WP 3 Project Deliverable D3.3

Report on available compression techniques

Project Number	IST-2000-29266
Project Title	Virtual Real Time Fire Emergency Simulator
Deliverable Type	Report
Deliverable Class	Public

Deliverable Number	D3.3
Title of Deliverable	Report on available compression / optimisation techniques [WP3.3]
Nature of the Deliverable	Report
Contributing WPs	WP 2, WP 3
Contractual Date of Delivery	30. March 2002
Actual Date of Delivery	31. Oct. 2002
URL	www.virtualfires.org
Authors	Gunther Lenz (SiTu), Thomas Reichl (SiTu)
Contact Details	Institute for Structural Analysis / SiTu Research Univ. Prof. Dipl.-Ing. Dr. techn. Gernot Beer Lessingstrasse 25/II 8010 Graz / Austria Tel.: +43 316 8736180 Fax: +43 316 8736185 Email: gernot.beer@ifb.tu-graz.ac.at

Abstract	WP3 Task 3 is responsible for the evaluation of existing compression tools for the CFD data to assure real time visualisation at a given IO-rate.
Keywords	Compression, CFD

Contents

1 Overview

Compression and optimisation of the CFD data within the VFS is considered important due to the following reasons:

Reduction of the storage size: As one timestep will deliver data for about 100000 voxels which results in a CGNS file of about 4 MB without the data for the gridpoint coordinates, this yields to a storage volume of about 7 GB for one single simulation run for a period of 30 min at 1 second intervals.

Reduction of transmission times: Transfer of these 4 MB of data over a standard 100Mbit connection takes around 0,5 sec. which is already half the sampling period. As this results in a network load of about 50% it will only be possible to scale up to around 200000 voxels before reaching the network limits.

Support for multiresolution / resampling: As the CFD delivers data at a spatial resolution required for a stable calculation this is usually not the resolution required by the visualisation system. Espacially the arrangement of the data values may differ significantly, e.g. if the CFD works with tetrahedral meshes and the visualisation with regular structured grids.

Support for spatial selection: The resulting CFD data grid may be greater than the extent of the actual viewing pyramid, so it is not necessary to transmit data for points which are actually out of sight.

The delay of delivering this report accrues, because report D2.4 [VF1] has to be taken into account, which was closed on August 19th 2002. Also the specification of the CFD database (D3.2) had to be taken into account. The data for the testset was made available on October 14th 2002.

2 Abbreviations

VFS	Virtual Fires Simulator
CFD	Computational Fluid Dynamics
DM	Data Management
CGNS	CFD General Notation System

3 Evaluation of existing compression tools

3.1 Available compression tools

At the time of the evaluation there was no compression tool existing, that was specifically designed to compress CGNS-files. So the test was performed by using COTS and freely available compression tools.

The following programs were evaluated:

- arj32 3.10a from ARJ Software, Inc. [1]
- compress from Microsoft Corp. [2]
- bzip2-1.0.2 by Julian Seward [3]
- gzip 1.2.4 by Jean-loup Gailly [4]
- ace 2.11 by ACE Compression Software & e-merge GmbH [5]
- RAR 3.00 beta 5 by Eugene Roshal [6]
- WinZip 8.0 (3105) by WinZip Computing, Inc. [7]

3.2 Testset for the evaluation

Each of the programs were tested against a set of 100 CGNS-files. These are the results of the timesteps from 100 - 199 seconds from the simulation of the old MontBlanc-tunnel. Each of the CGNS-files had a size of 12107776 Bytes and incorproated the following values for each of the 142597 gridpoints:

- CoordinateX
- CoordinateY
- CoordinateZ
- VelocityX
- VelocityY
- VelocityZ
- TemperatureStagnation

3.3 Testenvironment

3.3.1 Hardware

The tests were performed on a Dell Precision M-40 Mobile Workstation with a Intel Pentium III-Mobile CPU running at 1,2 GHz and 512 MB RAM.

3.3.2 Software

The tests were performed under Windows2000 SP3. To ensure the availability of the maximum CPU-time for the compression/decompression task all other tasks were stopped. All tools were started from the commandline.

3.4 Results

The averaged measurements for the testset are given in Table 1.

Archiver	File size [Bytes]	Compression Ratio [%]	Compression Time [s]	Decompression Time [s]
RAR fastest	4819789,10	60,19	8,13	1,72
RAR fast	4805614,70	60,31	10,05	1,39
RAR normal	3465562,30	71,38	9,02	1,16
RAR good	3452976,90	71,48	11,67	1,11
RAR best	3447431,10	71,53	15,82	1,03
zip superfast	6147532,30	49,23	3,48	1,45
zip fast	6021874,20	50,26	4,62	1,82
zip normal	6015320,90	50,32	4,24	1,25
zip max	5976438,00	50,64	24,78	1,73
ace superfast	3333740,30	72,47	7,13	2,05
ace fast	3328812,20	72,51	7,82	0,79
ace normal	3322704,70	72,56	8,36	0,61
ace good	3320496,50	72,58	8,72	0,87
ace max	3319392,60	72,58	9,97	0,77
lha fast	6134783,20	49,33	8,59	2,25
lha normal	6134783,90	49,33	8,56	1,22
lha max	6134783,00	49,33	9,72	0,74
gzip fast	6147096,10	49,23	2,53	0,68
gzip normal	5970357,40	50,69	7,88	1,01
gzip max	5963379,50	50,75	28,74	0,75
ms-cab fast	5950798,40	50,85	20,33	2,05
ms-cab normal	5950798,20	50,85	21,45	0,94
ms-cab max	4721151,70	61,01	55,12	2,13
bzip2 fast	5703781,00	52,89	1,73	3,05
bzip2 best	5684684,00	53,05	1,61	2,45
arj -m4	7330853,10	39,45	1,52	1,13
arj -m3	6163413,70	49,10	2,31	0,92
arj -m2	6087465,50	49,72	3,55	0,83
arj -m1	6066063,30	49,90	5,37	1,05

Table 1: Averaged results for one CGNS file

3.4.1 Filesize

As it can be seen from Figure 1 the resulting compression ratios are almost independent from the level of compression for each compression tool. Only RAR, MS-Cab and arj show a gain of about 10% by increasing their compression level.

Generally ace performs best, always reaching a level of compression beyond 70%. It is directly followed by RAR if it's used with it's options set better than "fast". RAR with its poorer options and MS-Cab with the "maximum" option set are around 60%. All other compression tools reach a level of about 50%. The only tool falling below 40% is ARJ with option "-m4".

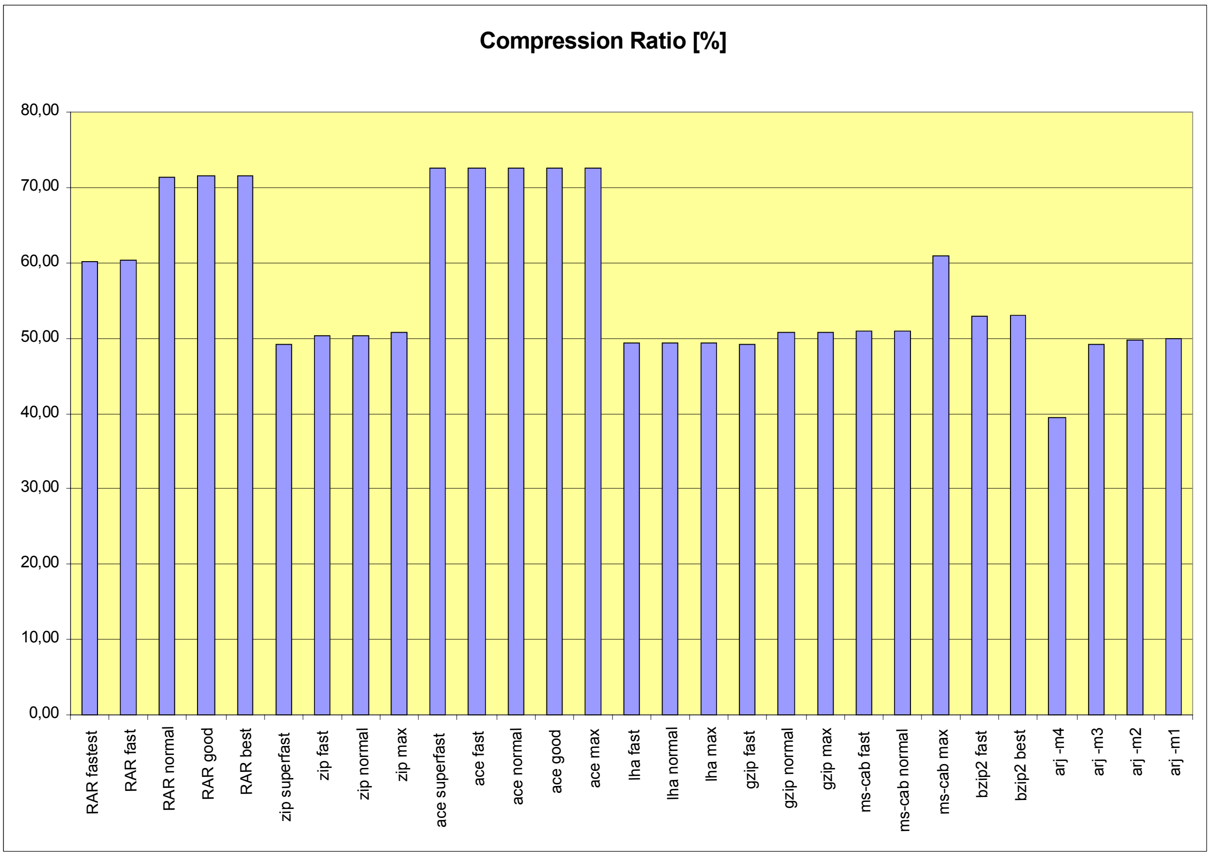

Figure 1: Compression ratio for each tool

3.4.2 Timeconsumption

Comparing the times required for compression and decompression, as shown in Figure 2, compression is the more time consuming task. Only bzip requires more time for decompression than for compression.

Although the resulting compression ratios are fairly equal, there's a big variation in the time consumption among the different tools and their options. MS-Cab is by far the slowest, requiring almost twice the time of the next better tool.

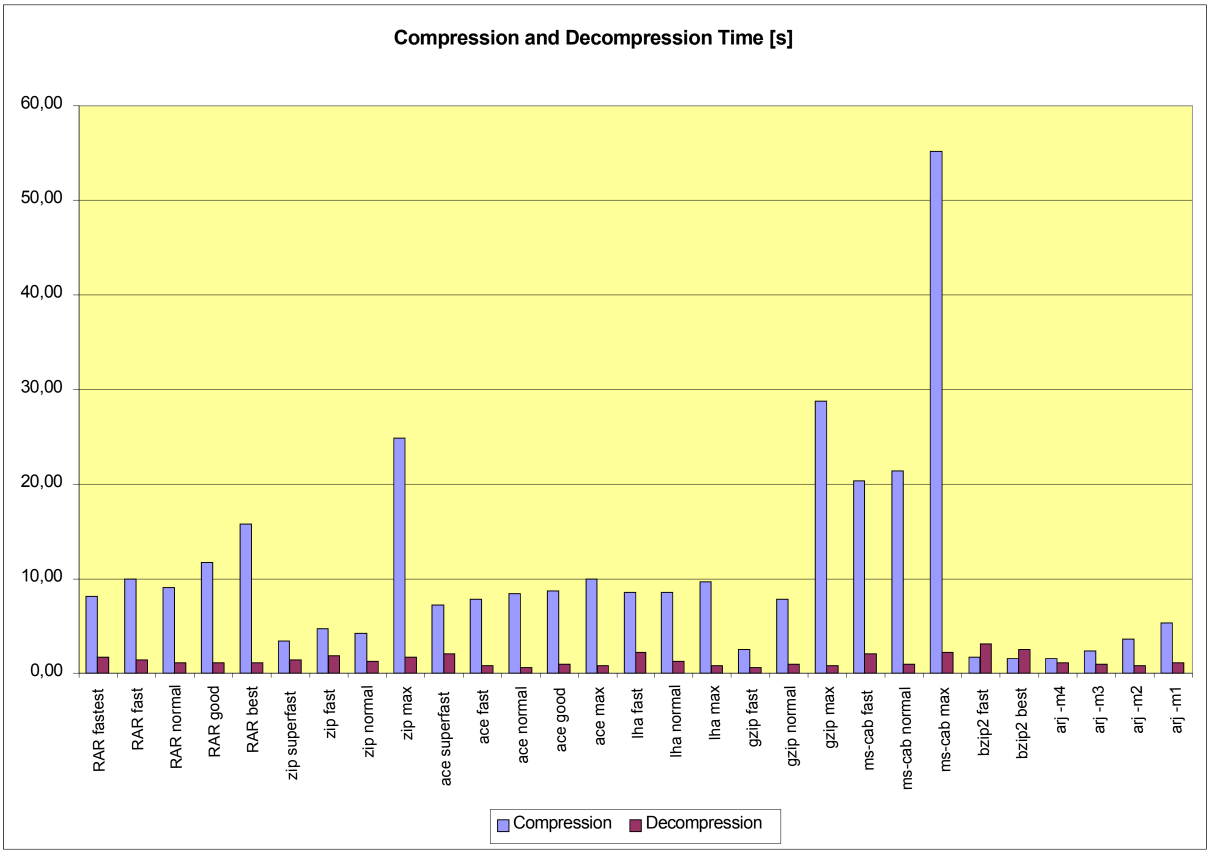

Figure 2: Time requirements for compression and decompression

3.4.3 Pipeline Bandwith

Figure 3 gives an overview of the possible dataflows from the server to the client:

- direct transmission: the data is transferred uncompressed over the network. Here the only limiting factor is the bandwith of the network connection.
- transmitting a precompressed file: compression has been performed in advance and the stored compressed file is sent to the client. This is only possible for a "playback"-scenario and not for the realtime version as there is no way of separating the compression task. The total bandwith depends on the throughput of the decompressor and the network connection bandwith.
- full compression - decompression pipeline: the datafile is compressed at the server side, transmitted to the client and decompressed there. Here the total bandwith depends on the throughput of the compression and decompression stages and the network connection bandwith.

The used symbols have the following meaning:

S_0	uncompressed size of the file [Bytes]	T_{tot}	total time for a scenario [s]
S_1	compressed size of the file [Bytes]	BW_c	bandwidth for compression [MB/s]
T_c	time required for compression [s]	BW_t	bandwith for transmission [MB/s]
T_t	time required for transmission [s]	BW_d	bandwidth for decompression [MB/s]
T_d	time required for decompression [s]	BW_{ctd}	bandwidth over all 3 stages [MB/s]
T_{ctd}	time required for all 3 stages [s]	BW_{tot}	total bandwith for a scenario [MB/s]

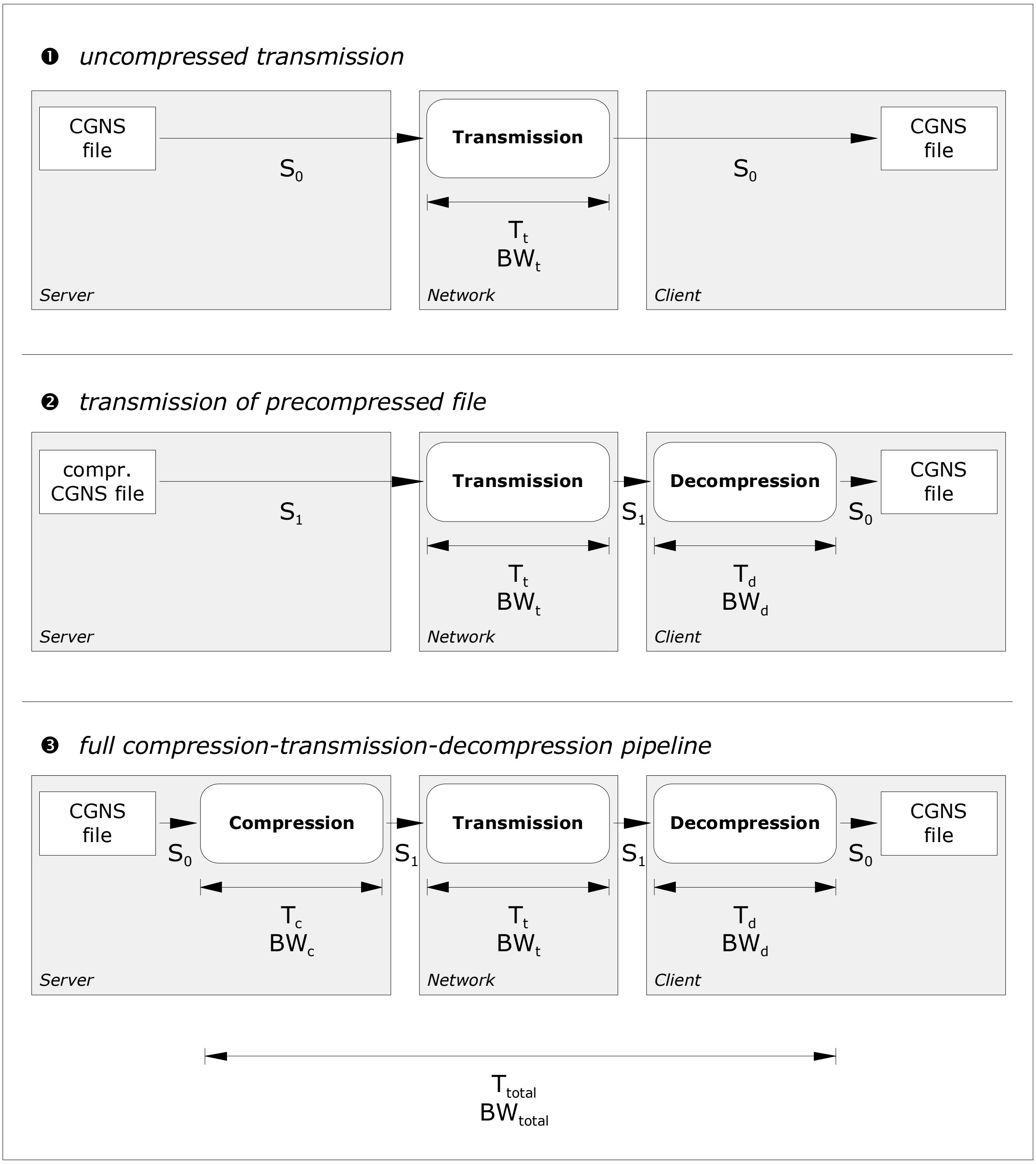

Figure 3: Dataflow for different transmission scenarios

The decision to select a useful compression scheme must be based on the achievable overall bandwidth BW_{tot} between the server and the client. This yields to the follwing bandwidths for the three different scenarios:

uncompressed transmission:

$$T_{tot} = T_t \qquad T_t = S_0 / BW_t \quad => \quad BW_{tot} = S_0 / T_{tot} = BW_t$$

transmission of precompressed file:

$$T_{tot} = T_t + T_d \qquad T_t = S_1 / BW_t \quad => \quad BW_{tot} = S_0 / T_{tot} = S_0 / (S_1/BW_t + T_d) = BW_{td}$$

full pipeline:

$$T_{tot} = T_c + T_t + T_d \qquad T_t = S_1 / BW_t \quad => \quad BW_{tot} = S_0 / T_{tot} = S_0 / (T_c + S_1/BW_t + T_d) = BW_{ctd}$$

From the times listed in Table 1 and the theoretical bandwith of a 100MBit network connection of 11,88 MB/s (excluding 5% protocol overhead) the possible throughput for each stage and the overall bandwith for the whole pipeline can be calculated. The results are given in Table 2 and Figure 4.

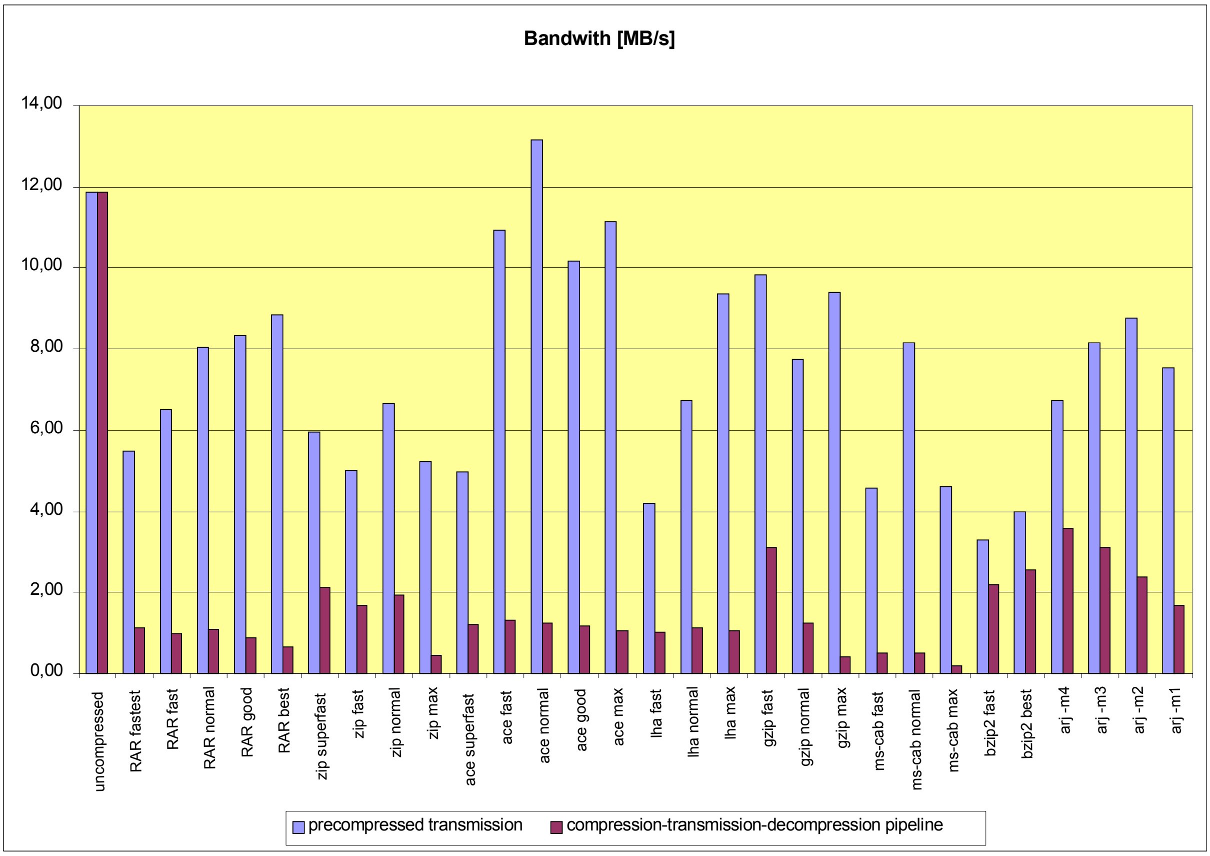

Figure 4: Achievable bandwith for the different scenarios

	compr. size S_1 [MB]	transport time T_t [s]	trans-deco time T_d+T_t [s]	trans-deco BW BW_{td} [MB/s]	3 stage time T_{ctd} [s]	3 stage BW BW_{ctd} [MB/s]
MontBlanc		0,97	0,97	11,88	0,97	11,88
CGNS original	11,55					
RAR fastest	4,60	0,39	2,11	5,48	10,24	1,13
RAR fast	4,58	0,39	1,78	6,50	11,83	0,98
RAR normal	3,31	0,28	1,44	8,03	10,46	1,10
RAR good	3,29	0,28	1,39	8,32	13,06	0,88
RAR best	3,29	0,28	1,31	8,84	17,13	0,67
zip superfast	5,86	0,49	1,94	5,94	5,42	2,13
zip fast	5,74	0,48	2,30	5,01	6,92	1,67
zip normal	5,74	0,48	1,73	6,66	5,97	1,93
zip max	5,70	0,48	2,21	5,22	26,99	0,43
ace superfast	3,18	0,27	2,32	4,98	9,45	1,22
ace fast	3,17	0,27	1,06	10,92	8,88	1,30
ace normal	3,17	0,27	0,88	13,17	9,24	1,25
ace good	3,17	0,27	1,14	10,16	9,86	1,17
ace max	3,17	0,27	1,04	11,14	11,01	1,05
lha fast	5,85	0,49	2,74	4,21	11,33	1,02
lha normal	5,85	0,49	1,71	6,74	10,27	1,12
lha max	5,85	0,49	1,23	9,37	10,95	1,05
gzip fast	5,86	0,49	1,17	9,84	3,70	3,12
gzip normal	5,69	0,48	1,49	7,75	9,37	1,23
gzip max	5,69	0,48	1,23	9,40	29,97	0,39
ms-cab fast	5,68	0,48	2,53	4,57	22,86	0,51
ms-cab normal	5,68	0,48	1,42	8,14	22,87	0,50
ms-cab max	4,50	0,38	2,51	4,60	57,63	0,20
bzip2 fast	5,44	0,46	3,51	3,29	5,24	2,20
bzip2 best	5,42	0,46	2,91	3,97	4,52	2,56
arj -m4	6,99	0,59	1,72	6,72	3,24	3,57
arj -m3	5,88	0,49	1,41	8,16	3,72	3,10
arj -m2	5,81	0,49	1,32	8,76	4,87	2,37
arj -m1	5,79	0,49	1,54	7,51	6,91	1,67

Table 2: Achievable bandwidths for the different scenarios
($BW_t = 11{,}88$ MB/s, $S_0 = 11{,}55$ MB)

To be more effective than simple uncompressed data transmission the overall bandwidth BW_{tot} has to be greater than the network bandwidth BW_t. This yields to the following relations:

transmission of precompressed file:

$$c \, . \, BW_d > BW_t$$

full pipeline:

$$c \, . \, BW_{cd} > BW_t$$

where: c compression ratio: [0..1), where 0 means uncompressed transmission. $c = 1 - S_1/S_0$

BW_d decompression bandwidth

BW_{cd} compression-decompression bandwidth

The compression ratio - bandwidth product for each compression tool is given in Table 3. In Figure 4 the achieved bandwidth over the compression ratio is drawn. For comparison the lines for the minimum required bandwidth for a given network connection are also shown in this figure.

The results for a given network connection:

10 MBit/s:

All compression tools lead to a significant increase of overall bandwidth for the "playback"-scenario, i.e. the transmission of precompressed data files. The gain is in the range of 2 to 8.

For the full pipeline only zip-superfast, gzip-fast, bzip with all options and arj-m2 to -m4 lead to an increase in bandwidth.

100 MBit/s:

There is only 1 case where the employment of compression yields to an increase of bandwidth for the "playback"-scenario. This is the case for the transmission of a precompressed CGNS-file by using ACE with option "normal".

For the full pipeline all compression tools lead to a significant loss of overall bandwith.

1 GBit/s:

Even for the "playback"-scenario the loss of bandwidth is around 90%.

	compression ratio [%]	BW_d [MB/s]	BW_{cd} [MB/s]	$c . BW_d$ [MB/s]	$c . BW_{cd}$ [MB/s]	BW_t [MB/s]
RAR fastest	60,19	6,71	1,17	4,04	0,71	11,88
RAR fast	60,31	8,31	1,01	5,01	0,61	11,88
RAR normal	71,38	9,95	1,13	7,11	0,81	11,88
RAR good	71,48	10,40	0,90	7,44	0,65	11,88
RAR best	71,53	11,21	0,69	8,02	0,49	11,88
zip superfast	49,23	7,96	2,34	3,92	1,15	11,88
zip fast	50,26	6,34	1,79	3,19	0,90	11,88
zip normal	50,32	9,24	2,10	4,65	1,06	11,88
zip max	50,64	6,67	0,44	3,38	0,22	11,88
ace superfast	72,47	5,63	1,26	4,08	0,91	11,88
ace fast	72,51	14,62	1,34	10,60	0,97	11,88
ace normal	72,56	18,93	1,29	13,73	0,93	11,88
ace good	72,58	13,27	1,20	9,63	0,87	11,88
ace max	72,58	15,00	1,08	10,88	0,78	11,88
lha fast	49,33	5,13	1,07	2,53	0,53	11,88
lha normal	49,33	9,46	1,18	4,67	0,58	11,88
lha max	49,33	15,60	1,10	7,70	0,54	11,88
gzip fast	49,23	16,98	3,60	8,36	1,77	11,88
gzip normal	50,69	11,43	1,30	5,80	0,66	11,88
gzip max	50,75	15,40	0,39	7,81	0,20	11,88
ms-cab fast	50,85	5,63	0,52	2,86	0,26	11,88
ms-cab normal	50,85	12,28	0,52	6,25	0,26	11,88
ms-cab max	61,01	5,42	0,20	3,31	0,12	11,88
bzip2 fast	52,89	3,79	2,42	2,00	1,28	11,88
bzip2 best	53,05	4,71	2,84	2,50	1,51	11,88
arj -m4	39,45	10,22	4,36	4,03	1,72	11,88
arj -m3	49,10	12,55	3,57	6,16	1,76	11,88
arj -m2	49,72	13,91	2,64	6,92	1,31	11,88
arj -m1	49,90	11,00	1,80	5,49	0,90	11,88

Table 3: Compression ratio - bandwidth products

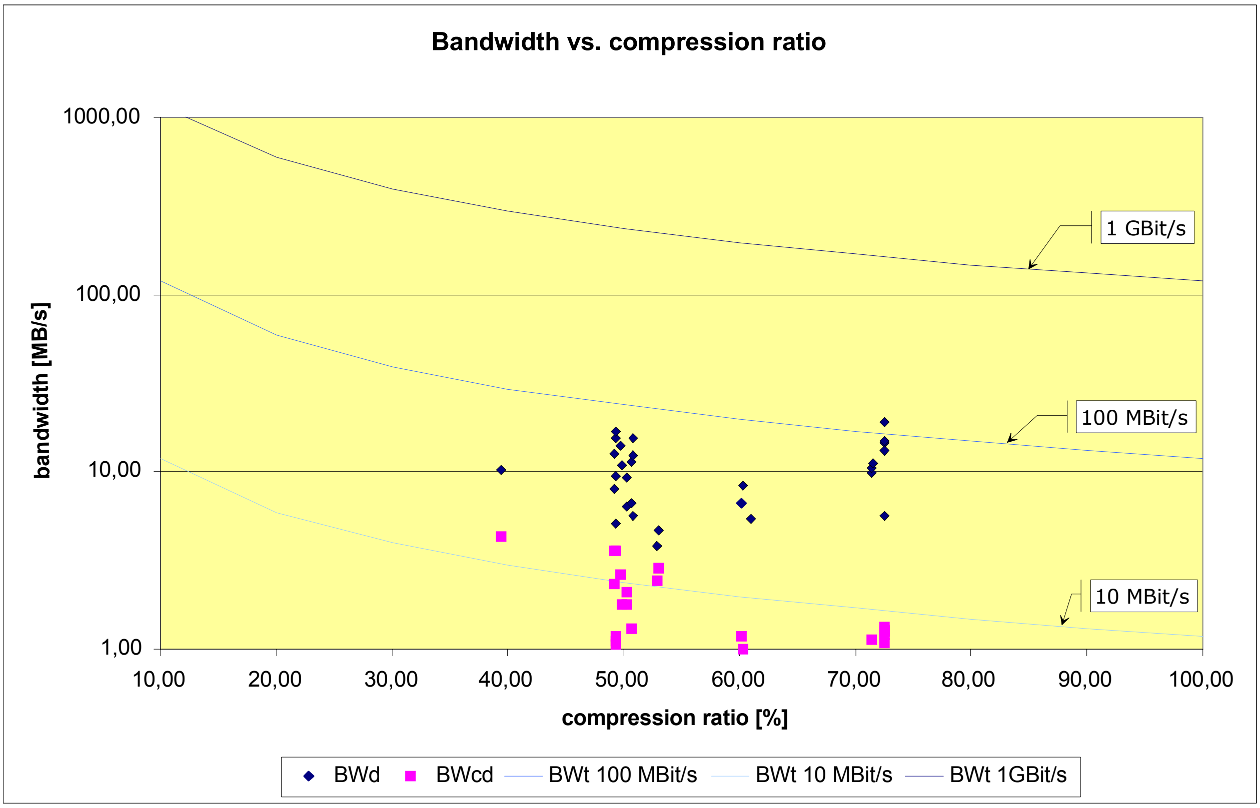

Figure 4: Bandwidth vs. compression ratio

4 Conclusion

Given a standard 100 MBit/s network connection, the tests have shown that there's currently no compression tool available, that leads to a gain in overall transmission bandwidth for the realtime simulation.

Only one tool leads to an improvement of bandwidth for the playback-scenario.

The measured compression and decompression bandwidths were made with full CPU usage and so they indicate peak values. On a single processor machine these values can only be reached, if there are no other tasks performed.

Concerning the overall transmission bandwidth the use of compression is useful only when the available network bandwidth drops significantly below 10 MBit/s. But also in this case the additional amount of required computing power must be taken into account. The tradeoff between increased transmission bandwidth and reduced CPU resources for the main applications tasks has to be verified seperately.

Looking at the compression ratio for storage reasons ace seems to be the tool of choice. It has also the advantage, that it's the only tool whose decompression bandwidth exceeds a 100MBit/s connection.

Therefore a two-way strategy is suggested:

- direct transmission of uncompressed CFD results for the realtime scenario
- offline compression for the storage within the DM to reduce the required storeage volume

The available CPU resources for decompression on the client side have to be evaluated when the first prototype of the visualisation system has been implemented.

5 Literature/Links

[1] http://www.arjsoft.com/

[2] http://www.microsoft.com/

[3] http://sources.redhat.com/bzip2/

[4] http://www.gzip.org/

[5] http://www.winace.com/

[6] http://www.rarlab.com/

[7] http://www.winzip.com/