



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

Farkas Attila
SZTAKI LPDS



ELKH | Eötvös Loránd
Research Network

Farkas Attila

- SZTAKI – tudományos segédmunkatárs, kutató
 - PERL – Párhuzamos és Elosztott Rendszerek Kutatólaboratórium
- farkas.attila@sztaki.hu

A felhők általános előnyei

- **Könnyű felhasználhatóság**
 - Kulcsrakész nagy kapacitású hardver-infrastruktúra és speciális szoftverkörnyezetek
 - Elkerülhetőek egy saját rendszer beszerzésével, kialakításával és üzemeltetésével járó terhek
- **Költséghatékonyság**
 - A virtualizáció és skálázhatóság lehetővé teszi az eltérő és folyamatosan változó igények hatékony kiszolgálását
 - Központosított, professzionális informatikai menedzsment csökkenti a fajlagos költségeket

Az ELKH Cloud további előnyei

- **Közösségi megközelítés**
 - A felhő fejlesztői, üzemeltetői és felhasználói is a magyarországi tudományos közösség tagjai
 - A hazai környezethez alkalmazkodó megoldások és felhasználói támogatás
 - Hazai kutatási eredmények gyakorlatba ültetésével
- **Nyílt tudomány (Open Science) irányelvek**
 - Nyílt forráskódú megoldások mind az operációs rendszerek, mind a felhő alapszolgáltatásai, mind a kutatási szoftvereszközök vonatkozásában
 - Federált, osztott megközelítés – a szuverenitás szem előtt tartása mellett már 2015-ben

***2015: Európai Nyílt Tudományos Felhő
kezdemenyezés***

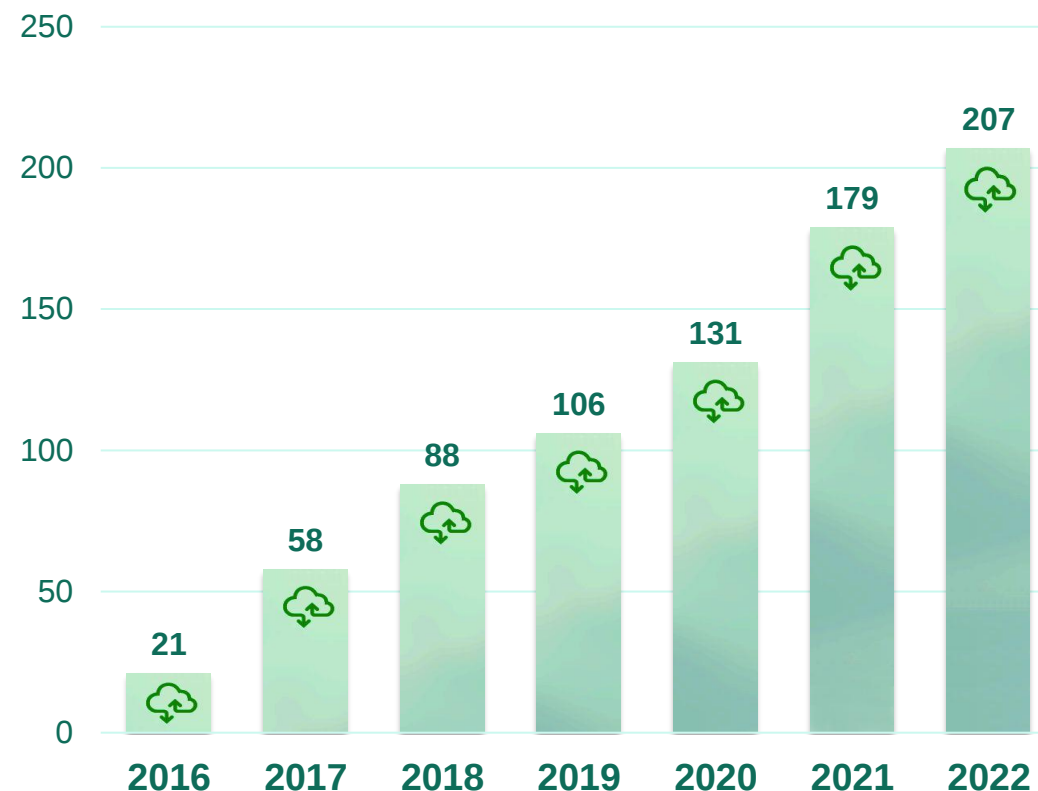


- 2016-ban az MTA égisze alatt a SZTAKI és a Wigner FK elindítja az MTA Cloud szolgáltatást
- Folyamatosan növekvő igények a mesterséges intelligencia, valamint az adatelvű és nyílt tudomány térnyerésével
- 2020-ban az ELKH Irányító Testületének döntése: 1 Mrd Ft támogatás biztosítása a hazai tudományos felhő kapacitásainak és szolgáltatásainak jelentős bővítésére

Összesen igényelt projektek száma

Aggregált adatok évenkénti lépésekben

(db)



Az ELKH Cloud projekt eredményei számokban



- Az ELKH Cloud projekt eredményeinek további mérőszámai:
- **Az ELKH Cloud kapacitásának bővülése**
- A projektek eloszlása a magyarországi kutatóhelyek között
- A szolgáltatás kihasználtsága

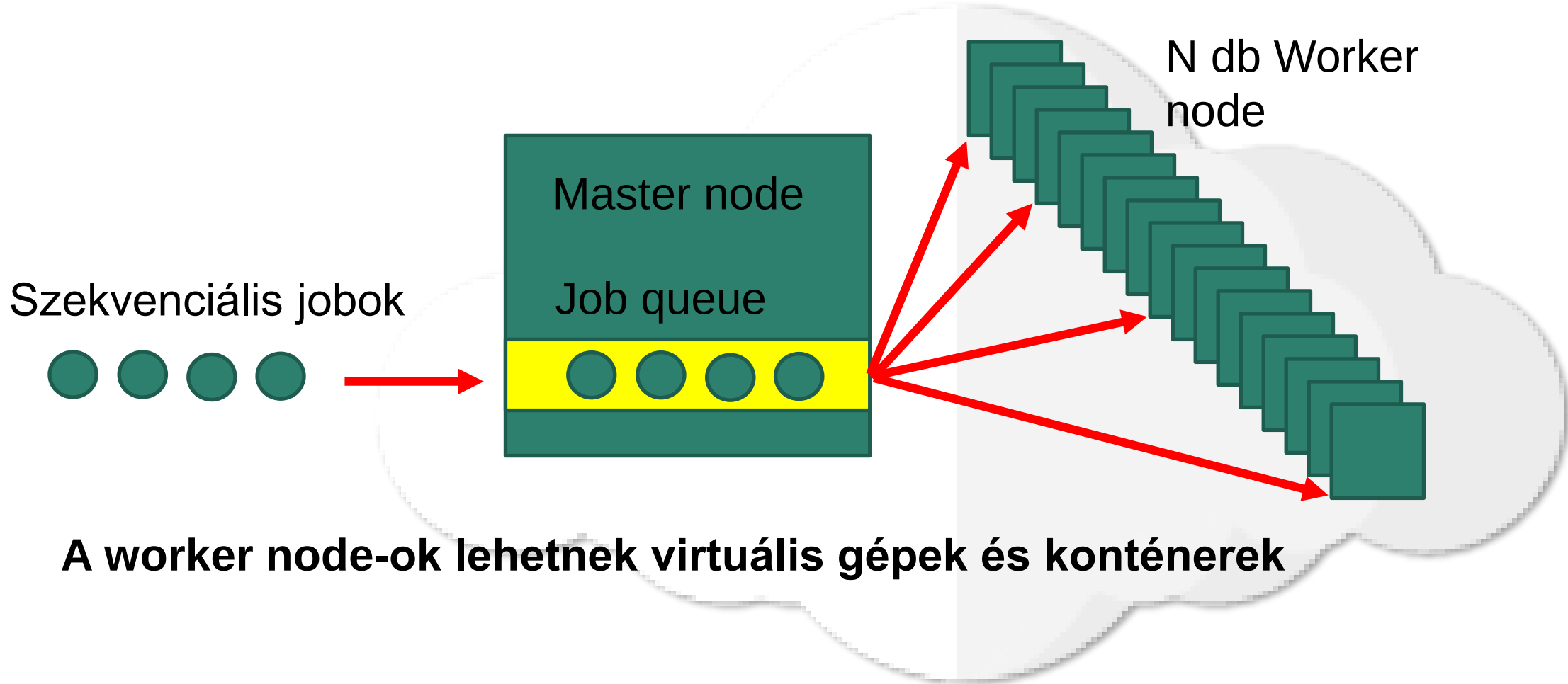
	MTA Cloud	ELKH Cloud
vCPU	1356 db	5900 db
vGPU	12 db	512 db*
RAM	3.25 TB	28 TB
SSD	0 TB	338 TB
HDD	527 TB	1250 TB
Hálózati kapacitás	10 GBPS	100 Gbps

*elméleti maximum 2060 db

- **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

- 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

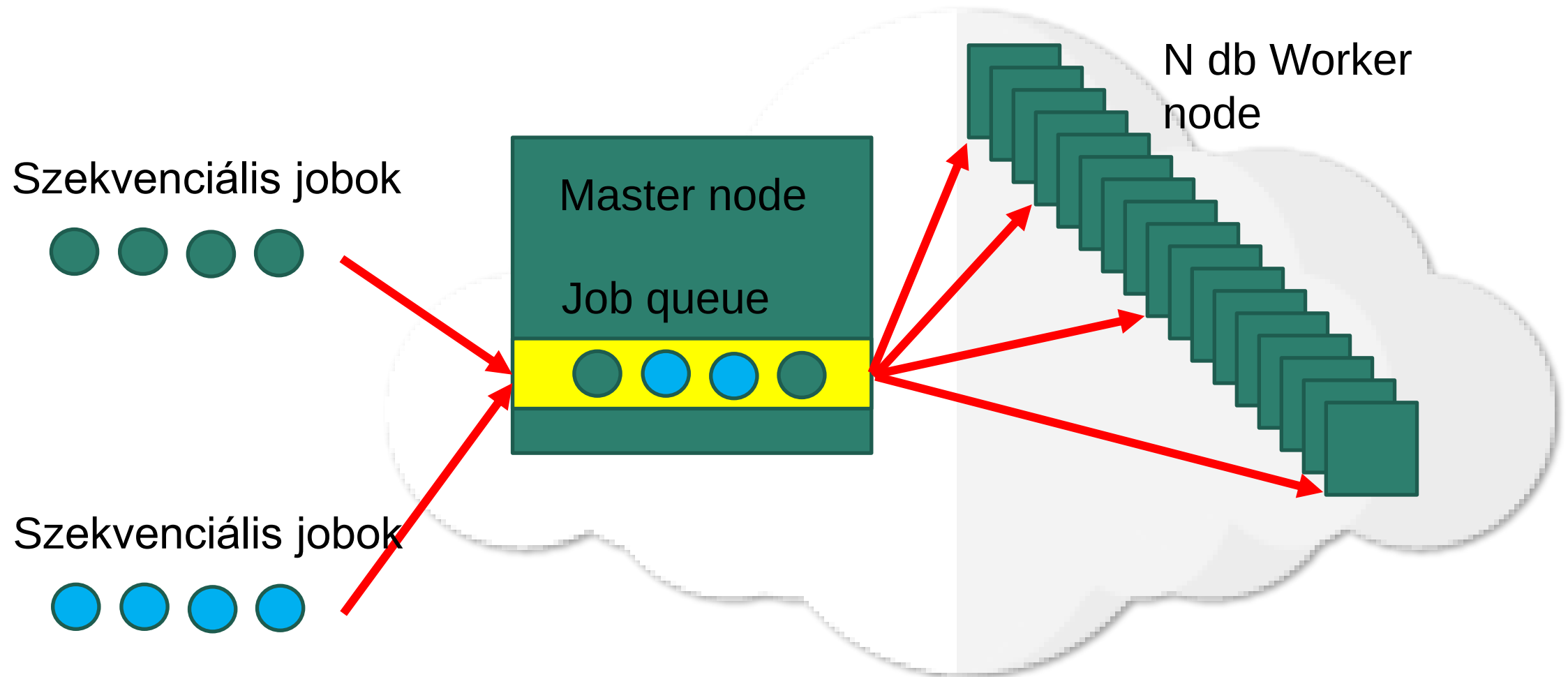


A worker node-ok lehetnek virtuális gépek és konténerek

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

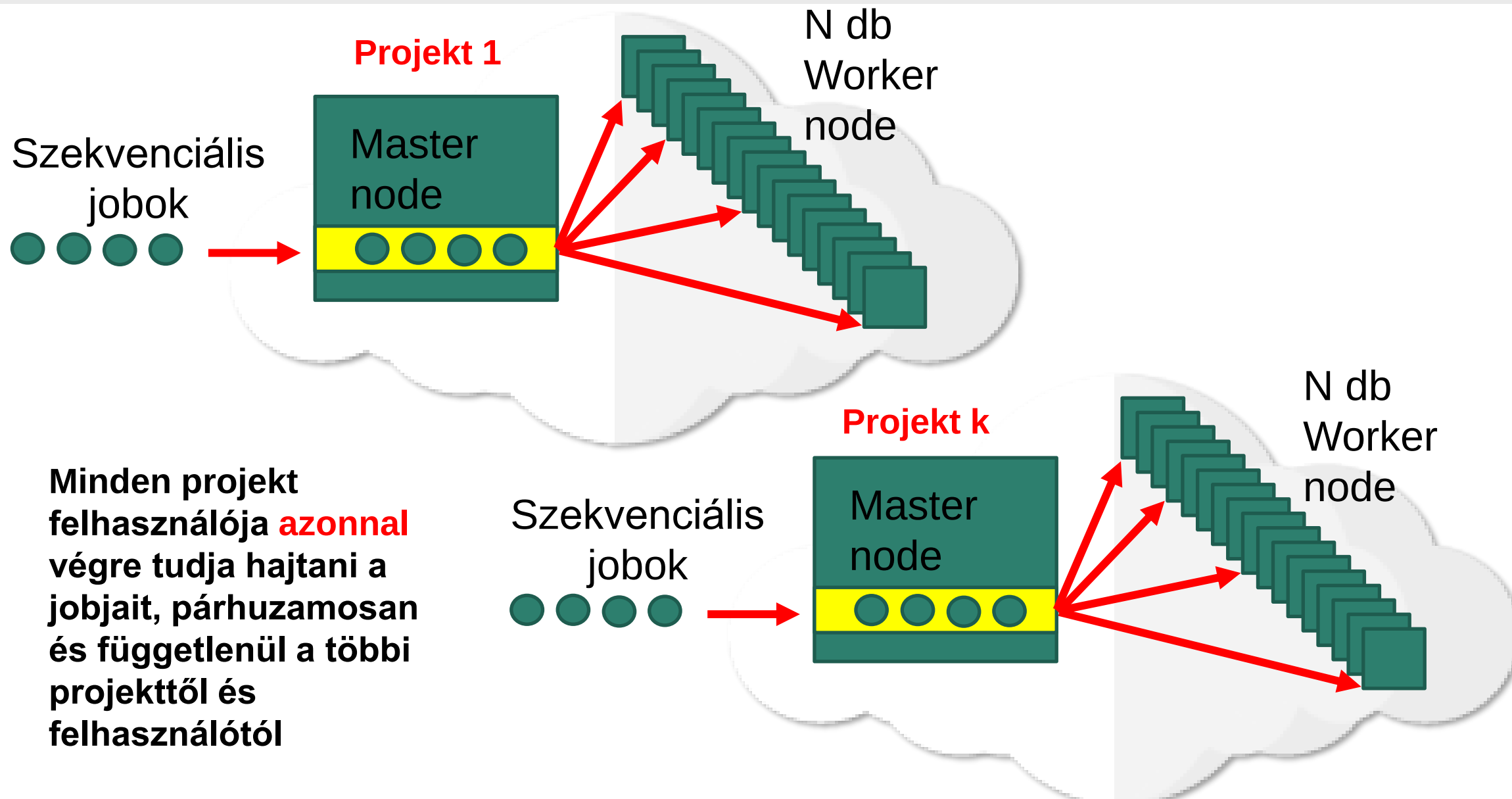
- A számítási felhő architektúrában a processzorok kommunikációjára nincs sebességre optimalizált összekötő hálózat
- E miatt (és még más okokból is) a számítási felhő sokkal olcsóbb, mint egy szuperszámítógép
- Mivel a HTC párhuzamosságban a jobok végrehajtása független a worker node-ok nem kommunikálnak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

A HTC párhuzamosság több felhasználóval egy projekten belül



Előfordulhat, hogy egy felhasználónak várnia kell a másik felhasználó miatt!

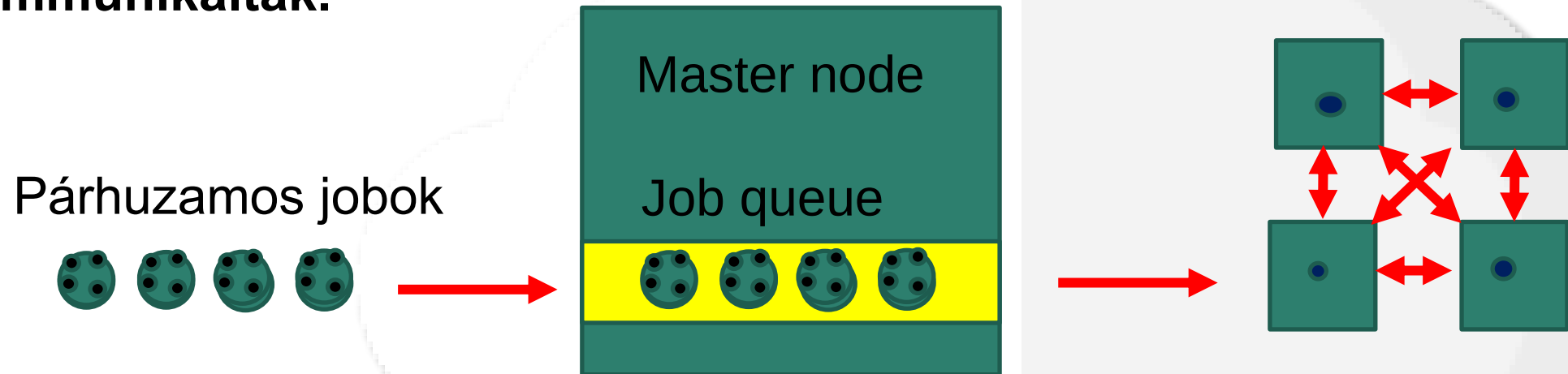
HTC párhuzamosság projektenként az ELKH Cloudban



- Cél: **egyetlen program** futtatásának felgyorsítása, hogy a lehető legrövidebb idő alatt fusson le
- A program több összetartozó programrészletből (process, v. task) 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 (típusan **MPI programok**)
 2. Ahol közös memórián keresztül kommunikálnak a processzek (típusan **OpenMP programok**)

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

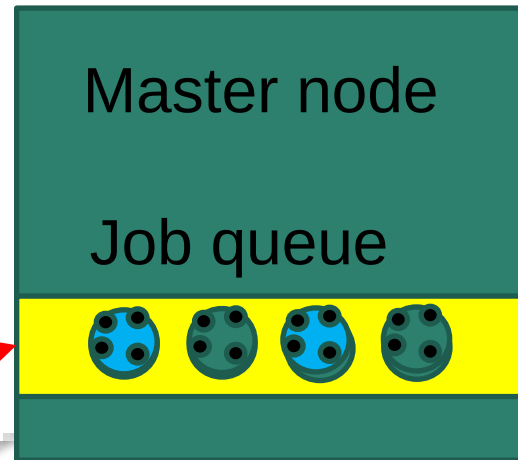


Ahhoz, hogy a kommunikáció az egyes taszkok között, amik külön processzorokon futnak gyors legyen, optimalizált kommunikációs hálózatot alkalmaznak a HPC gépekben

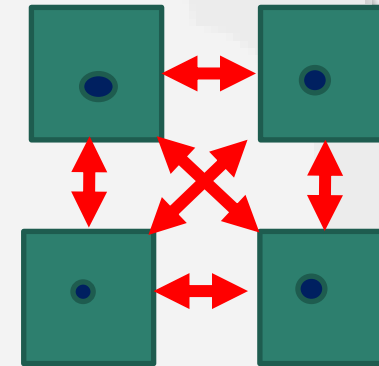
Párhuzamos jobok



Párhuzamos jobok



N db
kommunikáló
Worker node



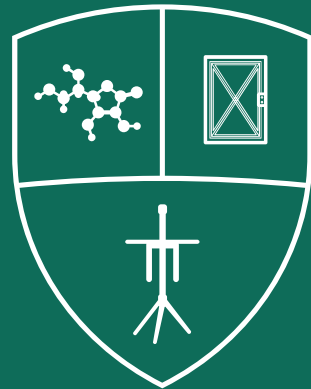
Workload manager/ütemező (SLURM, PBS, stb.) alokálja a szükséges erőforrásokat és osztja szét a párhuzamos jobokat ezekre az erőforrásokra

- 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 kiszolgáni.
- 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**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 – szeres gyorsítás is elérhető)
- De a HPC gép terheltségétől függően akár több hétig is várni kell, hogy a párhuzamos programunk elinduljon a HPC gépen
- **Számítási felhőn** HTC jellegű alkalmazások tudnak optimálisan futni, ahol sok független szekvenciális program hajtható végre egyidejűleg különböző VM-ekben vagy konténerekben (100–1.000 – szeres gyorsítás is elérhető)
- A számítási felhő erőforrás projektekre előre le van osztva. Ez sokkal kisebb erőforráshalmaz, mint amit HPC gépen lehet elérni. De ezeken a projekt tagok programjai azonnal elindulhatnak.

- Mind a HPC, mind a számítási felhő 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 **felhő HPC-szerű használata** irányába is, de ezzel majd egy későbbi alkalommal fogunk részletesebben foglalkozni
- Az ELKH Cloud kibővítése megtörtént. Az új, bővített felhő kapacitása már lehetővé teszi a HPC jellegű alkalmazást is főleg a megnövekedett GPU kapacitás miatt.

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



ELKH

Eötvös Loránd
Research Network

Köszönöm a figyelmet!

www.elkh.org