

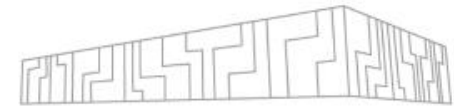


# TRENDS IN HPC NAISS 2023

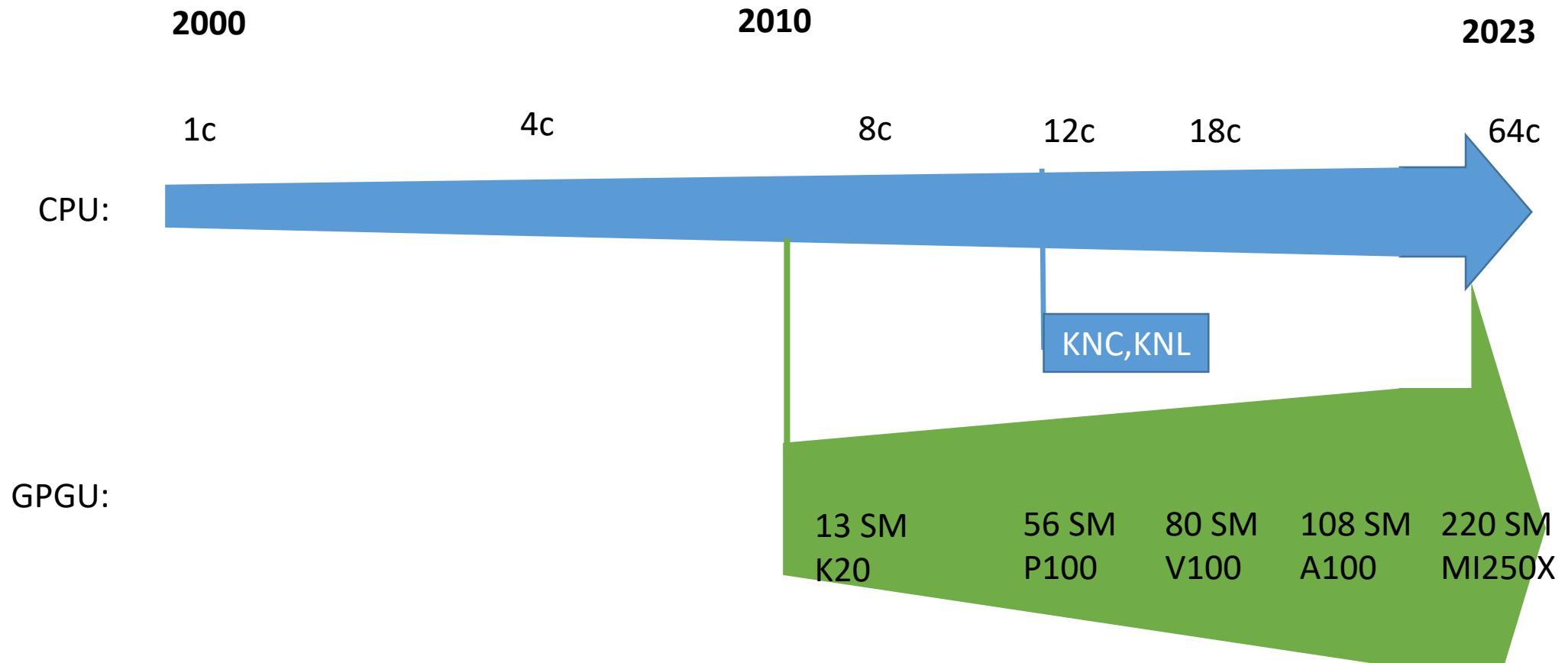
**BRANISLAV JANSÍK**

**6 DECEMBER, 2023  
LIU, LINKÖPING**

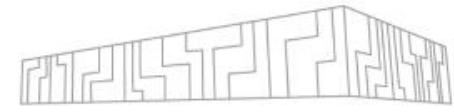
# COMPUTE TRENDS



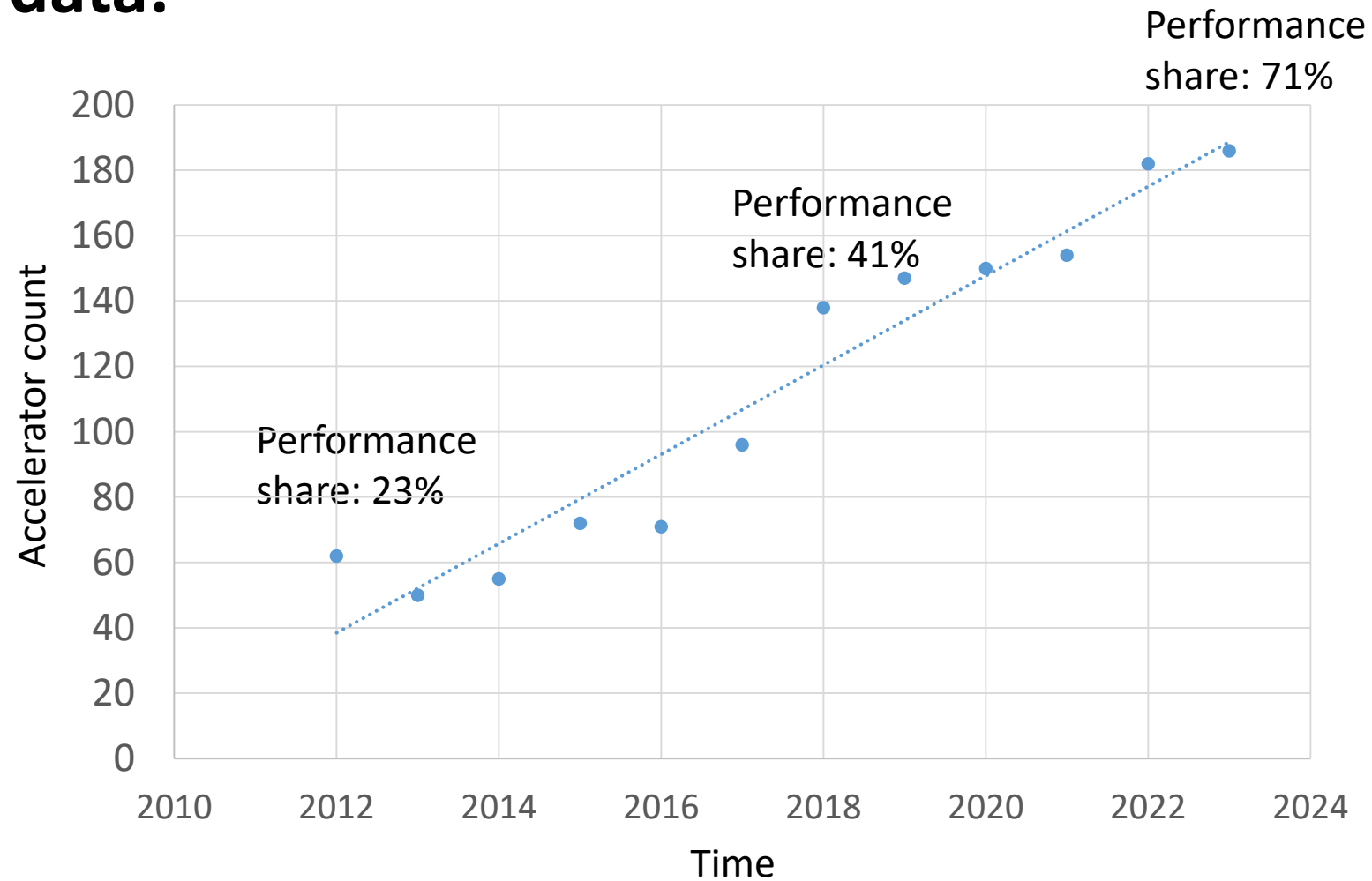
## Subjective perception :



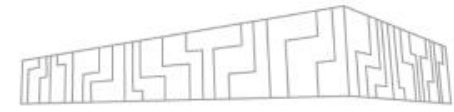
# COMPUTE TRENDS



## TOP500 data:



# COMPUTE TRENDS



## From scalars to FMAs on matrices:

$$\boxed{A} + \boxed{B} = \boxed{C}$$

$$\begin{array}{|c|c|c|c|} \hline A1 & & & A4 \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline B1 & & & B4 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline C1 & & & C4 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|} \hline A1 & & & A4 \\ \hline \end{array} * \begin{array}{|c|c|c|c|} \hline B1 & & & B4 \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline C1 & & & C4 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline D1 & & & D4 \\ \hline \end{array}$$

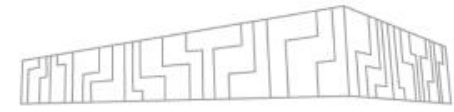
$$\begin{array}{|c|c|c|c|} \hline & & & \\ \hline & A & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} * \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & B & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & C & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & D \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

Independent data streams

CPU:  $2*8*5 = 80$

GPU:  $12*32*8 = 3072$

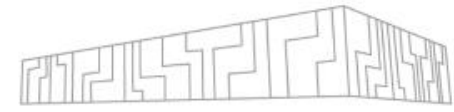
# COMPUTE TRENDS - PROGRAMMING



**From “code it yourself” to “use frameworks”:**

	Code yourself	Use libs	Use Frameworks
Language	FORTRAN, C	FORTRAN, C, PYTHON	PYTHON, etc.
Lib	none	BLAS, LAPACK,TBB	BLAS, LAPACK,TBB
Framework	none	none	PyTorch, etc.

# HYPERQUEUE



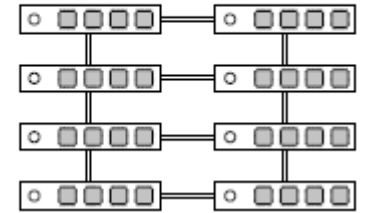
## Problem:

- Large amount of jobs?
- Very large amount of small jobs?
- Only 1 or few cores per job needed?
- Complicated job dependencies?

Many simple tasks



Slurm/PBS Cluster

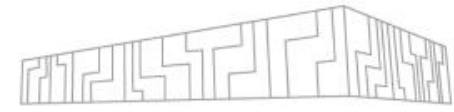


**Answer: Use hq!**

<https://it4innovations.github.io/hyperqueue/stable/>

<https://docs.it4i.cz/general/capacity-computing/#hyperqueue>

# COMPUTE TRENDS - SERVICES



jansik@nb-jan0369: ~/work/lorenz

```
printf("Norm y: %f\n",dnrm2_(&NMAT2,y,&ONEI)/NMAT);  
printf("Norm z: %f\n",dnrm2_(&NMAT2,z,&ONEI)/NMAT);
```

```
//release memory
```

```
free(x);
```

```
free(y);
```

```
free(z);
```

```
free(dx);
```

```
free(dy);
```

```
free(dz);
```

```
free(e);
```

```
free(xe);
```

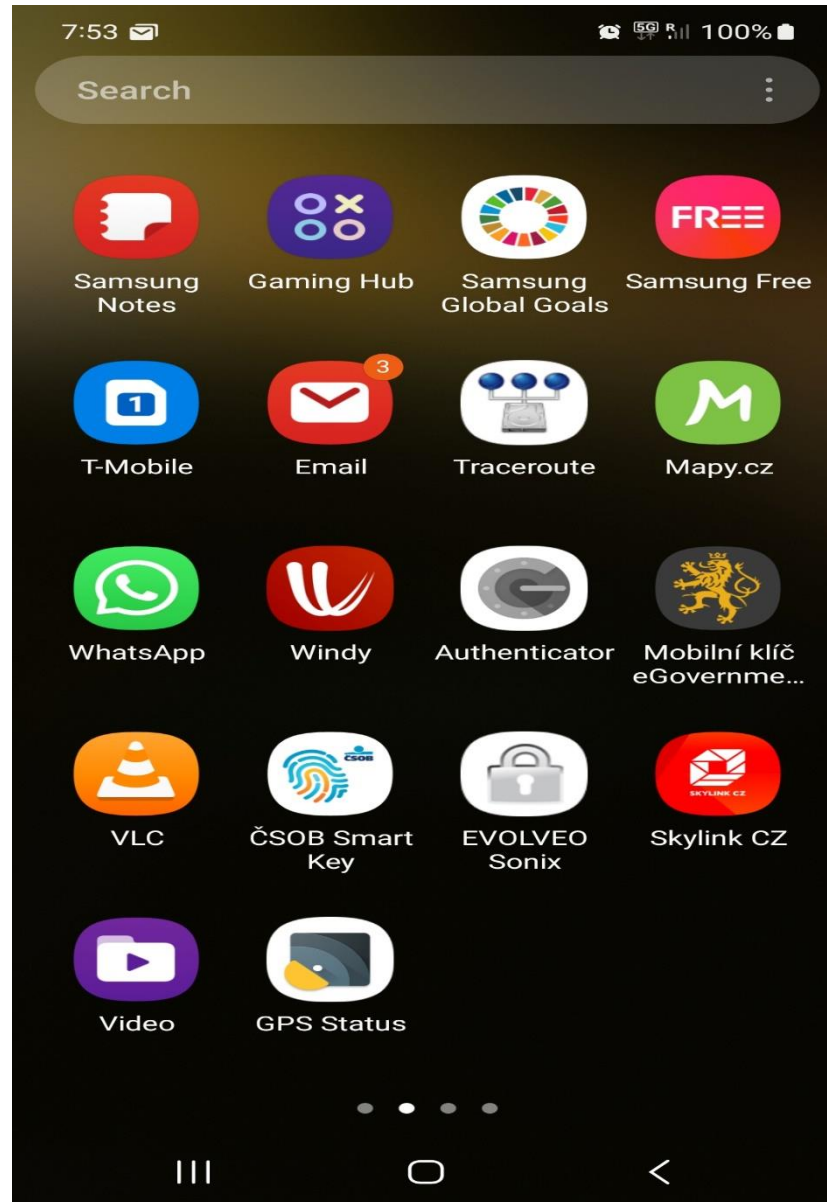
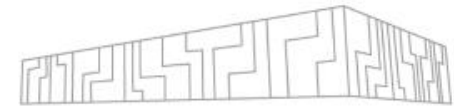
```
return 0;
```

```
}
```

```
jansik@nb-jan0369:~/work/lorenz$
```



# COMPUTE TRENDS - SERVICES





# COMPUTE TRENDS - OPENONDEMAND



The image displays a multi-windowed environment for running a C program on a remote server. The windows include:

- mbi-blas.c (~/.work/lorenz) - GVIM:** A C source code editor showing the Lorenz attractor code. The code includes OpenMP directives for parallelization and memory allocation. The code is as follows:

```
printf("Running %d iterations\nMatrix size: %d\nMemory: %.1f MiB\n", NITER, NMAT, (sizeof(double)*(double)*(2*NMAT2)/1024/1024));

//allocate memory
posix_memalign((void *) &c, (size_t)64, (size_t) sizeof(double)*NMAT2 );
posix_memalign((void *) &z, (size_t)64, (size_t) sizeof(double)*NMAT2 );

//initialize matrices
#pragma omp parallel for default(shared) private(i, rng) schedule(static, 8)
for (i=0; i<NMAT2; i++) {
    c[i]=(double)(pcg32_random_r(&rng)%100000)/100000/NMAT;
    z[i]=(double)0.0;
}

//run mandelbrot inspired iterations
t0 = omp_get_wtime();
for (i=0; i<NITER; i++) {
    //Z=2*Z+C
    dgemv_ (&N, &N, &NMAT, &NMAT, &NMAT, &ONED, z, &NMAT, z, &NMAT, &ONED, c, &NMAT);

    //Z=C+C+Z
    dgemv_ (&N, &N, &NMAT, &NMAT, &NMAT, &ONED, c, &NMAT, c, &NMAT, &ONED, z, &NMAT);

    //C=Z*Z-C
    dgemv_ (&N, &N, &NMAT, &NMAT, &NMAT, &ONED, z, &NMAT, z, &NMAT, &ONED, c, &NMAT);

    //Z=C*C-Z
    dgemv_ (&N, &N, &NMAT, &NMAT, &NMAT, &ONED, c, &NMAT, c, &NMAT, &ONED, z, &NMAT);
}
t = omp_get_wtime();
```
- MATLAB R2021a - academic use:** The MATLAB interface showing the Command Window with the following code:

```
>> na -jst = linspace(-10,10,1000);
xt = exp(-t./10).*sin(5*t);
yt = exp(-t./10).*cos(5*t);
p = plot3(xt,yt,t);
na -jst = linspace(-10,10,1000);
t = linspace(-10,10,1000);
xt = exp(-t./10).*sin(5*t);
```
- Figure 1:** A 3D plot of the Lorenz attractor, showing a complex, chaotic trajectory in a 3D space.
- Mate Terminal:** A terminal window showing the execution of the MATLAB command to run the C program. The output shows the execution of the C program and the resulting Lorenz attractor plot.

```
^Z
[1]+  Stopped                  matlab
[jansik@login1.karolina ~]$ bg
[1]+  matlab &
[jansik@login1.karolina ~]$ ls
20210701-500nodes.log      java.log.41238      ondemand
bin                        jitter.c            Pictures
Desktop                    linuxmint.img       Public
Documents                  MATLAB Add-Ons     Templates
Downloads                  MedeaA              test
intel-2021a-720nodes.log  Medea.settings    ubuntu_gvim.img     Videos
intel-2021a-720nodes-novainge.log  Music              octave-workspace    work
intel-2021a-748nodes.log  octave-workspace
[jansik@login1.karolina ~]$ cd work/
[jansik@login1.karolina work]$ cd lorenz
[jansik@login1.karolina lorenz]$ ls
cpuid.c      lorenz-avx512.s  lorenz-c.c  octave-workspace
cpuid.x      lorenz-avx512.x  lorenz-c.s  README.md
flags        lorenz-avx.s    lorenz-c.x  sde-footprint.txt
jansik-blas.c  lorenz-avx.x    Makefile    sde-mix-out.txt
lorenz-avx2.s  lorenz-blas.c   mbi-blas.c  test.f
lorenz-avx2.x  lorenz-blas.x   mbi-blas.x  token
[jansik@login1.karolina lorenz]$ gvim mbi-blas.c
[jansik@login1.karolina lorenz]$
```
- Configuration Panel:** A panel for configuring OpenMP threads and environment variables. It shows "Number of OpenMP threads: 1" and "Environment Variables: none".

# COMPUTE TRENDS - LEXIS PLATFORM



https://portal.lexis.tech/workflow/airflow/job\_dependency\_example/execution/Test run 1



WORKFLOWS

EUROHPC SU...



EUROHPC DEMO

USER: BRANISLAV JANSIK

LOGOUT

DETAIL PROGRESS LOGS



DATA SETS



ORGANIZATION



PROJECTS



WORKFLOWS



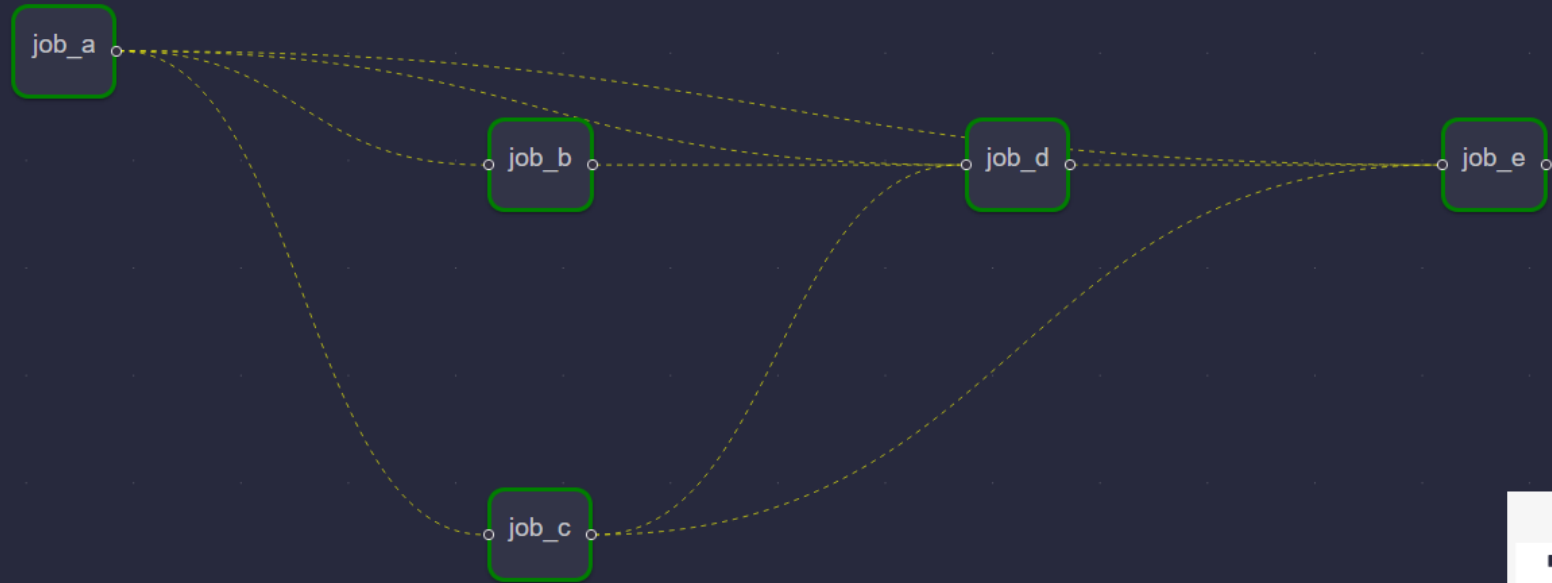
ABOUT LEXIS



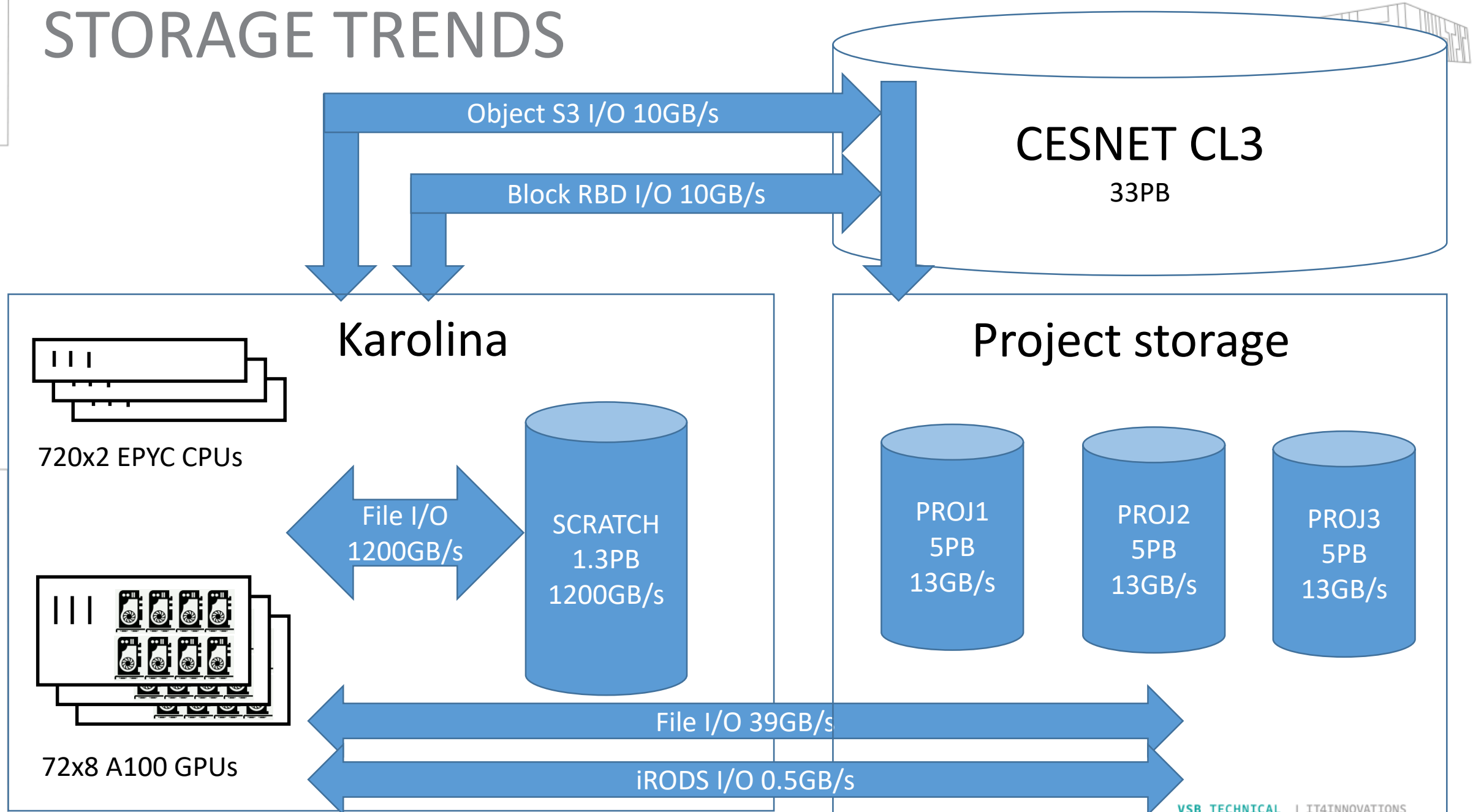
SEND FEEDBACK

v1.4.0-14-03-2023

## Execution Progress



# STORAGE TRENDS





Branislav Jansík  
branislav.jansik@vsb.cz

IT4Innovations National Supercomputing Center  
VSB – Technical University of Ostrava  
17. listopadu 2172/15  
708 00 Ostrava-Poruba, Czech Republic  
www.it4i.cz

VSB TECHNICAL  
UNIVERSITY  
OF OSTRAVA

IT4INNOVATIONS  
NATIONAL SUPERCOMPUTING  
CENTER