A hierarchical Model for the Analysis of Efficiency and Speed-up of Multi-Core Cluster-Computers

H. Kredel[1], H. G. Kruse[1] *retired*, S. Richling[2]

[1] IT-Center, University of Mannheim, Germany
[2] IT-Center, Heidelberg University, Germany

3PGCIC-2015, Krakow, Poland, 4.-6. November 2015

Outline

Introduction

Clusters with multi-core nodes
- Multi-core node performance
- Cluster of Multi-core nodes performance
- Dependency of the performance on q, p and x

Applications

Summary and Future Work

Introduction

Motivation

- We see a widening gap between theory and practice in performance analysis.
- Focus on simple models and mathematical methods.
- Try to identify a few key parameters describing main aspects of a computing system and algorithms.
- Device a simple hierarchical model for clusters assembled from multi-core nodes connected by a (high-speed) network.
- Applicable from compute clusters to clouds for big data analysis.
- Few, but not too few parameters (TOP500), now emerging multiparameter studies (HPCG, GREEN500, GRAPH500).

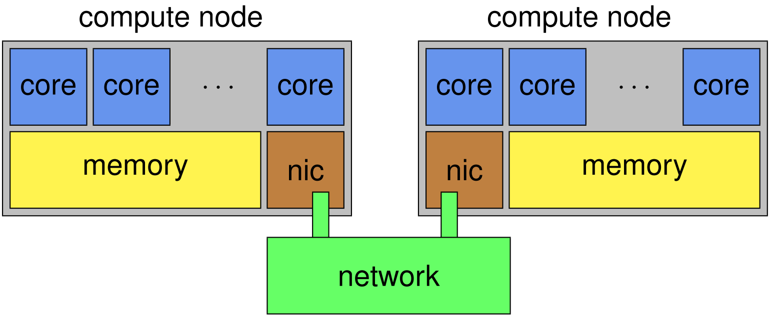
Goals

- Practical performance analysis of computer systems insight, evaluation and assessment for
  - computer design, planing of new systems, acquiring performance data, estimation of run-times.
- Using speed-up, efficiency and operations per time unit as dimensionless metric.
- Key hardware parameters
  - number of compute nodes, number of cores per node,
  - theoretical performance of a core,
  - bandwidth between cores and memory, and between nodes
- Key software parameters
  - number of bytes, number of operations,
  - number of bytes communicated.
- Validation of the results by modeling of standard kernels
  - scalar product, matrix multiplication, solution of linear equations (Linpack), Fast Fourier transformation (FFT).

Related Work

- Early performance models:
  - distinguished computation and communication phases, introduced speed-up and efficiency (Hockney 1987, Hockney & Jesshope 1988)
- Performance models with dimensionless parameters:
  - analogies to Newtons classical mechanics or electrodynamics (Numrich 2007, 2015)
  - dimension analysis and the Pi theorem (Numrich 2008, 2010)
- Linpack performance prediction model (Luszczek & Dongarra 2011)
- Performance models based on stochastic approaches (Gelenbe 1989, Kruse 2009, Kredel et al. 2010)
- Performance models for interconnected clusters (Kredel et al. 2012, 2013)
- Roofline model for multi-cores (Williams et al. 2009)

Basic concepts and assumptions



Hardware scheme

- p compute nodes
- q cores per compute node
- memory per compute node
- network interface per compute node
- nodes connected by a (fast) network

Hardware parameters

- p, number of nodes
- l_m, node performance [GFLOP/sec]
- b_{cl}, bandwidth between two nodes [GB/sec]
- q, number of cores per node
- l_c, core performance [GFLOP/sec]
- b_m, bandwidth between cores (caches) and memory [GB/sec]

Software parameters

- $\#op$, number of arithmetic operations per application problem
- $\#b$, number of bytes per application problem
- $\#x$, number of bytes exchanged between nodes per application problem

Performance

- t, computing time for a problem on a given system
- $I(q, p, \ldots) = \frac{\#op}{t}$, **performance** in terms of the parameters
- $\eta(q, p, \ldots) = \frac{I(q,p,\ldots)}{q\, p\, I_c}$, **efficiency** as a measure of how well the application uses its compute resources
- $S(q, p, \ldots) = \frac{I(q,p,\ldots)}{I_c}$, **speed-up** as a measure of how well the application scales with varying core and node numbers

efficiency and speed-up give insights

- what is the optimal number of cores and nodes for a given (or future) application on a given (or future) hardware ?
- use these optima as parameters for the batch system on a compute cluster to allocate the right number resources:

  determination of the right number of cores on clusters operated by sharing nodes between jobs

Multi-core node performance

computation time on one node

$$t \;\geq\; \frac{(\#op/q)}{l_c} + \frac{\#b}{b_m} = \frac{\#op}{q\, l_c}\left(1 + q \cdot \frac{l_c}{b_m}\frac{\#b}{\#op}\right) \qquad (1)$$

assume communication with the shared memory and the computation phases *do not overlap*

define $a = \frac{\#op}{\#b}$, $a^\star = \frac{l_c}{b_m}$

$$t \;\geq\; \frac{\#op}{q\, l_c}\left(1 + q\,\frac{a^\star}{a}\right)$$

a “software and problem demand”, $a^\star$ “hardware capabilities” performance of a node $l_m = \#op/t$

$$l_m \leq q\, l_c \frac{1}{1 + q \cdot \frac{a^\star}{a}} = q\, l_c \frac{\frac{a/a^\star}{q}}{1 + \frac{a/a^\star}{q}}$$

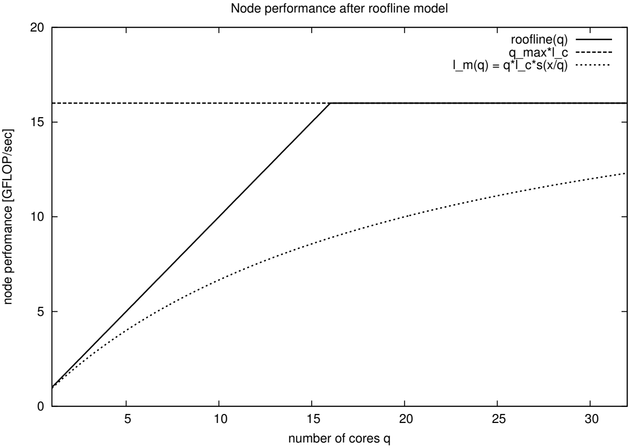
with dimensionless operational intensity $x = a/a^\star$

$$l_m(q) \leq q\, l_c \frac{\frac{x}{q}}{1 + \frac{x}{q}}.$$

with Hockney $s(z) = \frac{z}{1+z}$, overlapping $s(z) = \min(1, z)$

$$l_m(q) \leq q\, l_c\, s(\frac{x}{q}). \qquad (2)$$

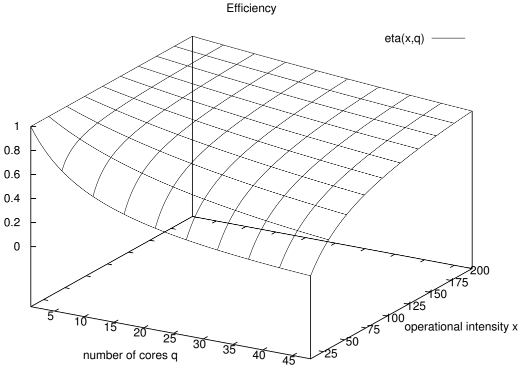
Performance of multi-cores in the roofline model for $l_c = 0.5$ GFLOP/sec and operational intensity $x = 20$



Efficiency of multi-cores depending on x and q

$$\eta_m(x, q) = \frac{l_m(q)}{q\, l_m(1)} \quad \leq \quad \frac{1}{q}\, \frac{1 + x}{1 + \frac{x}{q}}$$

Amdahl's law: along x-axis, Gustafson's law: along y-axis



Cluster of Multi-core nodes performance

computation time of all nodes

$$t \geq \frac{(\#op/p)}{l_m(q)} + \frac{\#x}{b_{cl}(q,p)} \tag{3}$$

Bandwidth $b_{cl}(q,p)$ between the nodes of the cluster choosen as $b_{cl}(q,p) = \beta(q,p)\, b_{cl} = \frac{\beta_1(p)}{q} b_{cl}$, where b_{cl} is a constant.

using $\beta(q,p)$ and sorting the expression

$$\frac{t}{\#op} \geq \frac{1}{p\, l_m} \left(1 + \frac{p\, l_m}{\beta(q,p) b_{cl}} \cdot \frac{\#x}{\#op} \right)$$

using the transformations

$$\frac{p\, l_m}{\beta(q,p) b_{cl}} \frac{\#x}{\#op} \quad = \quad \frac{q\, p}{\beta(q,p)} \frac{l_c}{b_m} s(\frac{x}{q}) \frac{b_m}{b_{cl}} \frac{\#b}{\#op} \frac{\#x}{\#b}$$

already known definitions of a, $a^\star$ and x and the new definitions of r and v

$$a = \frac{\#op}{\#b},\ a^\star = \frac{l_c}{b_m},\ r = \frac{\#b}{\#x},\ v = \frac{b_{cl}}{b_m},\ x = \frac{a}{a^\star}.$$

r defines the ratio of total number of bytes $\#b$ to the number of exchanged bytes between the nodes $\#x$, and v defines the ratio of the bandwidth in the cluster network b_{cl} to the memory bandwidth in a node b_m.

performance $l_{cl}(q,p)$ of the whole cluster

$$l_{cl}(q,p) \quad \leq \quad q\,p\,l_c \cdot \frac{s(\frac{x}{q})}{1 + \frac{q^2\,p}{\beta(q,p)} \cdot \frac{s(\frac{x}{q})}{v\,r\,x}}. \tag{4}$$

efficiency $\eta_{cl}(q,p) = \frac{l_{cl}(q,p)}{q\,p\,l_c}$

$$\eta_{cl}(q,p) \quad \leq \quad \frac{s(\frac{x}{q})}{1 + \frac{q^2\,p}{\beta(q,p)} \cdot \frac{s(\frac{x}{q})}{v\,r\,x}} \tag{5}$$

speed-up $S(q,p) = \frac{l_{cl}(q,p)}{l_c} = q\,p\,\eta_{cl}(q,p)$

Dependency of the performance on q, p and x

assumptions

The ratio $v = \frac{b_{cl}}{b_m}$ of the bandwidths will be chosen as 0.25 by current hardware.
The application parameter $r(x, q, p)$ depends on hard- and software, assume $r(x, q, p) = \frac{c(x)}{d(p)}$ where $c(x)$, $d(p)$ are monotone increasing functions of their arguments.
The interesting cases are $c(x) = c_0$, $c(x) = c_1 x$ and $d(p) = d_1 p$, $d_2 \sqrt{p}$, $d_3 \log_2 p$

cases considered

- $\mathbf{q} = \mathbf{p} = \mathbf{1}$: $\eta < x/(1 + x) \leq 1$
- $\mathbf{q} \geq \mathbf{1}$ fixed, $\mathbf{p} \gg \mathbf{1}$ resp. $\mathbf{p} \to \infty$: $\eta \sim 0$,
- $\mathbf{q} \gg \mathbf{1}$, $\mathbf{p}$ fixed: $\eta \sim 0$.

efficiency $\eta(q,p)$

The derivatives $(\frac{\partial\eta}{\partial q}, \frac{\partial\eta}{\partial p})$ show no extrema.
$\eta_{cl}(q,p,x)$ is monotone decreasing with increasing arguments (q,p). This behaviour is due to Amdahl's Law.
The translation of the efficiency-surface with increasing load x reflects Gustafson's Law.

$S(q,p,x)$ maximum for q: $\frac{\partial S(q,p,x)}{\partial q} = 0$

$$q_E^3 = \frac{\beta_1(p)}{2p} v\, r(x,p)\, x$$

$S(q,p,x)$ maximum for p: $\frac{\partial S(q,p,x)}{\partial p} = 0$

$$p_E^2 \frac{\partial}{\partial p}\left(\frac{1}{\beta_1(p) r(x,p)}\right)\Bigg|_{p=p_E} = v\frac{x+q}{q^3}.$$

reasonable solution for $\beta_1(p) = \beta_0$, β_0 constant

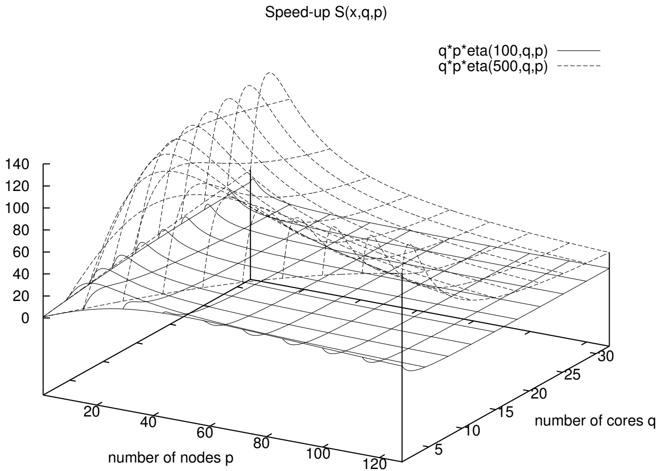
$$p_E^2 \cdot d'(p) \quad = \quad \frac{\beta_0}{q^2} w(x), \text{ with } w(x) = v\frac{x}{q}c(x).$$

Optimal values p_E for communication $d(p)$

Values of p_E for different functions $d(p)$ increasing from left to right

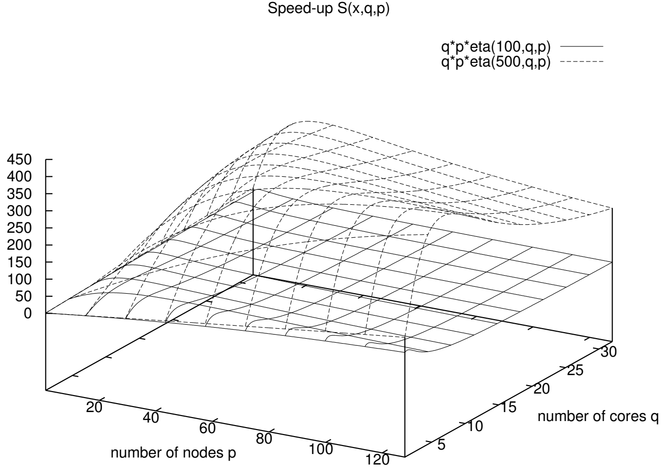
	$d(p) = p$	$d(p) = \sqrt{p}$	$d(p) = \log_2(p)$
p_E	$\sqrt{\frac{\beta_0}{q^2} w(x)}$	$(2\frac{\beta_0}{q^2} w(x))^{\frac{3}{2}}$	$\frac{\beta_0}{q^2} w(x) \cdot \ln(2)$

Speed-up depending on $\beta_0 = 1$ and $d(p) = p$



Note the existence of an optimal number of cores q_E and nodes p_E

Speed-up depending on $\beta_0 = 1$ and $d(p) = \sqrt{p}$.



Note the existence of an optimal number of cores q_E and nodes p_E

Applications

Verifying the model

with real life compute systems and some important HPC codes and kernels.

- scalar product of vectors,
- matrix-matrix multiplication,
- high performance Linpack,
- fast Fourier transformation.

efficiency $\eta(q,p)$

Use characteristic hardware parameters and characteristic values for applications. Simplification of eq (5) using $s(z) = z/(1+z)$

$$\eta(q,p) \quad \leq \quad \frac{\beta(q,p)\, r\, v\, x}{\beta(q,p)\, r\, v\, x + r\, v\, q + q^2 p}$$

Characteristic hardware parameters

l_c in [GFLOP/sec], b_m in [GB/sec], b_{cl} in [GB/sec], $a^\star = l_c/b_m$ in [FLOP/B], $v = b_{cl}/b_m$

system	l_c	b_m	b_{cl}	$a^\star$	v	q
bwGRiD	8.5	6	1.4	1.41	0.233	≤ 8
bwUniCluster	15.4	77	5.4	0.20	0.070	≤ 16
bwForHLR 1	19.1	95	5.4	0.20	0.056	≤ 20
bwForCluster	33	89	3	0.37	0.033	≤ 16

the (Intel) processores are:
bwGRiD E5440, 2.83 GHz
bwUniCluster E5-2670, 2.6 GHz
bwForHLR 1 E5-2670v2, 2.5 GHz
bwForCluster E5-2630v3, 2.4 GHz

the parameters can be estimated using the following benchmarks:
l_c Linpack DGEMM
b_m STREAM aggregated triade
b_{cl} MPI bandwidth, 1.4 (DDR), 3 (QDR), 5.4 (FDR)

Characteristic values for applications

apps = applications, s prod = scalar product, mm = matrix matrix multiplication, lin eq = linear equations (Linpack), FFT = 2-dim FFT, FFTW = 2-dim FFTW

apps	$\#b$	$\#op$	$\#x$	a	r
s prod	$2nw$	$2n-1$	pw	$\frac{1}{w}$	$\frac{2n}{p}$
mm	$2n^2w$	$2n^3-n^2$	$2n^2\sqrt{p}w$	$\frac{n}{w}$	$\frac{1}{\sqrt{p}}$
lin eq	$2n^2w$	$\frac{2}{3}n^3$	$3\alpha\gamma n^2 w$	$\frac{n}{3w}$	$\frac{3}{\gamma}$
FFT	n^2c	$2n^2\log_2(n)$	$\frac{n^2}{p}\log_2(p)c$	$\frac{2\log_2(n)}{c}$	$\frac{p}{\log_2(p)}$
FFTW	”	”	$n\log_2(p)c$	”	$\frac{n}{\log_2(p)}$

$a = \frac{\#op}{\#b}$ in [FLOP/B], $r = \frac{\#b}{\#x}$, $\alpha \sim 1/3$, $\gamma = (1 + \frac{\log_2 p}{12})$.

$w = 8$ bytes is the size of a `double` and $c = 2w$ bytes is the size of a `complex double`.

Hardware parameters from row bwGRiD.

Scalar product

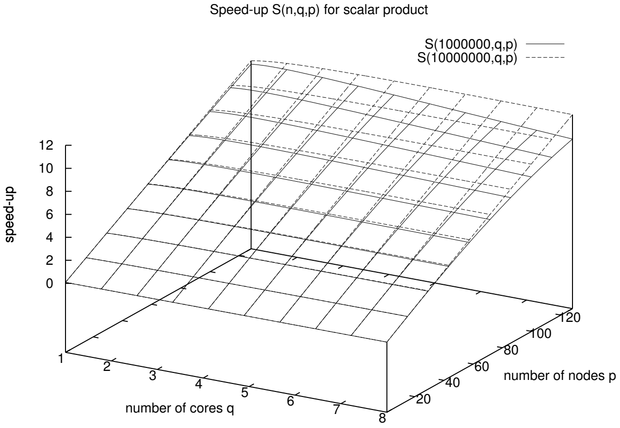
$$\eta_{sp}(q,p,n) \quad \leq \quad \frac{\frac{3}{34}n - \frac{3}{68}}{qn + \frac{3}{34}n + \frac{15}{7}q^2p^2 - \frac{3}{68}} \qquad (6)$$

Matrix-matrix multiplication

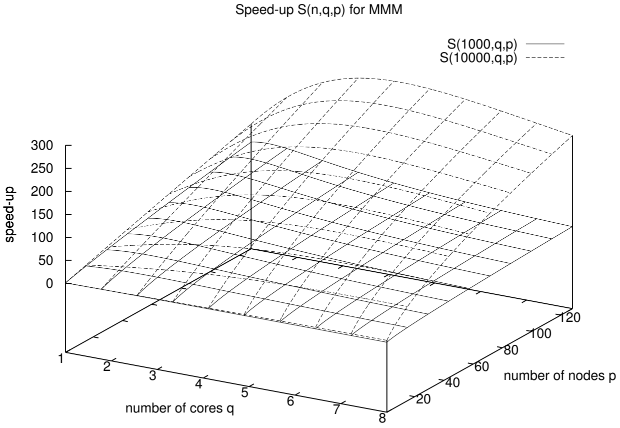
choosing $\beta(q,p) = \beta_1(p) = p$

$$\eta_{mm}(q,p,n) \quad \leq \quad \frac{n - \frac{1}{2}}{n + \frac{340}{7}q^2\sqrt{p} + \frac{34}{3}q - \frac{1}{2}} \qquad (7)$$

Model speed-up of scalar product depending on q und p for $n = 10^6, 10^7$, according to eq (6).



Model speed-up of MMM depending on *q* und *p*, according to eq (7).



Hardware parameters from row bwUniCluster.

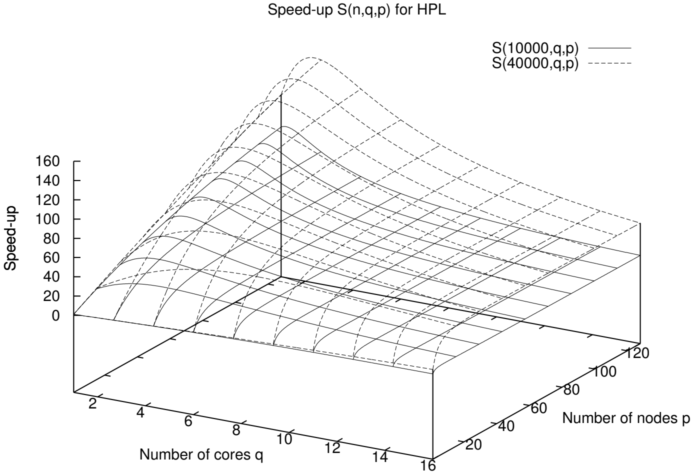
high performance Linpack

$$\eta_{hpl}(q,p,n) \quad \leq \quad \frac{n}{n + \frac{77}{17}q^2 p \log_2(p) + \frac{308}{9}q^2 p + \frac{24}{5}q} \qquad (8)$$

2-dim FFTW

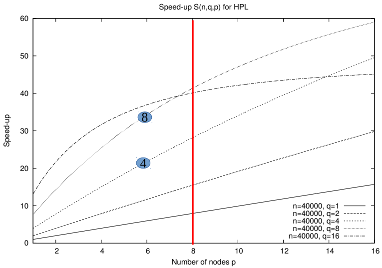
$$\eta_{2fftw}(q,p,n) \quad \leq \quad \frac{n\log_2(n)}{n\log_2(n) + \frac{8}{5}qn + \frac{616}{27}q^2 p \log_2(p)} \qquad (9)$$

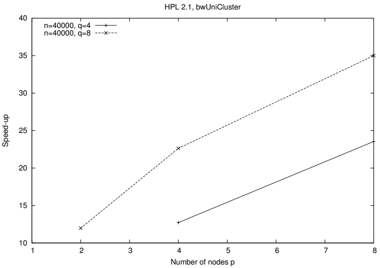
Model speed-up of HPL Linpack depending on *q* and *p*, according to eq (8).



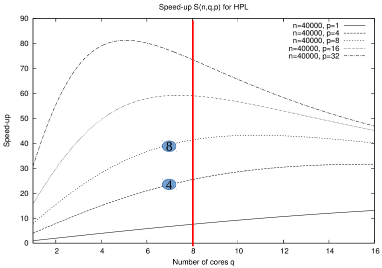
Model speed-up of HPL Linpack depending on *q* and *p*, according to eq (8).

Measurement of speed-up for HPL Linpack depending on *q* und *p* for bwUniCluster.

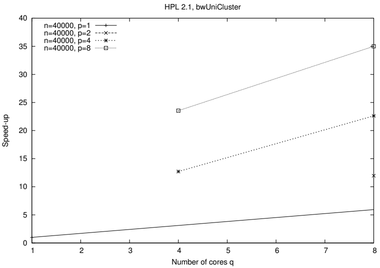




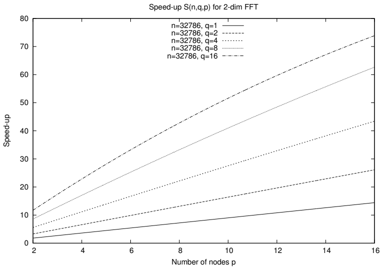
Model speed-up of HPL Linpack depending on *q* and *p*, according to eq (8).



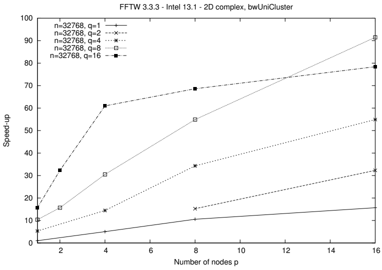
Measurement of speed-up for HPL Linpack depending on *q* und *p* for bwUniCluster.



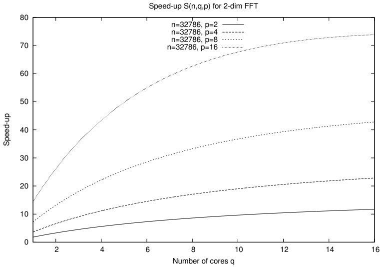
Model speed-up of 2-dim FFTW depending on *q* and *p*, according to eq (9).



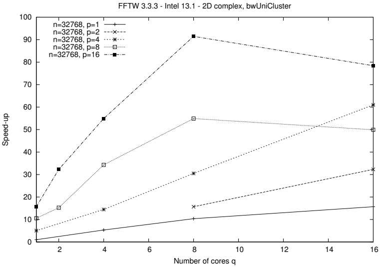
Measurement of speed-up of 2-dim FFTW depending on *q* und *p* for bwUniCluster.



Model speed-up of 2-dim FFTW depending on *q* and *p*, according to eq (9).



Measurement of speed-up of 2-dim FFTW depending on *q* und *p* for bwUniCluster.



Summary and Future Work

- Modeled a cluster of p multi-core-nodes with q cores to describe the performance by few dimensionless parameters, as metrics we choose the efficiency and the speed-up.
- Over all levels of the cluster we can find the important parameters, which integrate hard- and software-characteristics, like the operational intensity x, the ratio of data-bytes and exchange-bytes between nodes r, and the ratio of the nodes-interconnect-bandwidth and the internal bandwidth of the multi-core v.
- With the dimensionless product $v \cdot r(x, q, p) \cdot x$ and the scaling function $s(x/q)$ for a single multi-core we are able to understand the behaviour of a cluster by analyzing speed-up and efficiency.
- The transformation to a flat cluster with $(q = 1)$-cores is possible and reproduces the results of a earlier paper [Kredel, et. al. 2013] including the measures.

Weak points and future work

- Lack of a optimization procedure in order to find the best speed-up or efficiency by a given load or application. A possible model would be a flow of "operations on bytes", shaped like the current of a river, and executed by a number of processors, like a ship crossing the river in shortest time. This corresponds a non-linear optimization and will fix the needed number of processors at each time.
- The hierarchy structure of a cluster may open the way to renormalization group theory. Using the scaling function $s(z) = \frac{z}{1+z}$ one can try to analyze the cluster-system from single cores, multi-core-nodes, region of nodes and the cluster. Such an approach could give an optimal balancing of the application, distributed on nodes and multi-cores.
- Finding the best load-balancing may be the same objective as in finding the shortest time. Our concept of few dimensionless parameters is essential for both.

Questions?

Thank you!

Acknowledgments

- We thank Erich Strohmaier for discussion and contribution to the overall modeling approach.
- Part of this work was performed on the computational resource bwGRiD of the German D-Grid initiative funded by the Ministry for Education and Research, the Ministry for Science, Research and Arts, Baden-Württemberg.
- Part of this work was performed on the computational resource bwUniCluster funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC.
- Thanks also to the anonymous referees for the helpful suggestions to extend the scope of the paper.