ÓBUDAI EGYETEM

HP-SEE USER FORUM, Belgrade, 2012. October 17-19

Performance and scalability evaluation of short fragment sequence alignment applications

Windisch, G - John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary
Kozlovszky, M. - John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary
Balasko, A. - Lab. of Parallel & Distrib. Comput., MTA SZTAKI, Budapest, Hungary

Scientific motivation

- The recently used deep sequencing techniques present a new data processing challenge: mapping short fragment reads to open-access eukaryotic (animal: focusing on mouse and rat) genomes at the scale of several hundred thousands.
- This task is solvable by algorithms like BLAST, BWA. - which is one of the most frequently used tool in bioinformatics
- Local installations of these algorithms are typically not able to handle such problem size therefore the procedure runs slowly, while web based implementations cannot accept high number of queries.
- SEE-HPC infrastructure allows accessing massively parallel architectures and the sequence alignment code is distributed free for academia.
- The aim of the task is threefold,
  - the first task was to port the BLAST algorithm to the massively parallel HP-SEE infrastructure
  - create a BLAST service, which is capable to serve the short fragment sequence alignment demand of the regional bioinformatics communities,
  - to do sequence analysis with high throughput short fragment sequence alignments against the eukaryotic genomes to search for regulatory mechanisms controlled by short fragments
- For more details, please see [1, 2]

Ó
B
U
D
A
I

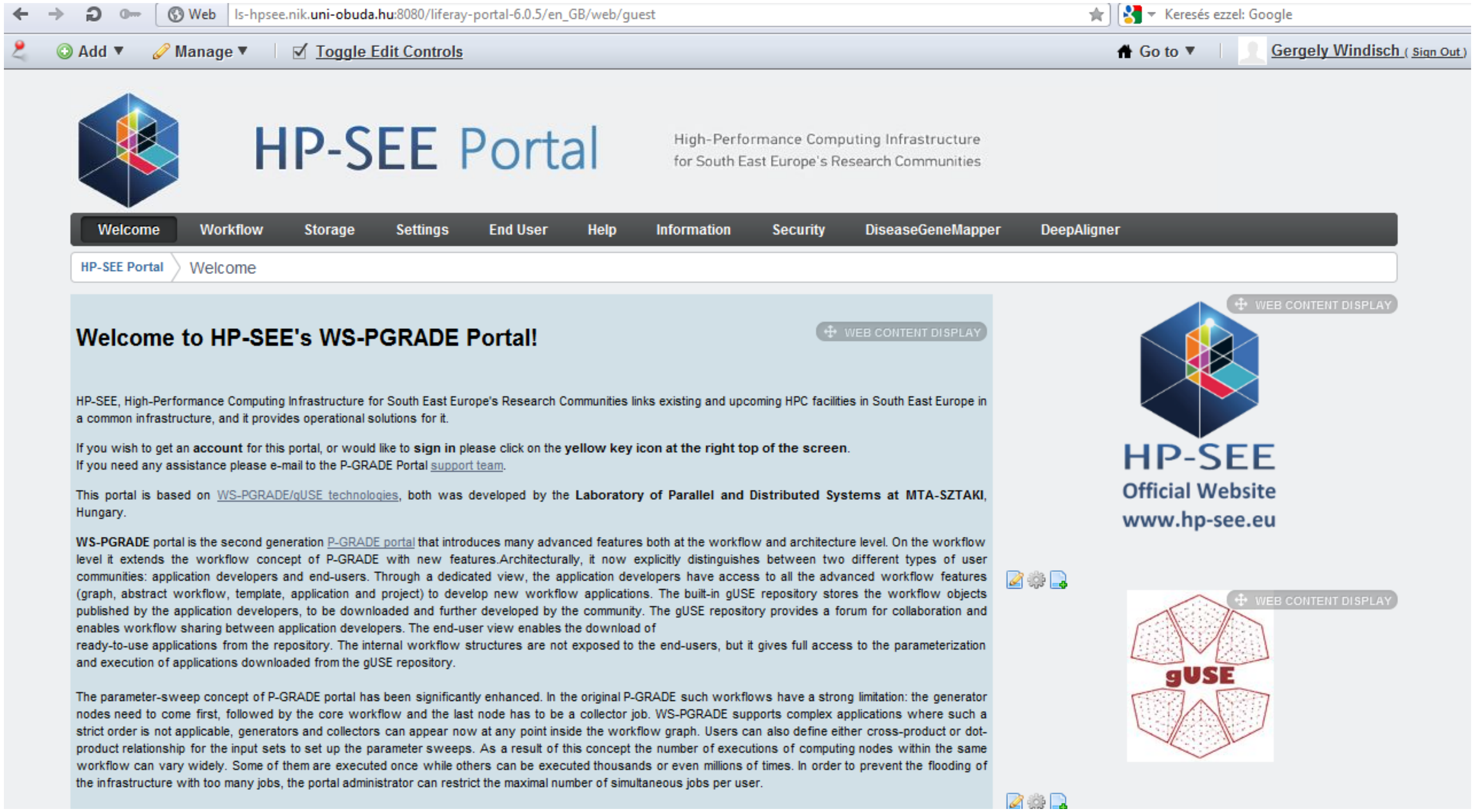
E
G
Y
E
T
E
M

Role of Obuda University in the project

- Create and operate a Life Science portal
- Port the applications to supercomputing infrastructure
  - Enhancing wall clock performance by optimization
- Create services that use the ported applications and make them available on the portal

ÓBUDAI EGYETEM

LS-HPSEE portal @ Obuda University


Web
ls-hpsee.nik.uni-obuda.hu:8080/liferay-portal-6.0.5/en_GB/web/guest
Keresés ezzel: Google
Add ▼
Manage ▼
Toggle Edit Controls
Go to ▼
Gergely Windisch (Sign Out)
HP-SEE Portal
High-Performance Computing Infrastructure for South East Europe's Research Communities
Welcome
Workflow
Storage
Settings
End User
Help
Information
Security
DiseaseGeneMapper
DeepAligner
HP-SEE Portal
Welcome
WEB CONTENT DISPLAY
HP-SEE Official Website www.hp-see.eu
gUSE

Welcome to HP-SEE's WS-PGRADE Portal!

HP-SEE, High-Performance Computing Infrastructure for South East Europe's Research Communities links existing and upcoming HPC facilities in South East Europe in a common infrastructure, and it provides operational solutions for it.

If you wish to get an **account** for this portal, or would like to **sign in** please click on the **yellow key icon at the right top of the screen**.
If you need any assistance please e-mail to the P-GRADE Portal support team.

This portal is based on WS-PGRADE/gUSE technologies, both was developed by the **Laboratory of Parallel and Distributed Systems at MTA-SZTAKI**, Hungary.

WS-PGRADE portal is the second generation P-GRADE portal that introduces many advanced features both at the workflow and architecture level. On the workflow level it extends the workflow concept of P-GRADE with new features.Architecturally, it now explicitly distinguishes between two different types of user communities: application developers and end-users. Through a dedicated view, the application developers have access to all the advanced workflow features (graph, abstract workflow, template, application and project) to develop new workflow applications. The built-in gUSE repository stores the workflow objects published by the application developers, to be downloaded and further developed by the community. The gUSE repository provides a forum for collaboration and enables workflow sharing between application developers. The end-user view enables the download of ready-to-use applications from the repository. The internal workflow structures are not exposed to the end-users, but it gives full access to the parameterization and execution of applications downloaded from the gUSE repository.

The parameter-sweep concept of P-GRADE portal has been significantly enhanced. In the original P-GRADE such workflows have a strong limitation: the generator nodes need to come first, followed by the core workflow and the last node has to be a collector job. WS-PGRADE supports complex applications where such a strict order is not applicable, generators and collectors can appear now at any point inside the workflow graph. Users can also define either cross-product or dot-product relationship for the input sets to set up the parameter sweeps. As a result of this concept the number of executions of computing nodes within the same workflow can vary widely. Some of them are executed once while others can be executed thousands or even millions of times. In order to prevent the flooding of the infrastructure with too many jobs, the portal administrator can restrict the maximal number of simultaneous jobs per user.

Figure 1: HP-SEE Bioinformatics eScience Gateway Portal running at Obuda University.
Server is mainaned by SZTAKI, backend infrastructure provided by NIIF
The portal has just been opened for the public

ÓBUDAI EGYETEM

Developed Services

- Short sequence analysis
  - Deep Aligner
    - Runs BLAST on a huge number of short fragments against a large database
- Disease Mapping
  - Disease Gene Mapper
    - Maps known genes associated to a disease to other organizations
- Design goals
  - Easy to use for the scientists
  - High performance
  - Highly scalable
- Requirements
  - Web browser
  - User account on the HP-SEE server (available at Obuda University)
  - User account and certificate for the NIIF supercomputing centers (available at NIIF)

Developed Services

- Both services are gUSE portlets
- gUSE is a WS-PGrade portal developed at MTA SZTAKI, Hungary
  - Workflow based operation

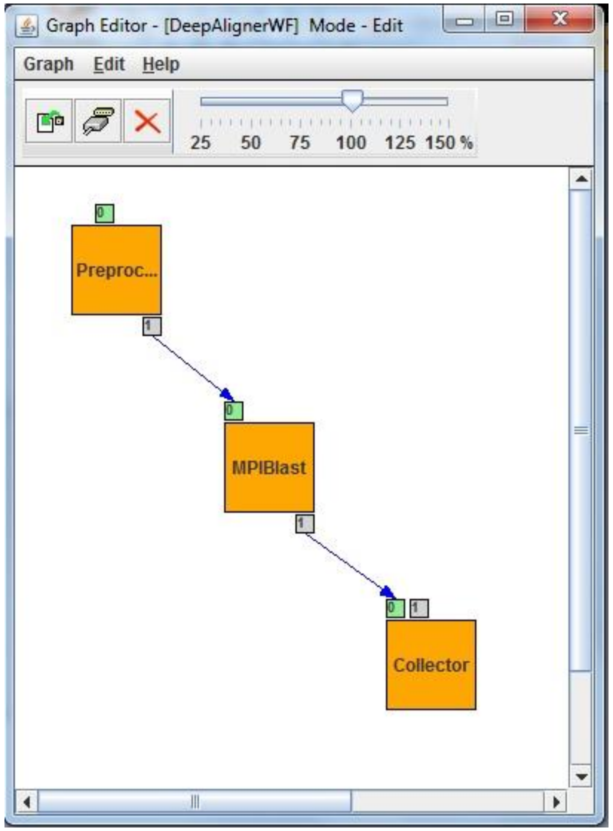
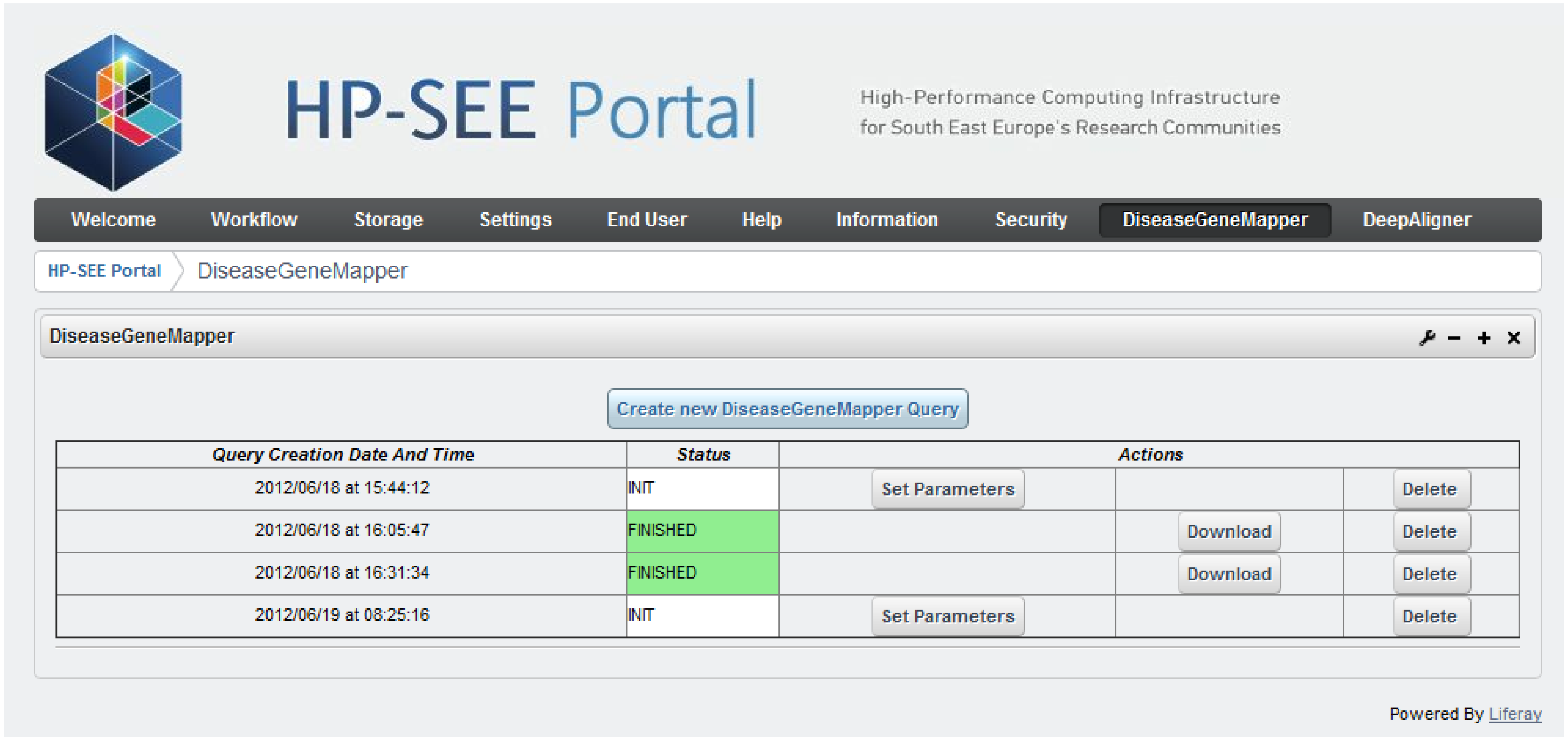


Figure 2: Simplified workflow graph of Deep Aligner

ÓBUDAI EGYETEM

Developed Services - DGM

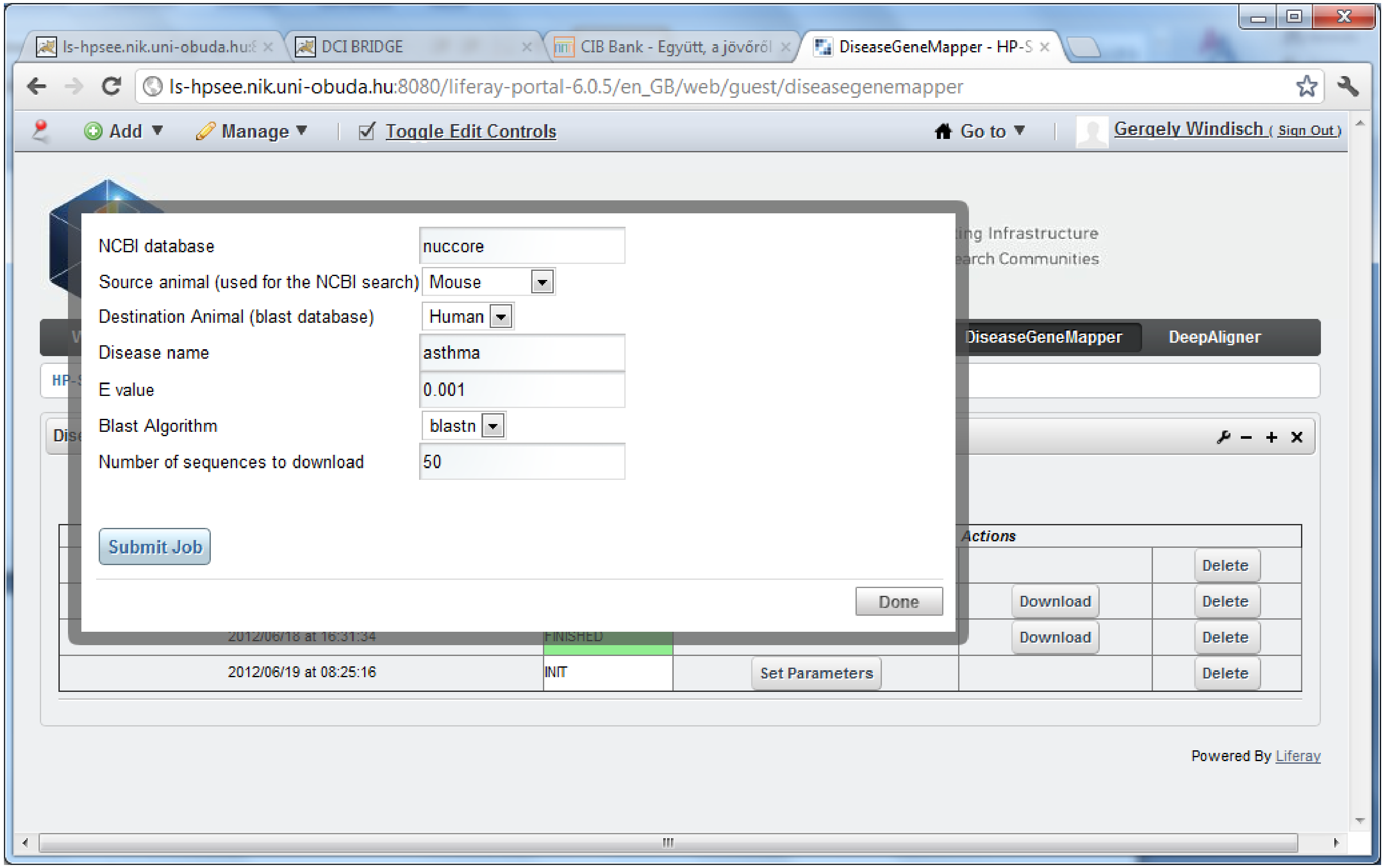


Query Creation Date And Time	Status	Actions		
2012/06/18 at 15:44:12	INIT	Set Parameters		Delete
2012/06/18 at 16:05:47	FINISHED		Download	Delete
2012/06/18 at 16:31:34	FINISHED		Download	Delete
2012/06/19 at 08:25:16	INIT	Set Parameters		Delete

Figure 3: Disease Gene Mapper portlet main window

ÓBUDAI EGYETEM

Developed Services - DGM

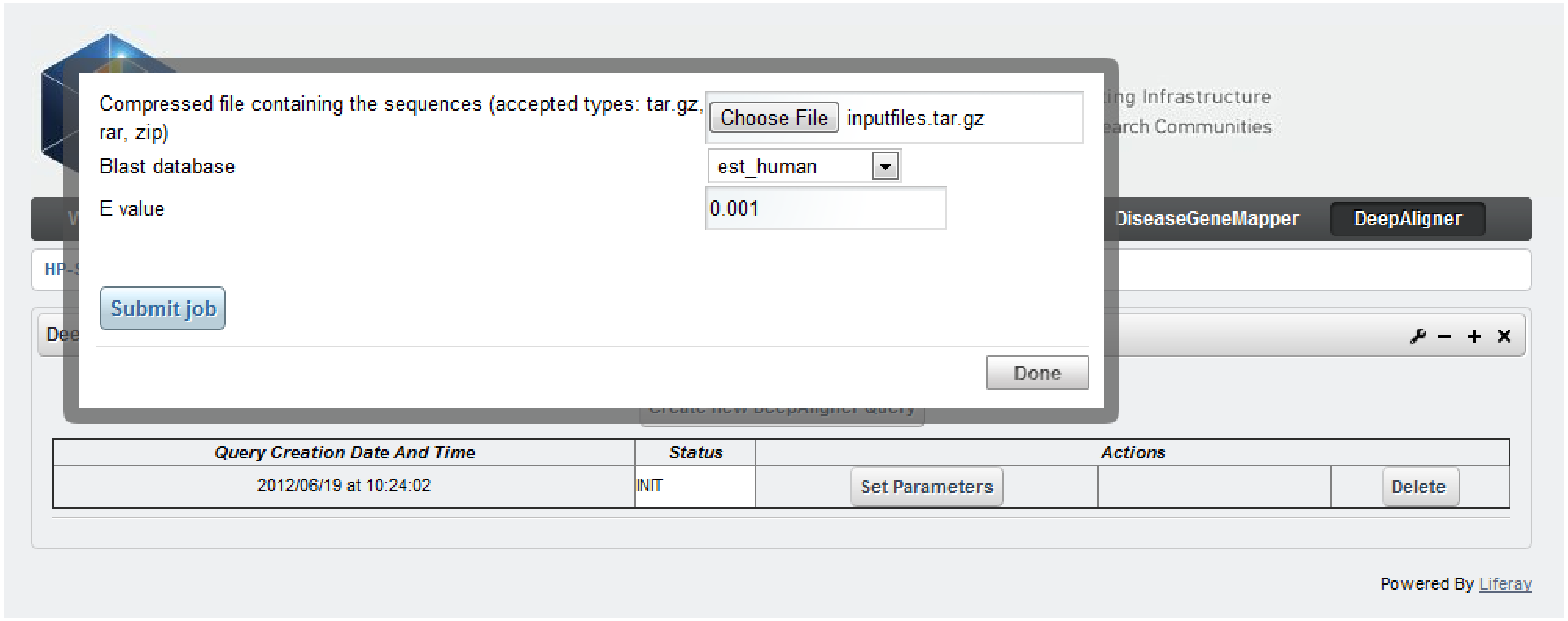

ls-hpsee.nik.uni-obuda.hu:8080/liferay-portal-6.0.5/en_GB/web/guest/diseasegenemapper

NCBI database	nuccore
Source animal (used for the NCBI search)	Mouse
Destination Animal (blast database)	Human
Disease name	asthma
E value	0.001
Blast Algorithm	blastn
Number of sequences to download	50

2012/06/18 at 16:31:34	FINISHED		Download	Delete
2012/06/19 at 08:25:16	INIT	Set Parameters		Delete

Figure 4: Disease Gene Mapper set properties

Developed Services - DeepAligner


Compressed file containing the sequences (accepted types: tar.gz, rar, zip) inputfiles.tar.gz

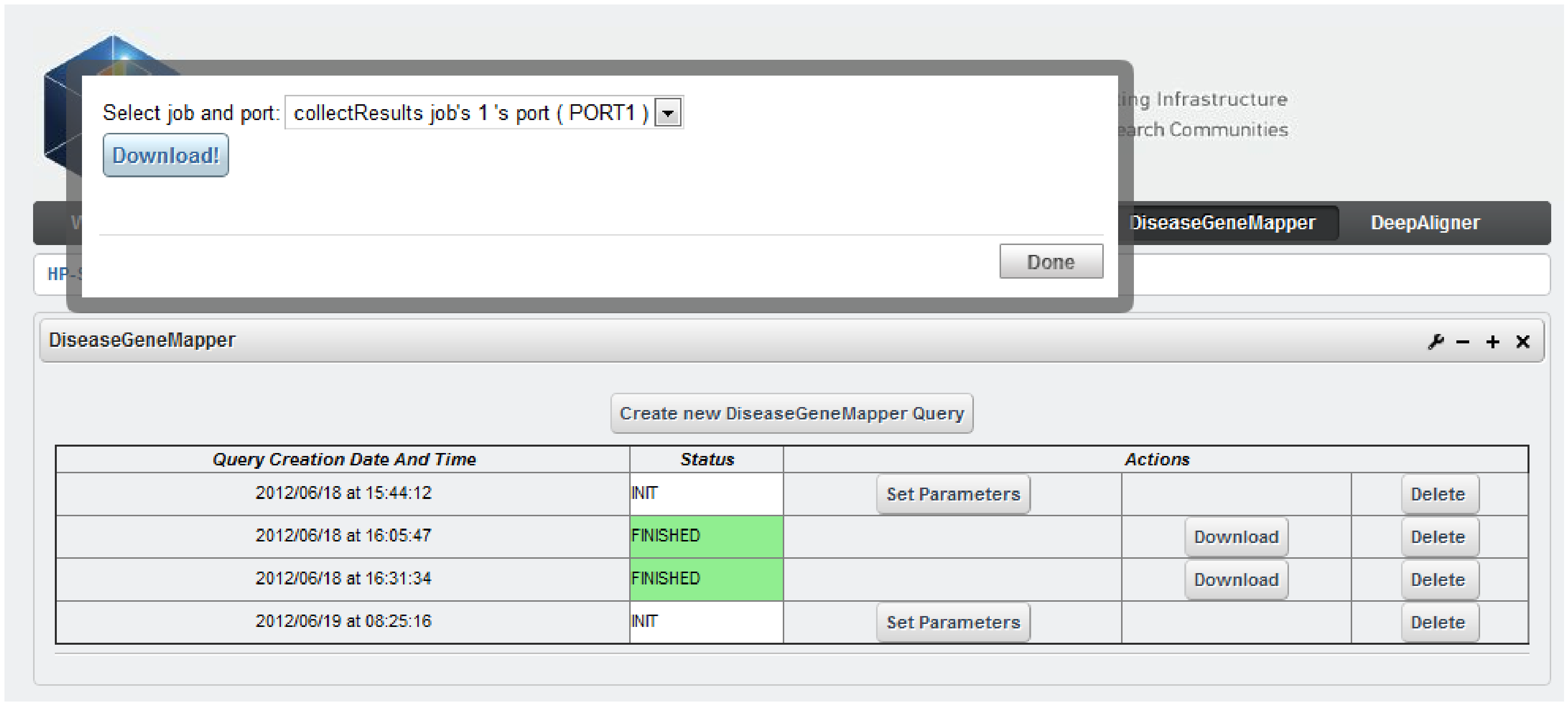
Blast database: est_human

E value: 0.001

Query Creation Date And Time	Status	Actions		
2012/06/19 at 10:24:02	INIT	Set Parameters		Delete

Figure 5: Deep Aligner – set properties

Developed Services - DeepAligner



Select job and port: collectResults job's 1 's port (PORT1)

Query Creation Date And Time	Status	Actions		
2012/06/18 at 15:44:12	INIT	Set Parameters		Delete
2012/06/18 at 16:05:47	FINISHED		Download	Delete
2012/06/18 at 16:31:34	FINISHED		Download	Delete
2012/06/19 at 08:25:16	INIT	Set Parameters		Delete

Figure 6: Downloading results

Execution time analysis

- Amdahl’s law
  - „The speedup of a program using multiple processors in parallel computing is limited by the time needed for the sequential fraction of the program”
- Both applications consist of three jobs
  - job1: preparation (sequential)
  - job2: execution (highly parallel OpenMPI)
  - job3: results collection (sequential)
- Applications differ mainly in Job1 and Job2
  - Job2 uses the same algorithms (they only differ in their parameters) so the performance evaluation holds for both applications

Execution time analysis – Job 1

- Job 1 in DeepAligner
  - Takes n input sequences from the user
  - Input files come in a tar.gz
    - faster to process than all sequences in one big file
    - pigz could be used for multithreaded decompress
      - not used in our app for compatibility reasons
  - Overall percentage of execution time is about 0.01% of the whole job – 32 node MPI
  - No real reason to parallelize

Execution time analysis – Job 1

- Job 1 in DiseaseGeneMapper
  - Gets the name of a disease from the user
  - Downloads gene sequences from the NCBI database associated with the given disease
- The speed of the internet connection is vital in this application
  - on average downloading one sequence takes about 0.913 s
  - total execution time is O(n)
  - can be parallelized – multithreaded downloader
    - problem: NCBI server detects abuse with too many threads and shuts down the connection
    - we use a single thread downloader to avoid accidents

ÓBUDAI EGYETEM

Execution time analysis – Job 2

- Job 2 in both applications
  - Uses MPIBlast to search for the gene sequence
  - Most time consuming job by far
    - ~99.3% of the total execution time is spent in this job
    - ~99.1% of Job2's execution time is spent on MPIBlast
    - profiling MPIBlast [5] shows that on average 85% of the time is spent on actual BLAST search, about 7% is fragment copy & communication 3% is printing the results. Other functions use up the rest of the time

ÓBUDAI EGYETEM

Execution time analysis – Job 3

- Job 3 in both applications
  - Receives the results from the MPIBlast jobs (one from each)
  - Compresses the results
  - Sends it back to gUSE
- Sequential execution
  - pigz could be used to speed it up
- Does not run long enough to worth optimizing

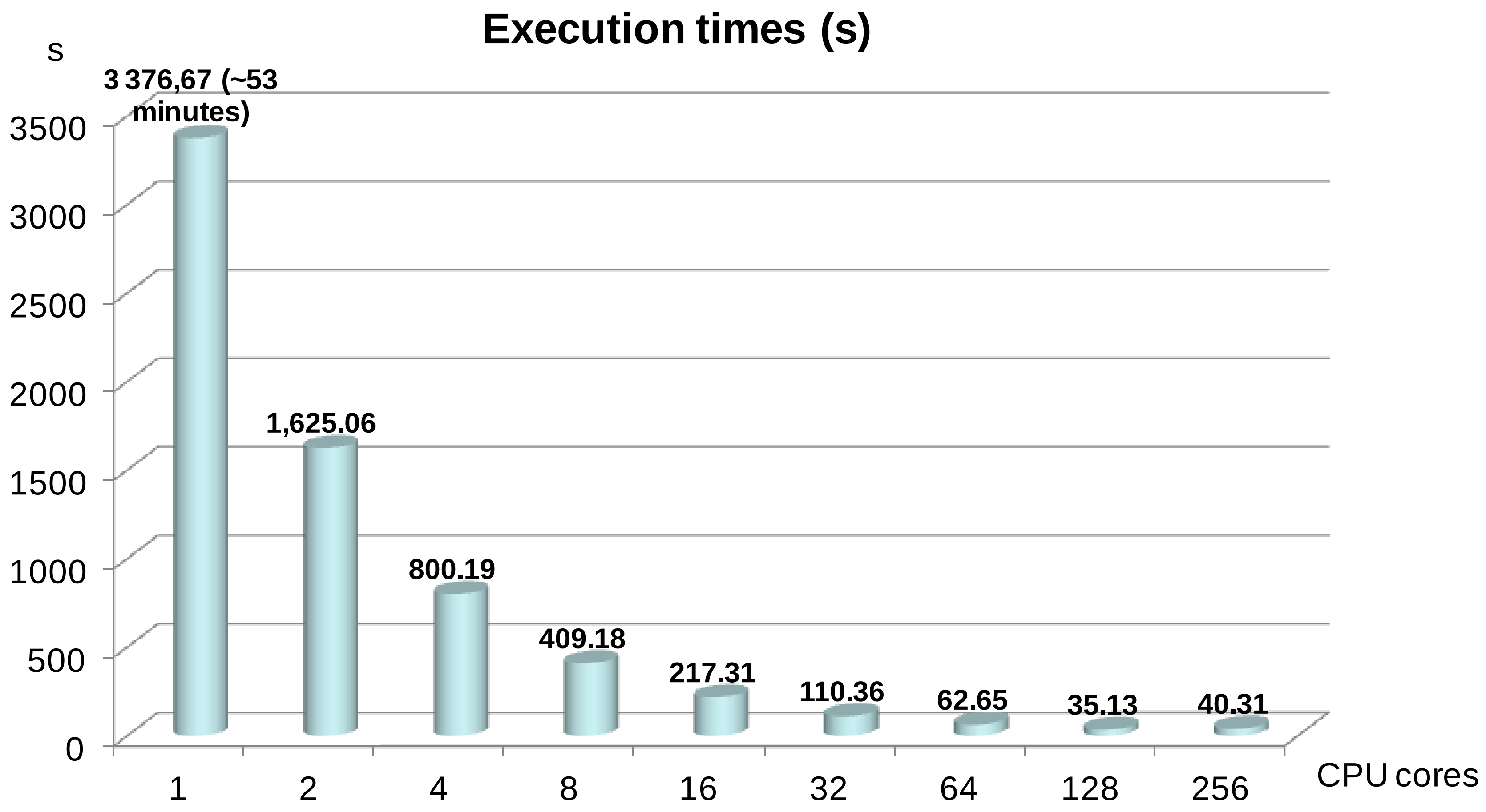
Performance and scalability measurements

- Job2 was the real candidate for performance optimization
- We chose MPIBlast for the main algorithm because of it's proven speed and reliability [3,4]
- Following performance measurements were executed in the NIIFI supercomputing center
  - Database size: 5.1 GB
  - Input sequence sizes:
    - 29.13 kB
    - 58.42 kB
    - 130.41 kB
      - note: the scalability figures were similar for all three, the execution times on the following slides are for the first input (29.13 kB)

Measurement methodology

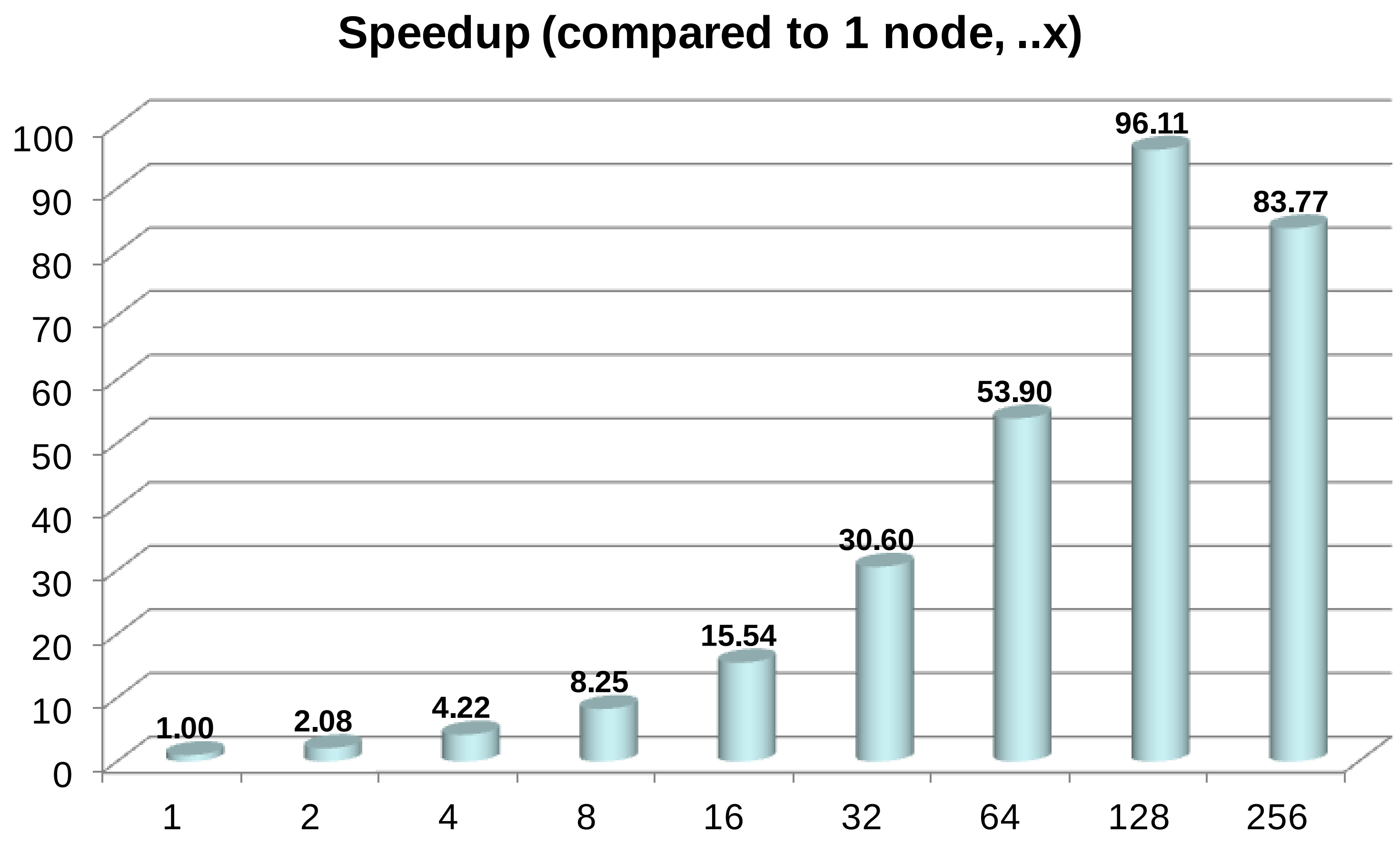
- Each measurement was executed 10 times
- The average of the executions was taken as the value
- note: the measurements have actually been executed on x+2 nodes, but 2 nodes are always used for administration purposes only

Scalability & Performance results



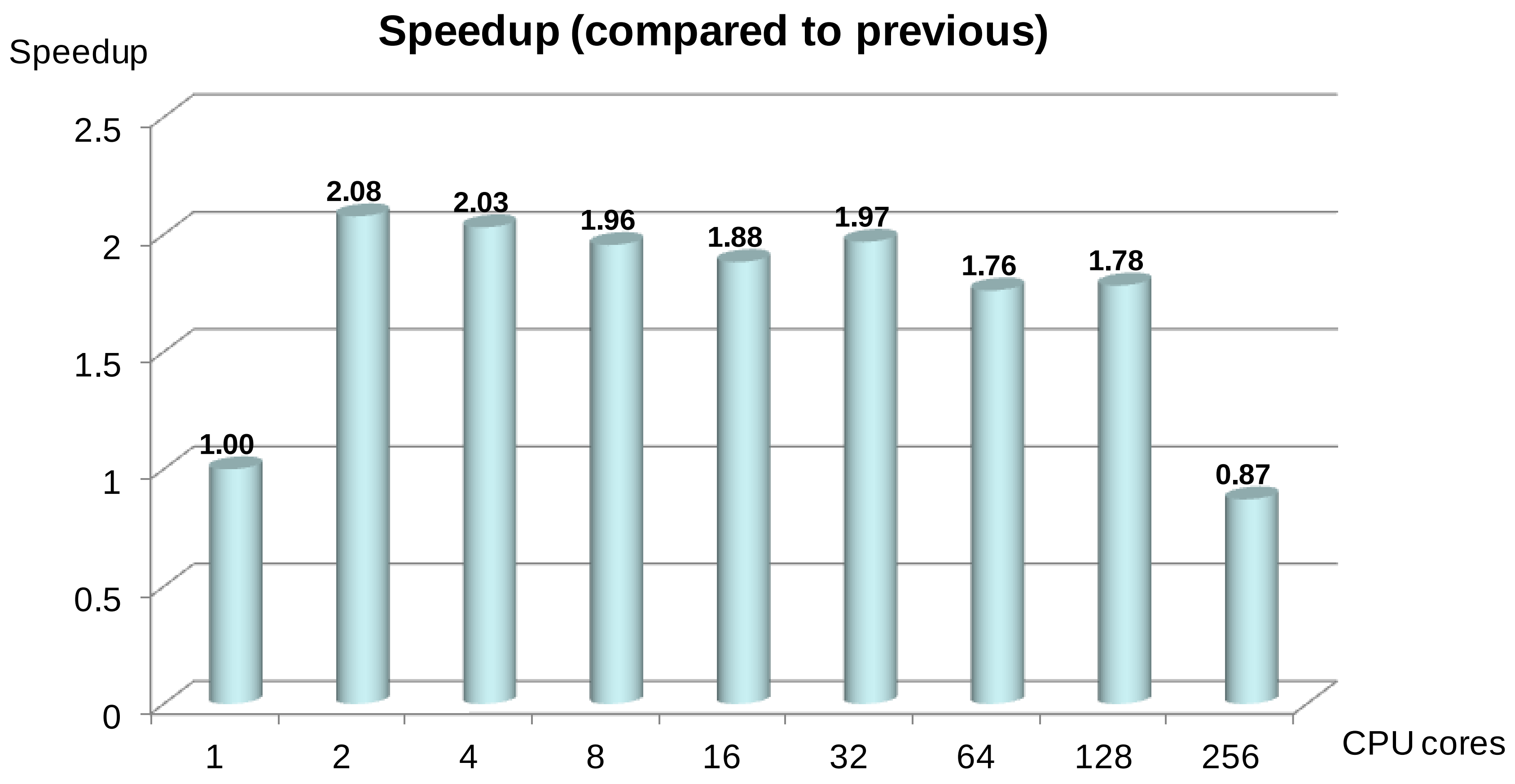
Graph 1: Execution times in seconds. The application scales well.

Scalability & Performance results



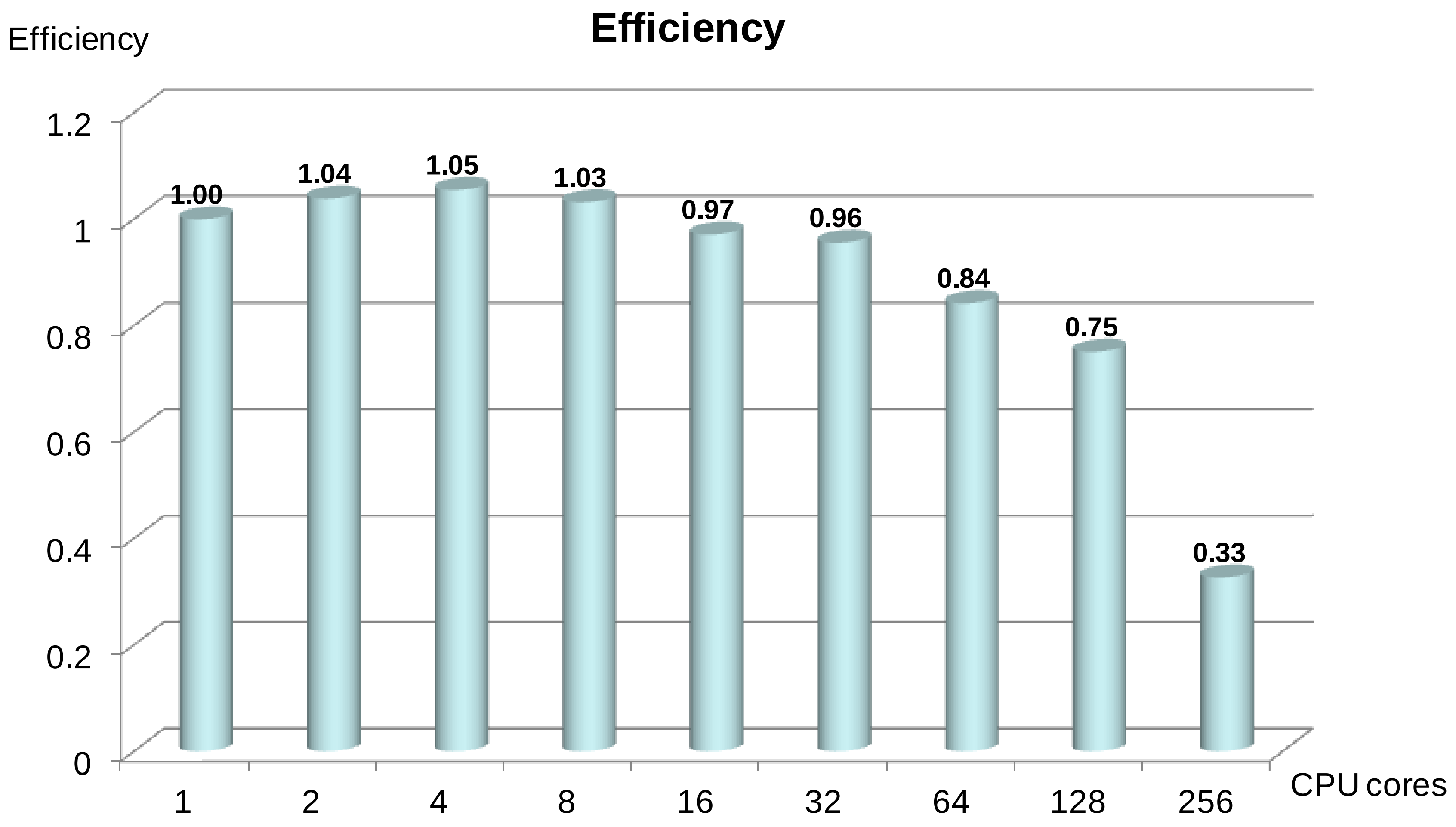
Graph 2: Speedup – performance factor compared to 1 node.

Scalability & Performance results



Graph 3: Speedup – performance factor compared to the previous node number.

Scalability & Performance results



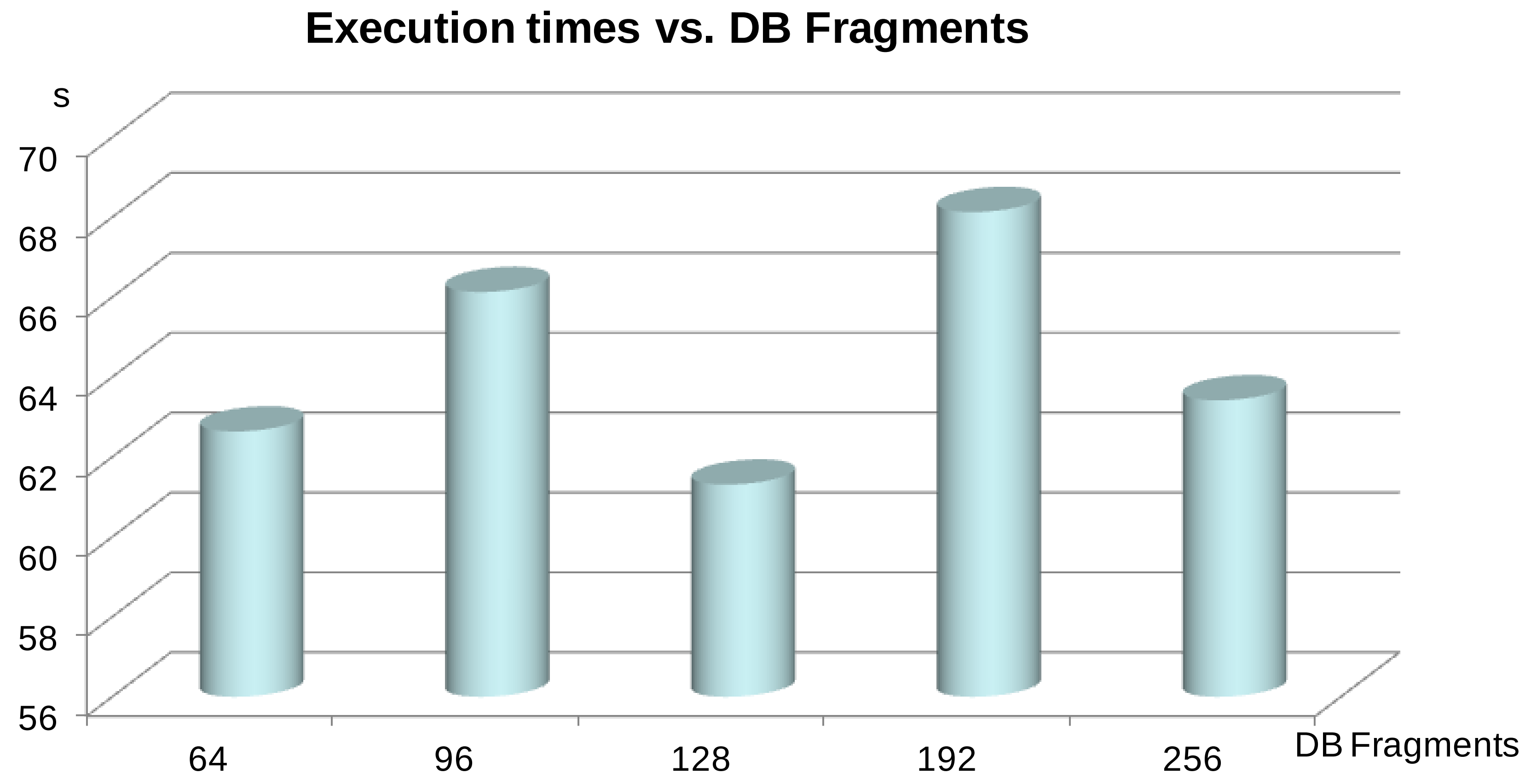
Graph 4: Speedup – Efficiency – Performance / no. of nodes.

ÓBUDAI EGYETEM

Optimizing DB Fragment number

- BLAST aligns the sequences in large gene databases
- MPIBlast uses the same databases, but the databases are split up into smaller pieces
- According to our measurements, DB frament number impacts performance
  - important to find the optimal number of fragments

Optimizing DB Fragment number



Graph 5: Execution time on 64 CPU cores. Fragment size should be an integer multiple of the CPU cores

Scalability conclusion

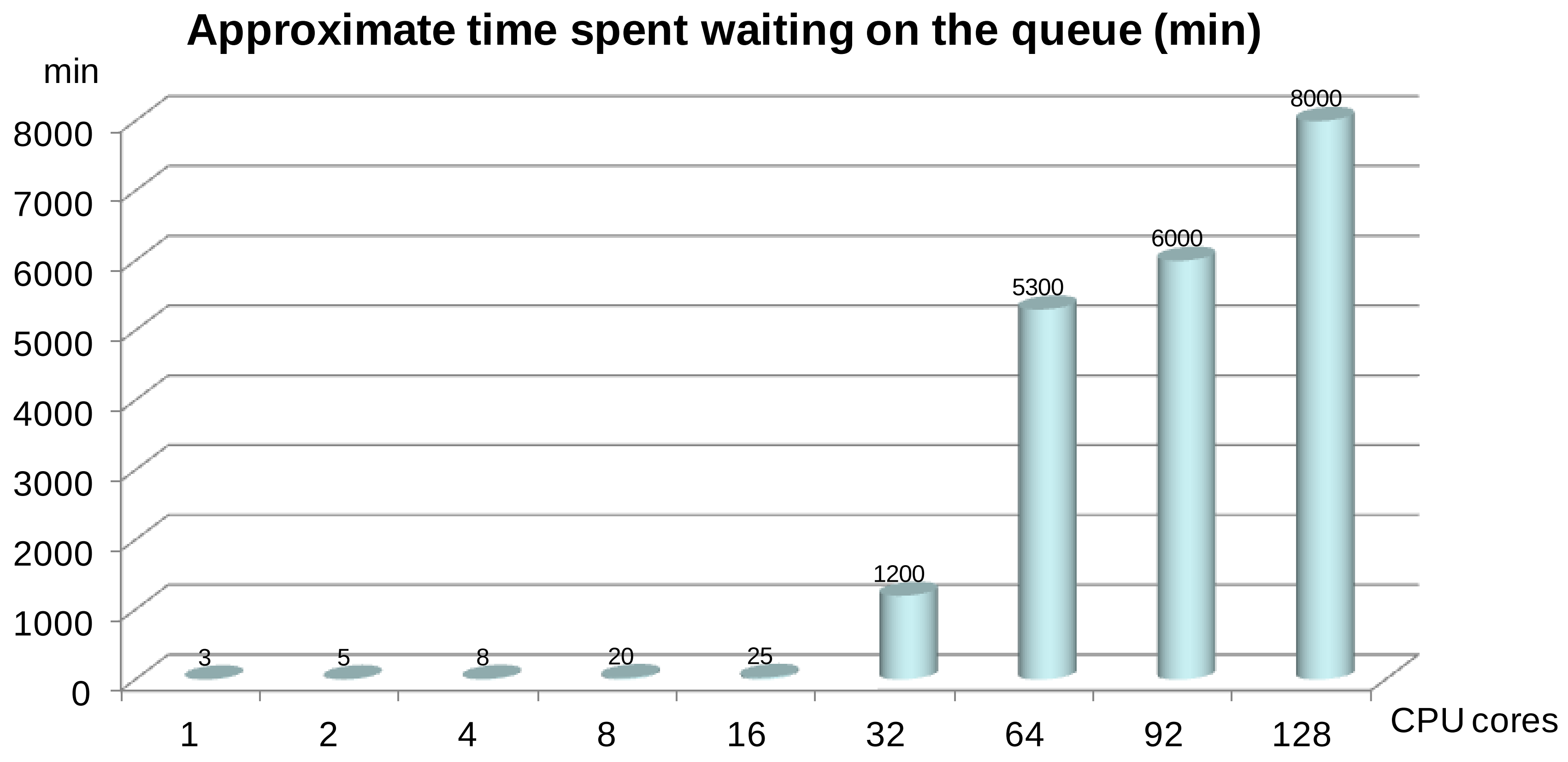
- Our current implementation peaks at around 128 cores
  - Speedup is almost linear – 96x at 128 nodes – even better at fewer.
- Increasing the number of MPI nodes any further yields only minor performance increase
  - reason is the communication overhead
- Further optimization did not help significally
  - --use-parallel-writes

ÓBUDAI EGYETEM

Optimizing for real world performance

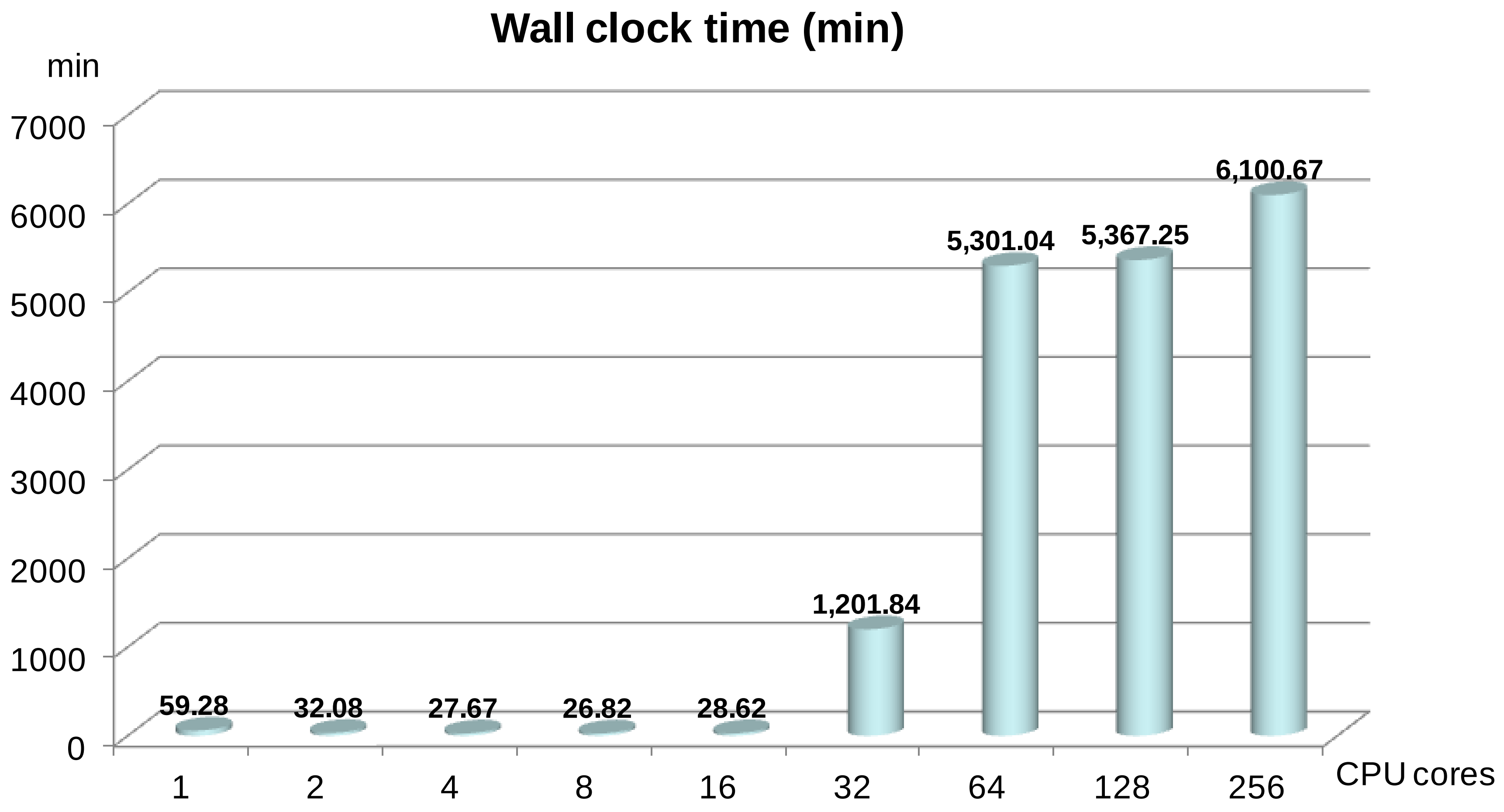
- Synthetic testing shows that the higher the number of nodes, the better performance it yields – up to 128 nodes
- However: life is not just fun and games
  - Depending on the Supercomputer's utilization smaller jobs actually finish faster in real life according to our experience
    - the scheduler policy decides when to execute applications based on required / available resources
- Measurements were executed on NIIF's Budapest server (Sun Grid Engine Open Grid Scheduler (OGS/GE 2011.11p1))
  - 768 CPU cores
  - The server is highly utilized at all times
  - Jobs were executed with normal user rights

Optimizing for real world performance



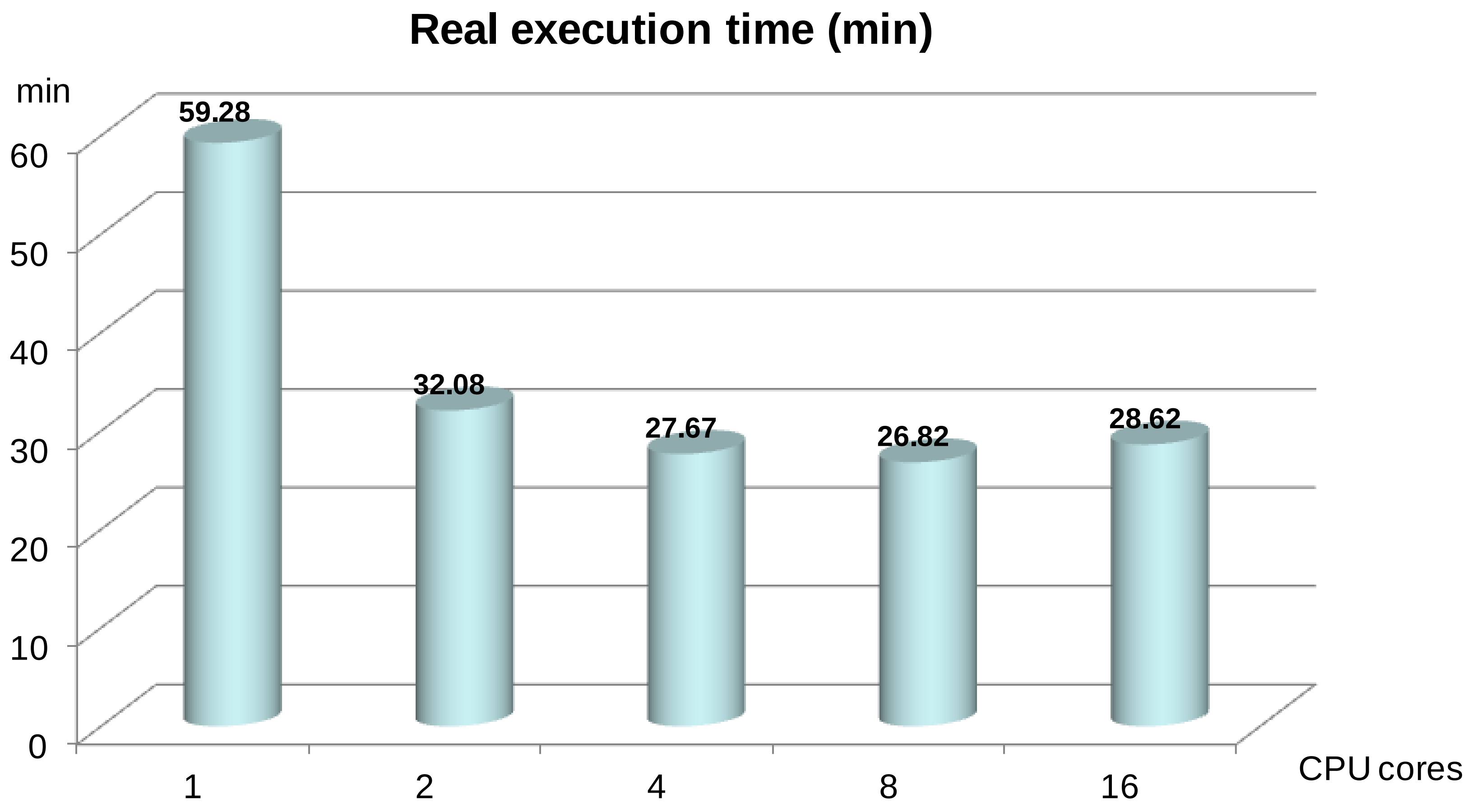
Graph 7: Minutes spent on the queue. On a busy HPC system jobs scheduled for a high number of nodes do not get scheduled for a long time

Optimizing for real world performance



Graph 8: Wall clock time – execution time + waiting
Huge gap at 32 nodes

Optimizing for real world performance



Graph 8: Wall clock time – execution time + waiting
Excluding nodes above 16

Optimizing for real world performance

- Conclusion
  - The best overall throughput was achieved by setting the MPI node count to 16
  - Optimal value for the currently used supercomputing infrastructure
  - The evaluations should be re-executed periodically or when there is an update on the servers (new scheduler version, HW upgrade etc.)
  - Our results apply for our servers – different configuration will behave differently
    - Optimization should be done for every supercomputer

Summary

- Life Science portal was developed and is available for the scientists
  - http://ls-hpsee.nik.uni-obuda.hu:8080/liferay-portal-6.0.5/en_GB/web/guest
  - researchers can add their services to the portal
- DiseaseGeneMapper and DeepAligner was ported to the supercomputing infrastructure
  - optimization lead to high performance
- Services were created which runs on the portal
  - http://ls-hpsee.nik.uni-obuda.hu:8080/liferay-portal-6.0.5/en_GB/web/diseasegene
  - http://ls-hpsee.nik.uni-obuda.hu:8080/liferay-portal-6.0.5/en_GB/web/deepaligner

ÓBUDAI EGYETEM

ÓBUDAI EGYETEM

Future work

- Enhancing the performance of the applications even further
  - Using different compilers
- Adding further applications to gUSE / HPC
- Making these applications available on the HP-SEE Bioinformatics eScience Gateway
- Connecting the HP-SEE Bioinformatics eScience Gateway Portal to Supercomputing infrastructures of other countries
  - portal is capable of communicating with different kinds of middlewares

ÓBUDAI EGYETEM

References

[1] M. Kozlovszky, G. Windisch, Á. Balaskó;Short fragment sequence alignment on the HP-SEE infrastructure;MIPRO 2012

[2] M. Kozlovszky, G. Windisch; Supported bioinformatics applications of the HP-SEE project's infrastructure; Networkshop 2012

[3] A. Darling, L. Carey, and W. Feng; The Design, Implementation, and Evaluation of mpiBLAST; 4th International Conference on Linux Clusters; June 2003.

[4] H. Lin, X. Ma, P.Chandramohan, A. Geist, and N. Samatova; Efficient Data Access for Parallel BLAST; IEEE International Parallel & Distributed Processing Symposium; April 2005.

[5] H. Lin, P. Balaji, R. Poole, C. Sosa, X. Ma, W. Feng ; Massively Parallel Genomic Sequence Search on the Blue Gene/P Architecture; IEEE/ACM SC2008; November 2008.