ldots, t_n(o_p, d_p)\,\} \simeq \frac{o_k}{l_k} \geq \frac{o_k}{l_k^{\text{peak}}}$$

Communication time

$$t^c \simeq \frac{\#x}{b_c}$$

Total execution time

$$t \simeq t^r + t^c$$

$$t \geq \frac{o_k}{l_k^{\text{peak}}} + \frac{\#x}{b_c}$$

Total execution time

$$t \geq \omega_k \cdot \frac{\#op}{l_k^{\text{peak}}} + \frac{\#x}{b_c} = \omega_k \cdot \frac{\#op}{l_k^{\text{peak}}} \cdot \left(1 + \frac{l_k^{\text{peak}}}{b_c} \cdot \frac{\#b}{\omega_k \#op} \cdot \frac{\#x}{\#b}\right)$$

$$t \geq \omega_k \cdot \frac{\#op}{l_k^{\text{peak}}} \cdot \left(1 + \frac{1}{x_k}\right)$$

One dimensionless parameter for “hardware + software”

$$x_k = \omega_k \cdot \frac{a}{a_k^*} \cdot r$$

$a = \frac{\#op}{\#b}$ computational intensity of the software [Float/Byte]

$a_k^* = \frac{l_k^{\text{peak}}}{b_c}$ ”computational intensity” of the hardware [Float/Byte]

$r = \frac{\#b}{\#x}$ ”inverse communication intensity” [-]

Performance and Speed-up

Performance

$$I = \frac{\#op}{t} \leq \frac{I_k^{\text{peak}}}{\omega_k} \cdot \frac{x_k}{1 + x_k}$$

Speed-up

$$S = \frac{I(p)}{I(1)} = \frac{I_k(\omega_k < 1)}{I_k(\omega_k = 1)} = \frac{1 + x_k(\omega_k = 1)}{1 + \omega_k \cdot x_k(\omega_k = 1)}$$

$$x_k(\omega_k = 1) = \frac{a}{a_k^*} \cdot r = a \cdot \frac{b_c}{I_k^{\text{peak}}} \cdot r = a \cdot \frac{b_c^0}{I_k^{\text{peak}}} \cdot \frac{b_c}{b_c^0} \cdot r = \hat{x}_k \cdot z \cdot r$$

$$S = \frac{1 + \hat{x}_k \cdot r \cdot z}{1 + \omega(k,p) \cdot \frac{\hat{x}_k \cdot r \cdot z}{p}}$$

general case with $\omega_k = \omega(k,p)/p$

$$S = \frac{1 + \hat{x} \cdot r \cdot z}{1 + \frac{\hat{x} \cdot r \cdot z}{p}}$$

homogeneous case with $\omega(k,p) = 1$

Application-oriented Analysis

Application characterized by problem size n.

Software Parameters

$\#op \to \#op(n)$ $\#b \to \#b(n)$ $\#x \to \#x(n,p)$

Analysis of the performance of a homogeneous cluster

$$l \leq p\, l^{\text{peak}} \frac{x}{x+1} = l^{\text{peak}} y \cdot \frac{r(n,p)}{1 + y\frac{r(n,p)}{p}}$$

With $x = \hat{x} \cdot z \cdot r(n,p)/p = y \cdot r(n,p)/p \simeq y \cdot \frac{c(n)}{d(p)}\frac{1}{p}$

- Number of PUs $p_{1/2}$ necessary to reach half of the maximum performance of all p PUs.

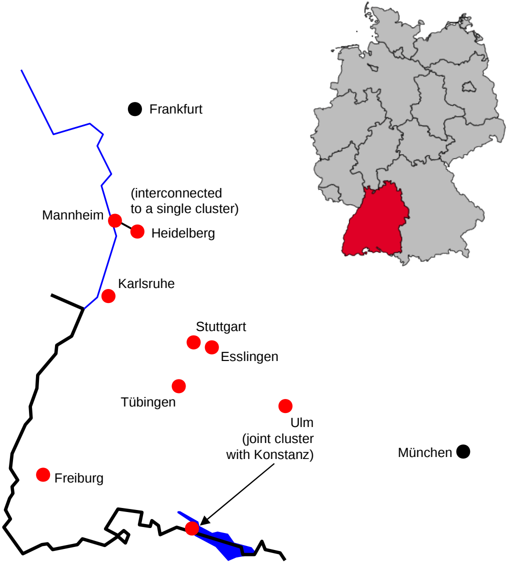
  $l(p_{1/2}) = \frac{1}{2} p l^{peak} \to y \cdot r(n, p_{1/2}) = p_{1/2}$
- Number of PUs p to obtain the maximum of the performance

  $\frac{dl}{dp} = 0 \to p_{\max}^2 \cdot d'(p_{\max}) = y = \hat{x} \cdot z \cdot c(n)$

Compute resources for the simulations

bwGRiD Cluster

Site	Nodes
Mannheim	140
Heidelberg	140
Karlsruhe	140
Stuttgart	420
Tübingen	140
Ulm/Konstanz	280
Freiburg	140
Esslingen	180
Total	1580



bwGRiD

Bundesministerium
für Bildung
und Forschung

Baden-Württemberg

bwGRiD – Hardware

Node Configuration

- 2 Intel Xeon CPUs, 2.8 GHz (each CPU with 4 Cores)
- 16 GB Memory
- 140 GB hard drive (since January 2009)
- InfiniBand Network (20 Gbit/sec)

Hardware parameters for our model

$$
\begin{aligned}
l^{\mathrm{peak}} &= 8 && \text{GFlop/sec (for one core)} \\
b_c &= 1.5 && \text{GByte/sec (node-to-node)} \\
b_c^0 &= 1.0 && \text{GByte/sec (reference bandwidth)}
\end{aligned}
$$

Scalar-Product of two Vectors

$$(u, v) = \sum_k u_k \cdot v_k$$

Software Parameters

$$\#op = 2\,n - 1 \simeq 2n \text{ if } n \gg 1$$
$$\#b = 2\,n\,w$$
$$\#x = p\,w = 8p$$

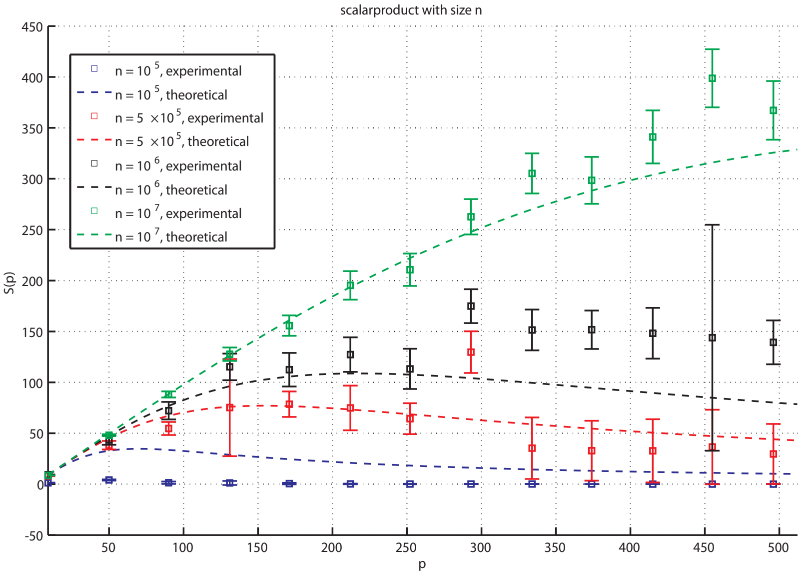
Speed-up

$$S = \frac{1 + x}{1 + x/p} \qquad \text{with } x = \tfrac{3}{64} \cdot \tfrac{n}{p}$$

Simulations

- Vector sizes up to $n = 10^7$
- 20 runs for each configuration (p, n)
- Speed-up calculated from mean run-times

Speed-up for Scalar Product



Matrix Multiplication

$A^{n \times n} \cdot B^{n \times n} = C^{n \times n}$ on a $\sqrt{p} \cdot \sqrt{p}$ processor-grid

Software Parameters

$$\#op = 2n^3 - n^2 \simeq 2n^3$$
$$\#b = 2n^2 w$$
$$\#x = 2n^2 \sqrt{p}(1 - \tfrac{1}{\sqrt{p}}) w \simeq 2n^2 w \sqrt{p}$$

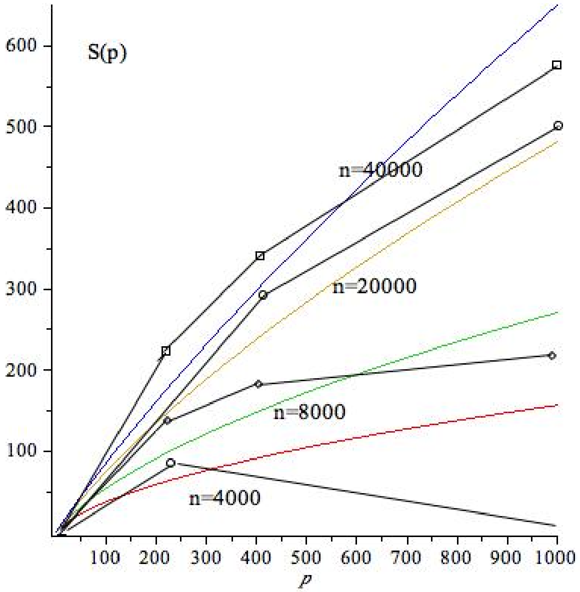
Speed-up

$$S = \frac{1 + x}{1 + x/p} \qquad \text{with } x = \tfrac{3}{2048} n \sqrt{p}$$

Simulations

- Matrix sizes up to $n = 40000$
- Cannon's algorithm
- Runs with 8 and 4 cores per node

Speed-up for Matrix Multiplication



Linpack

Solution of $Ax = b$

Software Parameters

$$\#op = \tfrac{2}{3}n^3$$
$$\#b = 2n^2 \cdot w$$
$$\#x = 3\alpha \left(1 + \tfrac{\log_2 p}{12}\right) n^2 \cdot w$$

Speed-up

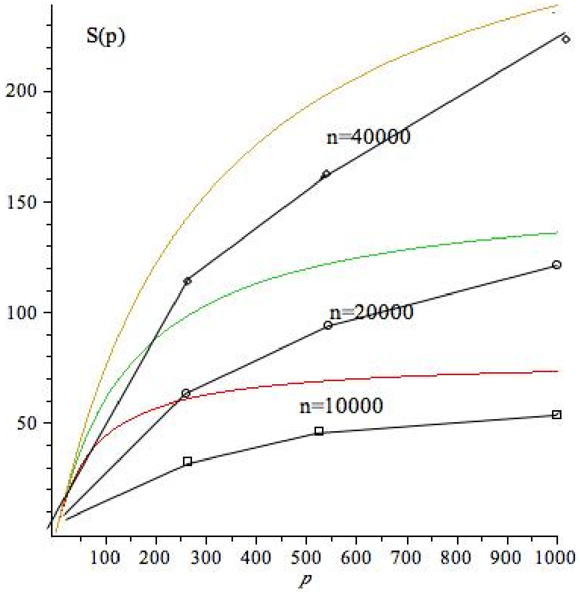
$$S \sim \frac{1 + x}{1 + x/p}$$

with $x = \frac{n}{128}$ and $\alpha = 1/3$

Simulations

- Matrix sizes up to 40000.
- Smaller α would lead to better fits for small p.

Speed-up for Linpack



Linpack on bwGRiD

Half of Peak performance at:

$$p_{1/2} = \frac{y}{3\alpha} = \frac{n}{128}$$

Maximum performance at:

$$p_{\max} = (24 \cdot \ln 2/128) \cdot n = 24 \ln(2) p_{1/2}$$

Region with 'good' performance for $n = 10000$

$$p = [p_{1/2}, p_{\max}] = [80, 1300]$$

Maximum performance

$$l_{\max} =\sim \frac{l^{\text{peak}} y}{3\alpha} \frac{9}{10}$$

$l_{\max} = 560$ GFlop/sec for $n = 10000$

TOP500

Maximum performance

$$l_{\max} = \frac{n \cdot b_c}{3w} \frac{9}{10}$$

In TOP500 list: $l_{\max} \rightarrow R_{\max}$ and $n \rightarrow N_{\max}$
Bandwidth b_c not in the list.

Derive Effective Bandwidth

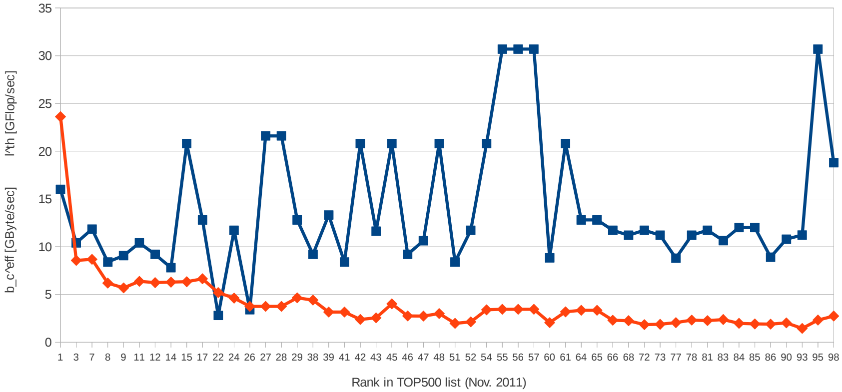
$$b_c^{\text{eff}} = \frac{R_{\max}}{N_{\max}} \cdot 3w \cdot \frac{10}{9}$$

Analyze which parameter predicts ranking best

- first 100 systems
- excluding systems with accelerators and missing $N_{\max}$
- comparison with single core performance $l^{\text{peak}} = R_{\max}/p_{\max}$

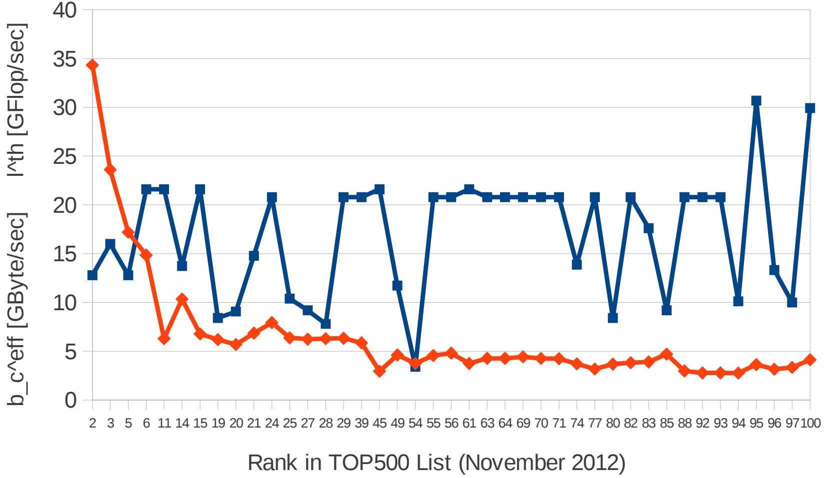
TOP500 – November 2011

Blue: Linpack-Performance per core
Red: Derived effective Bandwidth



TOP500 – November 2012

Blue: Linpack-Performance per core
Red: Derived effective Bandwidth



Conclusions

- Developed a performance model which integrates the characteristics of hardware and software with a few parameters.
- Model provides simple formulae for performance and speed-up.
- Results compare reasonably well with simulations of standard applications.
- Model allows estimation of the optimal size of a cluster for a given class of applications.
- Model allows estimation of the maximum performance for a given class of applications.
- Identified effective bandwidth as a key performance indicator for Linpack (TOP500) on compute clusters.
- Future work:
  - Analysis of inhomogeneous clusters with asymmetric load distribution
  - Further applications: Sparse matrix-vector operations and FFT