Using HPCFS

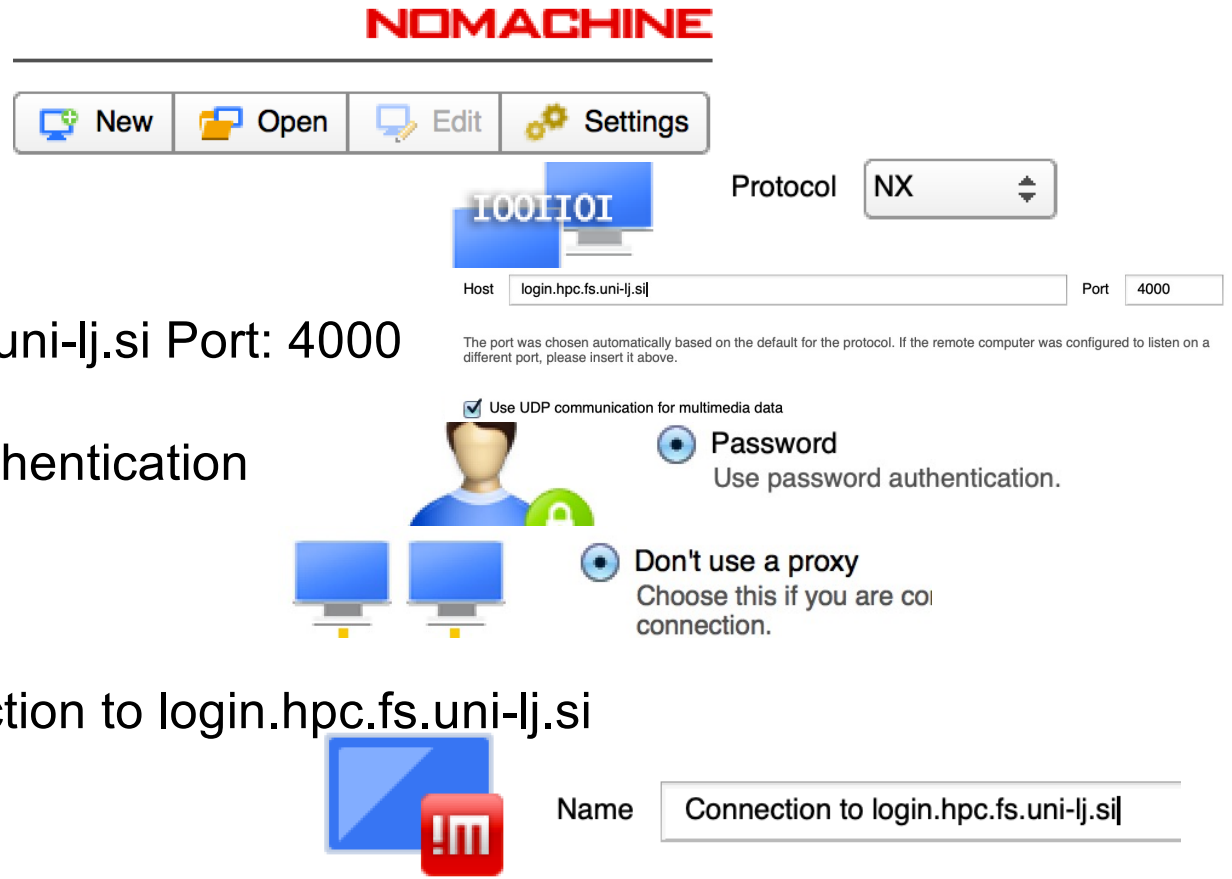
Leon Kos, University of Ljubljana

26 June 2023

Setting up the NoMachine client

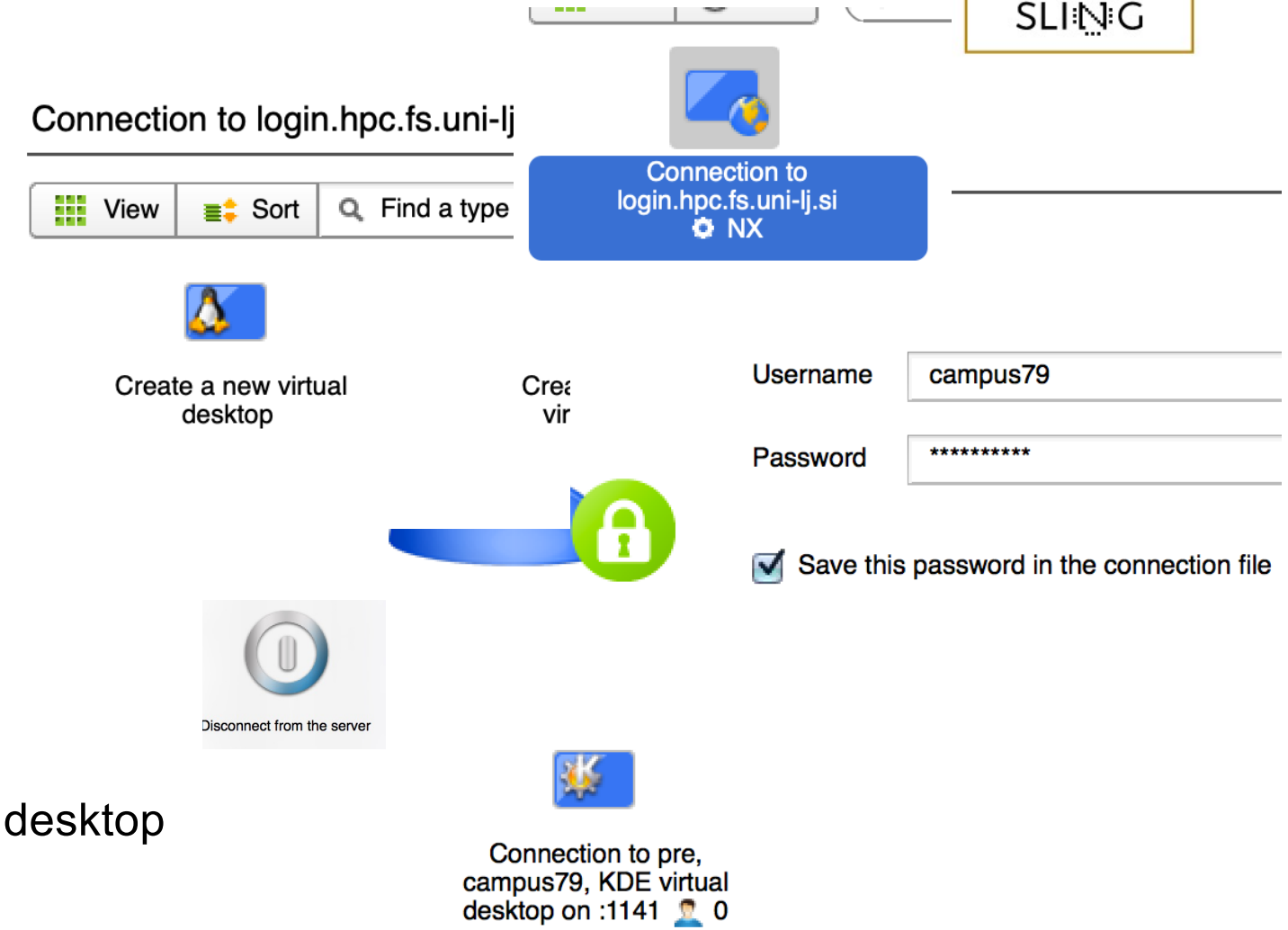
Available for installation at page https://www.nomachine.com/download-enterprise#NoMachine-Enterprise-Client

1. Select New
2. Protocol NX
3. Host: login.hpc.fs.uni-lj.si Port: 4000
4. Use Password authentication
5. Don’t use proxy
6. Done with Connection to login.hpc.fs.uni-lj.si

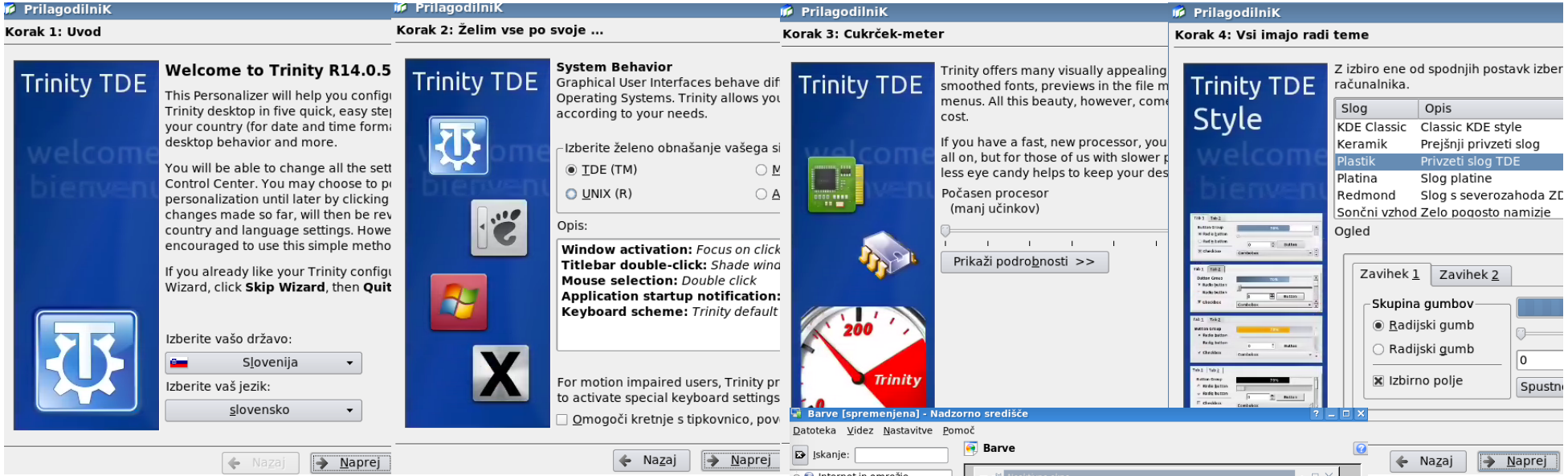


Connecting to HPCFS

1. Select and Connect
2. Use your account credentials
3. Create New desktop once
4. Use the Trinity (KDE) desktop
5. To Disconnect press Ctrl+Alt+T
6. To Reconnect select previous virtual desktop



Tuning desktop with KPersonalizer for remote speed (use less effects=slower processor)



Use Trinity Control Center to setup colors for Non TDE programs: Uncheck “Enforce colors for Non-TDE programs”

Basic HPCFS cluster usage

- Setting GNOME or KDE desktop locale preferences for keyboard, LANG environment
- Using NX client (Disconnect, Terminate, Logout)
- Console commands in Linux
- Editors for programming (emacs, gedit, kate, eclipse, vi, pico, ...) on login only!

Modules (LUA)
- module avail
- module help/info
- module show
- module load/unload
- module list
- module purge

SLURM batch scheduler

Compiled-in OpenMPI support
- `srun --nodes=N --ntasks=n cmd`
- `sbatch script.sh`
- `sinfo`
- `squeue`
- Alias for interactive usage of nodes:

```
alias node='srun -N1 --time=1:00:00 --pty bash -i'
```

Using SLURM (interactivelly) and Message Passing Interface (MPI)


```
[leon@viz mpi]$ module purge && module load foss/2019a
[leon@viz mpi]$ cat hello.f90
program hello
   use mpi
   integer rank, size, ierror, strlen, status(MPI_STATUS_SIZE)
   character(len=MPI_MAX_PROCESSOR_NAME) :: hostname


   call MPI_INIT(ierror)
   call MPI_COMM_SIZE(MPI_COMM_WORLD, size, ierror)
   call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierror)
   call MPI_GET_PROCESSOR_NAME( hostname, strlen, ierror )
   print*, trim(hostname), rank, size
   call MPI_FINALIZE(ierror)
end
[leon@viz mpi]$ mpif90 hello.f90
[leon@viz mpi]$ LD_PRELOAD= srun -n 4 --tasks-per-node=2 --kill-on-bad-
exit --partition=haswell ./a.out
 cn80           2           4
 cn79           0           4
 cn80           3           4
 cn79           1           4
```

OpenMP

```
#include <stdio.h>
#include <math.h>
#define N 1000000
int main()
{
  double area = 0.0;
  #pragma omp parallel for reduction(+:area)
  for(int i = 0; i < N; i++)
    {
      double x = (i+0.5)/N;
      area += sqrt(1.0 - x*x);
    }
  printf("Pi : %14lf\n", 4.0*area/N);
  return 0;
}
[leon@cn36 pi]$ module purge && module load foss/2019a
[leon@cn36 pi]$ gcc -fopenmp pi-openmp.c  -lm -o pi-openmp
[leon@cn36 pi]$ OMP_NUM_THREADS=4 ./pi-openmp
```

http://hpc.fs.uni-lj.si

Thanks!

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia

EuroHPC
Joint Undertaking