



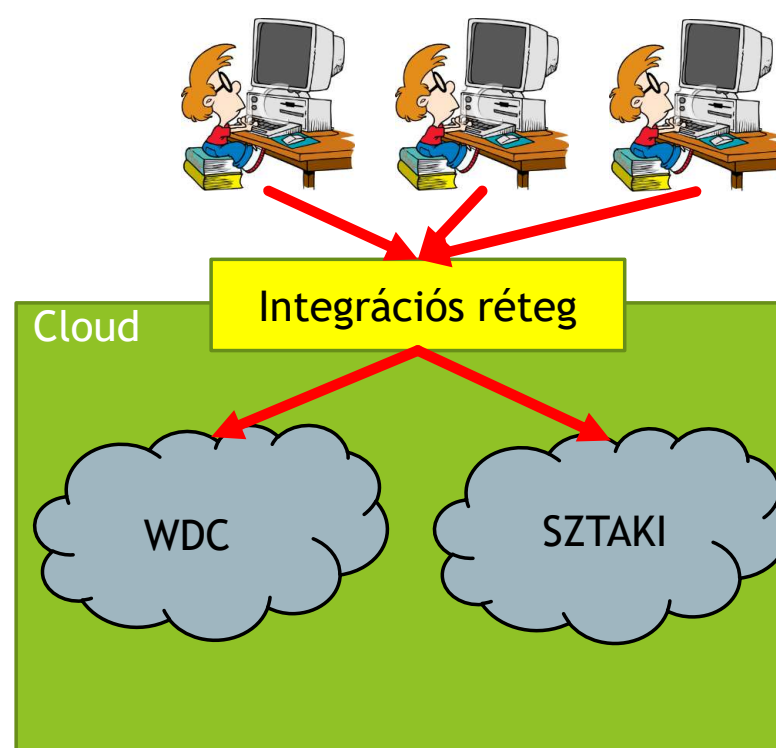
Az ELKH Cloud alkalmazási lehetőségei HTC (High Throughput Computing) és HPC (High Performance Computing) területen



Kacsuk Péter
SZTAKI

Az MTA és ELKH Cloud kapacitásának összehasonlítása

	MTA	ELKH
vCPU (max)	1368	5900
GPU core	12	76
vGPU (max)	12	2060
RAM (TB)	3,25	28
SSD storage (TB)	0	338
HDD storage (TB)	527	1248
Tensor GPU teljesítmény (PFLOPS)	~0	13.7
Lebegőpontos GPU teljesítmény (PFLOPS)	~0	1.2
Hálózati sávszélesség (Gbps)	10	100



Az ELKH Cloud koncepciója

- ▶ **Az ELKH Cloud egy e-infrastruktúra keretrendszer**, ami
 - ▶ Nem felhasználó-orientált, hanem **projekt-orientált**
 - ▶ Elsődlegesen projektek regisztrálnak
 - ▶ A felhasználóknak nem a cloudra, hanem a projekthez kell regisztrálniuk
 - ▶ Egy projekt **kvótát** kap és ezen belül olyan e-infrastruktúrát épít fel a felhőben, amelyet csak akar
 - ▶ Pl. csinálhat Hadoop v. Spark klasztert, de azt csak az ő felhasználói éri el

Az ELKH Cloud koncepciója



- ▶ e-infrastruktúrát felépíteni a felhőben nem könnyű, ezért az ELKH Cloud projekt
 - ▶ A tipikus, gyakran használt e-infrastruktúrák kiépítéséhez **referencia architektúrákat** biztosít, amikből a kívánt infrastruktúra gyorsan felépíthető
 - ▶ A nem tipikus e-infrastruktúrák felépítéséhez közvetlen segítséget nyújt a felhasználóknak, majd az így létrejött e-infrastruktúrából referencia architektúrát csinál
- ▶ A referencia architektúrák tárolására és elérésére repozitóriumot biztosítunk, ahonnan minden ELKH felhasználó elérheti és használhatja ezeket

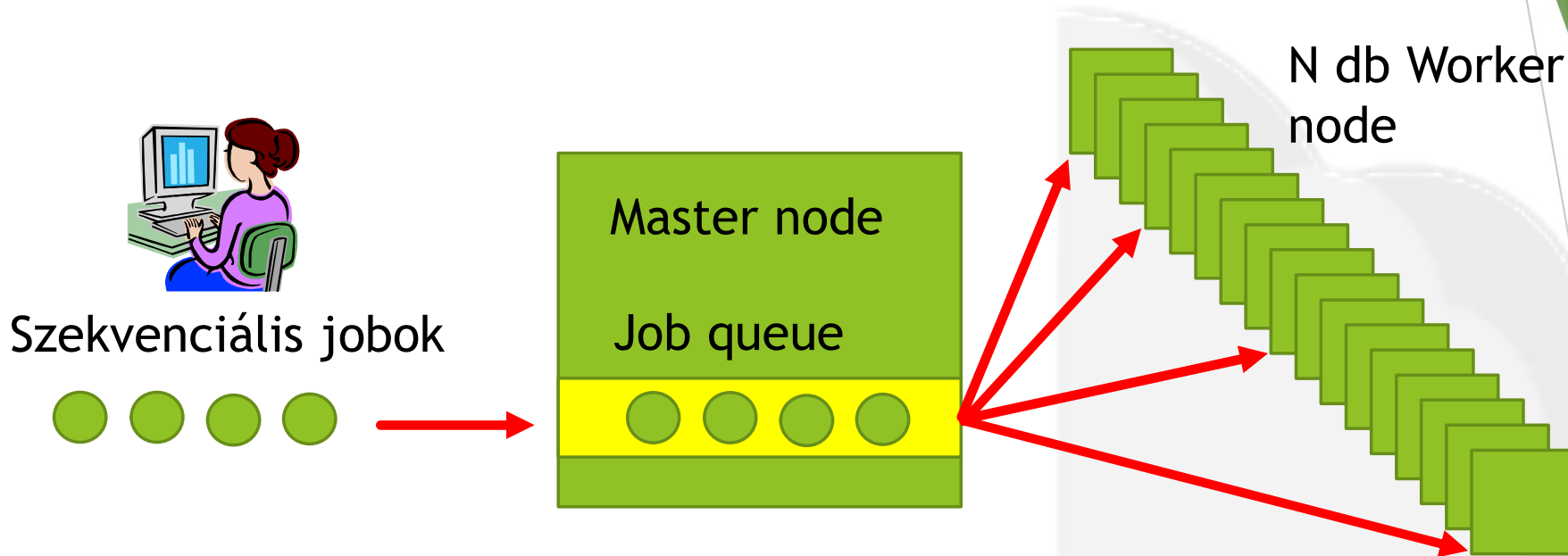
A párhuzamos programvégrehajtás 2 fő lehetősége

- ▶ **HTC (High Throughput Computing)** párhuzamosítás
 - ▶ Programok közötti párhuzamosítás: Egymástól független, párhuzamosan futó programok
- ▶ **HPC (High performance computing)** párhuzamosítás
 - ▶ Programon belüli párhuzamosítás: Egy programon belül vannak párhuzamosan végrehajtható részprogramok, amik nem függetlenek egymástól, időnként kommunikálnak egymással

HTC (High Throughput Computing) az ELKH Cloudon

- ▶ Cél: Egységnyi idő alatt **minél több program** lefuttatása.
- ▶ Módszer: HTC párhuzamos végrehajtás, azaz
 - ▶ egyidejűleg **sok egymástól független program** futtatása
 - ▶ Amik egyidejűleg **sok egymástól független processzort** használnak
 - ▶ Tipikus elnevezések:
 - ▶ lazán csatolt párhuzamosság
 - ▶ “Parameter sweep” párhuzamosság
 - ▶ “Embarrassingly parallel” program végrehajtás

A HTC párhuzamosság elve



A worker node-ok lehetnek virtuális gépek és konténerek

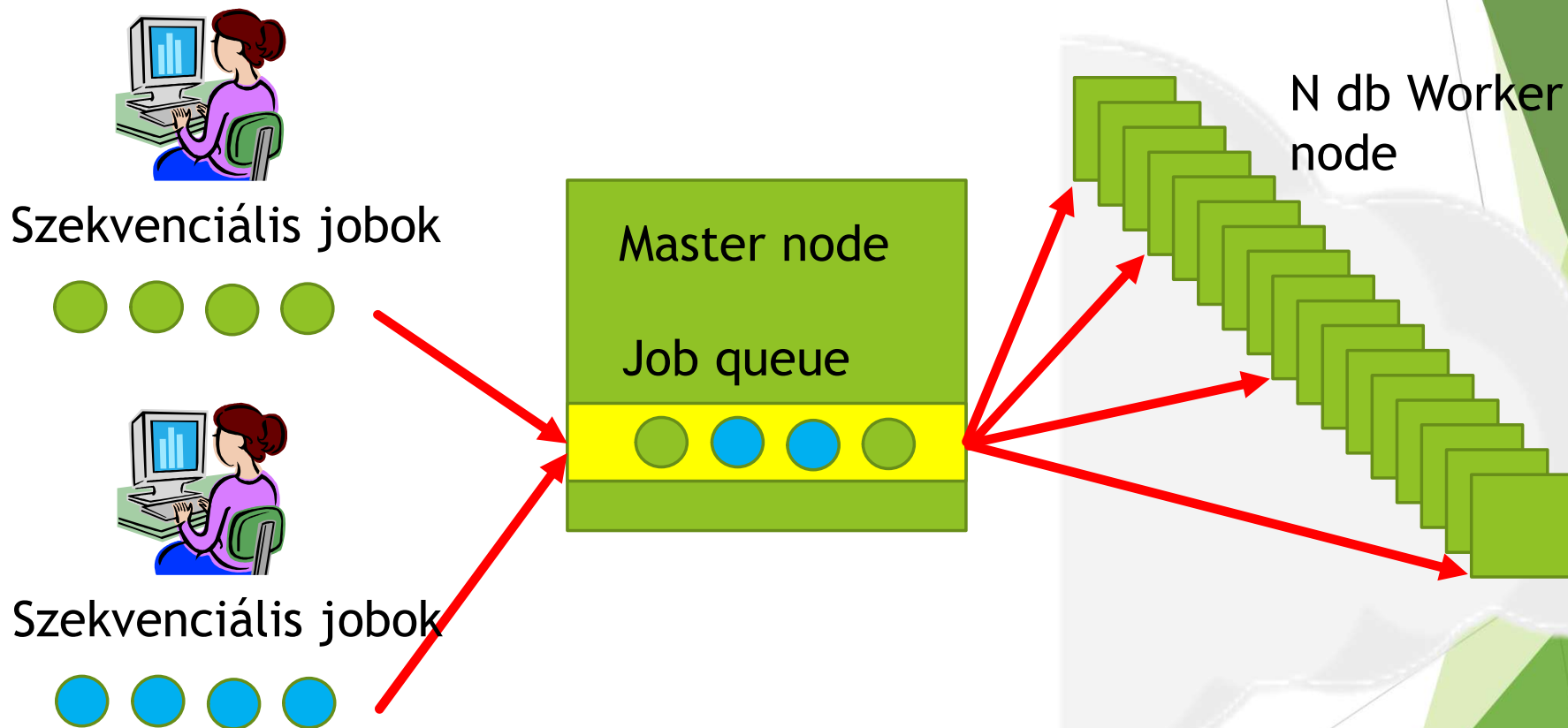
HTC job kezelő rendszer (Condor, DIRAC, **SLURM**, stb.)

HTC párhuzamosság és a cloud architektúra



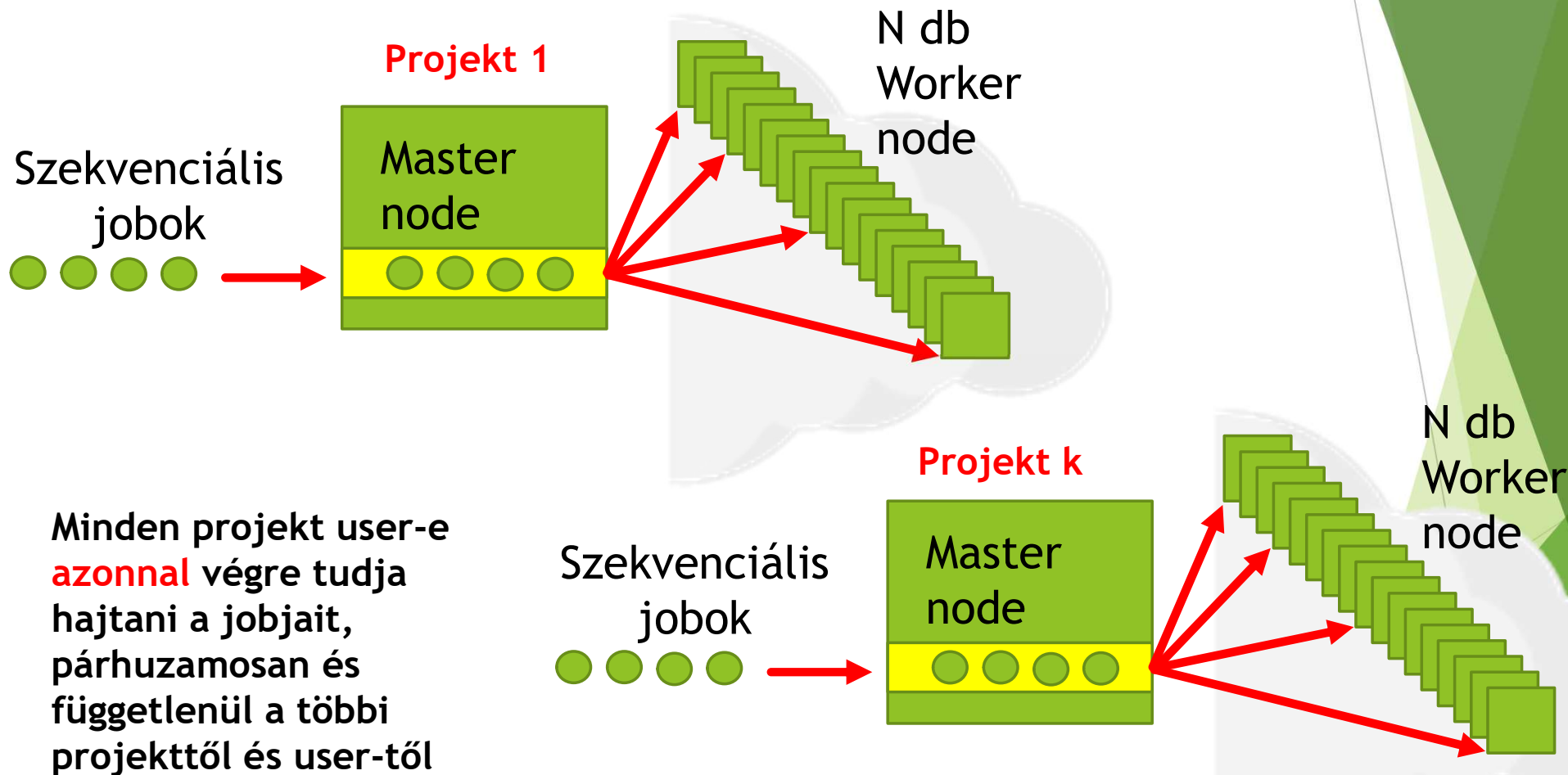
- ▶ A cloud architektúrában a processzorok kommunikációjára nincs sebességre optimalizált összekötő hálózat
- ▶ E miatt (és még más okokból is) a számítási cloud sokkal olcsóbb, mint egy szuperszámítógép
- ▶ Mivel a HTC párhuzamosságban a jobok végrehajtása független a worker node-ok nem kommunikálnak és így az optimalizált összekötő hálózatra nincs is szükség
- ▶ A cloud tehát tökéletesen alkalmas a HTC párhuzamosság megvalósítására és a szuperszámítógépnél sokkal olcsóbban képes erre
- ▶ Ha tehát HTC jellegű feladatunk van, akkor használjunk cloud-ot

A HTC párhuzamosság több felhasználóval egy projekten belül



Előfordulhat, hogy egy usernek várnia kell a másik user miatt!

HTC párhuzamosság projektenként az ELKH Cloudban



HPC (High performance computing)

- ▶ Cél: **egyetlen program** futtatásának felgyorsítása, hogy a lehető legrövidebb idő alatt fusson le
- ▶ A program több összetartozó programrészletből (procesz, v. taszk) áll, amik
 - ▶ párhuzamosan futnak
 - ▶ időnként kommunikálnak egymással
- ▶ A fenti programokat **párhuzamos programok**nak hívják
- ▶ Két fő fajtájuk van:
 1. Ahol egymásnak küldött üzenetekkel kommunikálnak a processzek (típusosan **MPI programok**)
 2. Ahol közös memórián keresztül kommunikálnak a processzek (típusosan **OpenMP programok**)

HPC (High performance computing)

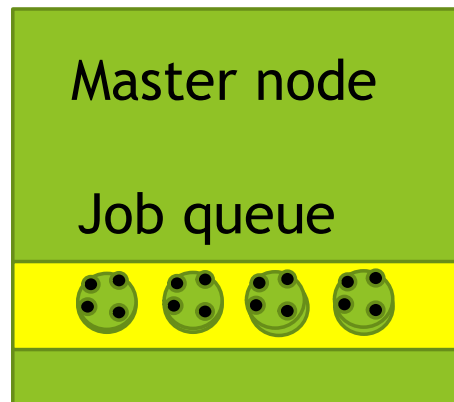
- ▶ A hatékony párhuzamos végrehajtáshoz szükséges:
 - ▶ egyidejűleg **sok koordinált processzor** alkalmazása
 - ▶ **optimalizált kommunikációs hálózat** a processzorok között
- ▶ Tipikus infrastruktúra: **szuperszámítógép**, ami a fentiek miatt sokkal drágább, mint a cloud



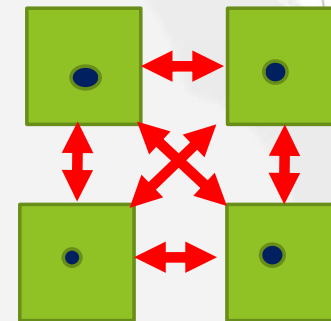
A HPC párhuzamosság elve

A worker node-okon szétterülnek a párhuzamos program taszkjai. Működésük már nem független egymástól, hiszen kommunikáltak.

Párhuzamos jobok

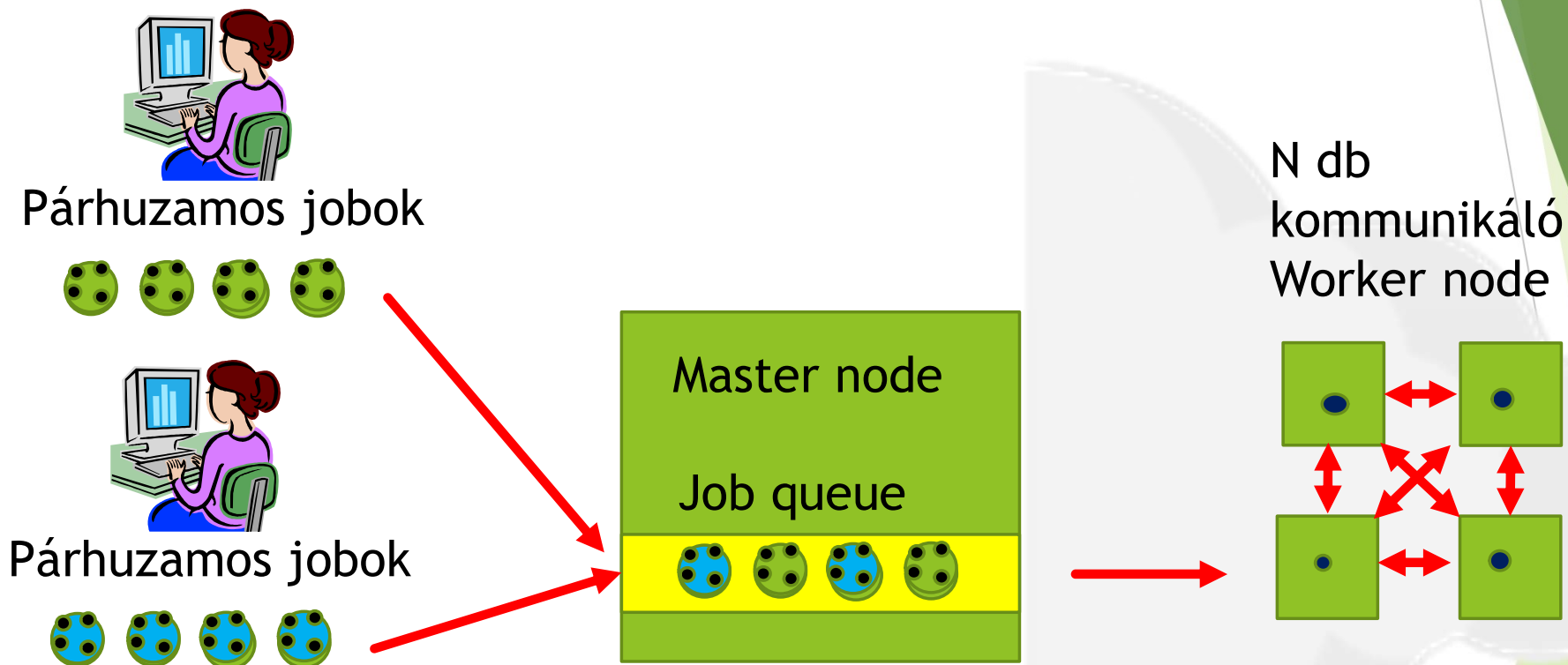


N db
kommunikáló
Worker node



Ahhoz, hogy a kommunikáció az egyes taszkok között, amik külön processzorokon futnak gyors legyen, optimalizált kommunikációs hálózatot alkalmaznak a HPC gépekben

A HPC párhuzamosság elve



Workload manager/ütemező (SLURM, PBS, stb.) allokálja a szükséges erőforrásokat és osztja szét a párhuzamos jobokat ezekre az erőforrásokra

HPC és workload managerek

- ▶ A HPC gépek a gyors interprocesszor kommunikációra és a nagyszámú processzorra vannak optimalizálva, ezért sokkal drágábbak, mint a cloud gépek
- ▶ A HPC gépek célja, hogy a nagyon nagy számításigényű feladatokat hajtsák végre, amik egy normál számítógépen akár több száz évig futnának (pl. emberi agy feltérképezése, csillagászati számítások, stb.)
- ▶ Ahhoz, hogy a drága HPC gépeket jól kihasználjuk és egyszerre minél több felhasználó párhuzamos programja tudjon futni rajta workload manager programot alkalmaznak
- ▶ A felhasználók nem közvetlenül kapják meg az erőforrásokat, hanem a workload manageren keresztül, hogy egyszerre minél több felhasználót lehessen kiszolgálni.
- ▶ A felhasználóknak viszont ki kell várni, amíg a job-uk sorra kerül és megkapja a szükséges erőforrásokat. **Ez akár több hét is lehet.**

HPC gépek és a cloudok összehasonlítása



- ▶ **HPC gépek**et nagy számításigényű párhuzamos programok végrehajtására használunk (Magyarország-on ezeket a KIFÜ szolgáltatja)
- ▶ A párhuzamos programokat a HPC gép nagy sebességgel és gyorsan képes lefuttatni (1.000-10.000 - szerez gyorsítás is elérhető)
- ▶ De a HPC gép terheltségétől függően akár több hétig is várni kell, hogy a párhuzamos programunk elinduljon a HPC gépen

- ▶ **Cloud**on HTC jellegű alkalmazások tudnak optimálisan futni, ahol sok független szekvenciális program hajtódhat végre egyidejűleg különböző VM-ekben vagy konténerekben (100-1.000 - szerez gyorsítás is elérhető)
- ▶ A cloud erőforrás projektekre előre le van osztva. Ez sokkal kisebb erőforráshalmaz, mint amit HPC gépen lehet elérni. De ezeken a projekt tagok programjai azonnal elindulhatnak.

Workload managerek HPC gépeken és cloudokon



- ▶ Mind a HPC, mind a cloud gépeken célszerű **workload manager**t használni, ha több felhasználó egyidejű kiszolgálása a cél
- ▶ Az ELKH Cloudon ez a **SLURM** és a mai tájékoztató eseményen ezt és ennek használatát kívánjuk bemutatni
- ▶ A SLURM használata az ELKH Cloudon megnyitja az utat a **cloud HPC-szerű használata** irányába is, de ezzel majd egy későbbi alkalommal fogunk részletesebben foglalkozni
- ▶ Az ELKH Cloud kibővítése megtörtént. Az új, bővített cloud kapacitása már lehetővé teszi a HPC jellegű alkalmazást is főleg a megnövekedett GPU kapacitás miatt.

További információk

- ▶ Web: <https://science-cloud.hu/>
- ▶ GYIK: <https://science-cloud.hu/gyakran-ismetelt-kerdesek>
- ▶ E-mail: info@science-cloud.hu



ELKH Cloud

Köszönöm a figyelmet!

