

MTA Cloud a tudományos alkalmazások támogatására



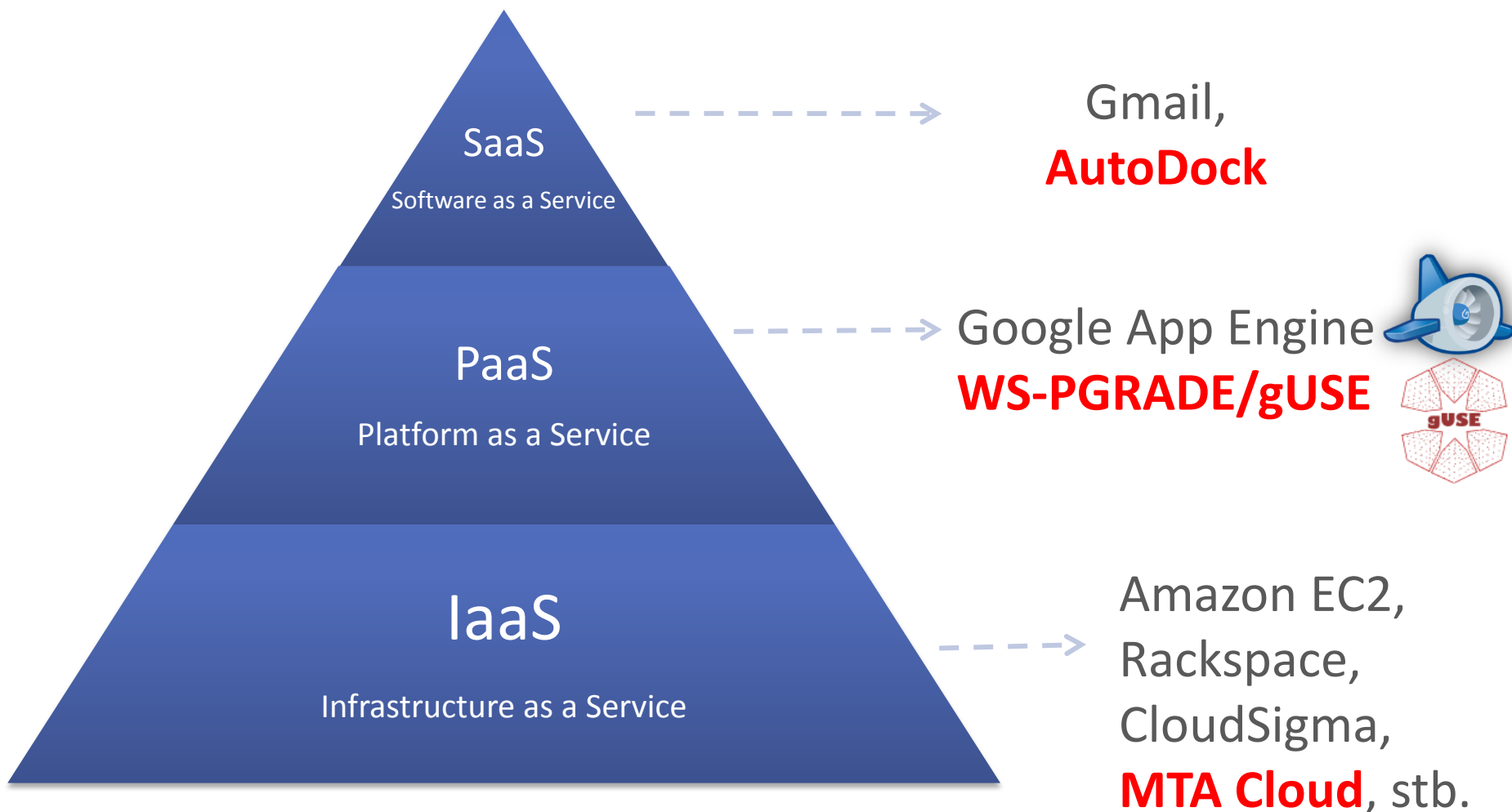
Kacsuk Péter
MTA SZTAKI

Kacsuk.Peter@sztaki.mta.hu

- Guilherme Galante et al.: An Analysis of Public Clouds Elasticity in the Execution of Scientific Applications: a Survey, Journal of Grid Computing, vol. 14, No. 2, 2016, pp. 193-216
 - <http://www.springer.com/computer/communication+networks/journal/10723>
- EU FP7 CloudSME project: Cloud based Simulation platform for Manufacturing and Engineering
 - <http://cloudsme.eu/>
- EU FP7 SCI-BUS project: [SCientific gateway Based User Support](#)
 - <http://www.sci-bus.eu/>

A felhő szolgáltatások 3 szintje

Példák



- Kétféle skálázhatóság:
 - **Vertikális**: dinamikusan változik a VM mérete (CPU core száma, memória, tároló) – általában nem támogatott
 - **Horizontális**: a virtuális gépek száma dinamikusan változik – ezt lehet különböző eszközökkel támogatni, ez **egyfajta párhuzamos végrehajtást feltételez a felhőben**



Erről szól ez az előadás

Cél: Nagy adatmennyiség feldolgozása

1. Egyszerű szekvenciális algoritmus végrehajtása
 1. párhuzamosítás nélkül
 2. HTC (high throughput computing) párhuzamosítással. Tipikus elnevezések:
 1. lazán csatolt párhuzamosság
 2. Parameter söprés (sweep) párhuzamosság
 3. “Embarrassly parallel” párhuzamosság
2. HPC (high performance computing) algoritmus végrehajtása
 1. MPI alkalmazás
 2. OpenMP alkalmazás
3. Docker alkalmazás végrehajtása
 1. párhuzamosítás nélkül
 2. Párhuzamosítással docker klaszteren
4. Mapreduce/Hadoop alkalmazás végrehajtása
5. Pipeline alkalmazás végrehajtása
 1. Egyedi adattal működő pipeline
 2. Adat folyam (data stream) pipeline
6. Workflow alkalmazás (a fentiek kombinációja) végrehajtása

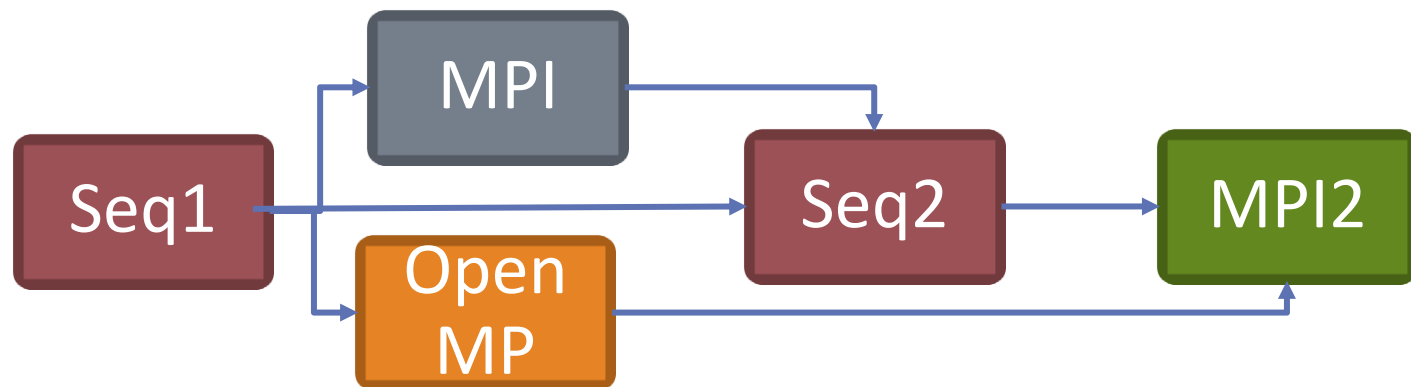
5.1 Egyedi pipeline:



5.2 Adat folyam pipeline:



6. workflow alkalmazás:



1.1 Egyszerű szekvenciális algoritmus végrehajtása párhuzamosítás nélkül

- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő Linux vagy Windows gép gyors létrehozása a felhőben
 - Linux és Windows lemezképeket adunk, amikből a kívánt gépek indíthatóak
 - Négyféle mérettel lehet ezeket elindítani (ld. Következő slide)
 - További kiegészítő tároló (volume) csatlakoztatása a VM-hez
 - Feldolgozandó adatok feltöltése a csatolt tárolóba
 2. Szekvenciális alkalmazás telepítése a létrehozott gépen
 3. A létrehozott gépről elérni a tárolót, ami a feldolgozandó adatot tárolja
 4. Futtatni az alkalmazást a létrehozott gépen
- Tipikus IaaS alkalmazás, ld. demo és gyakorlat
- **Ez az, amit ma megtanítunk használni**

Az Openstack 5-féle VM alaptípust támogat



```
$ nova flavor-list
```

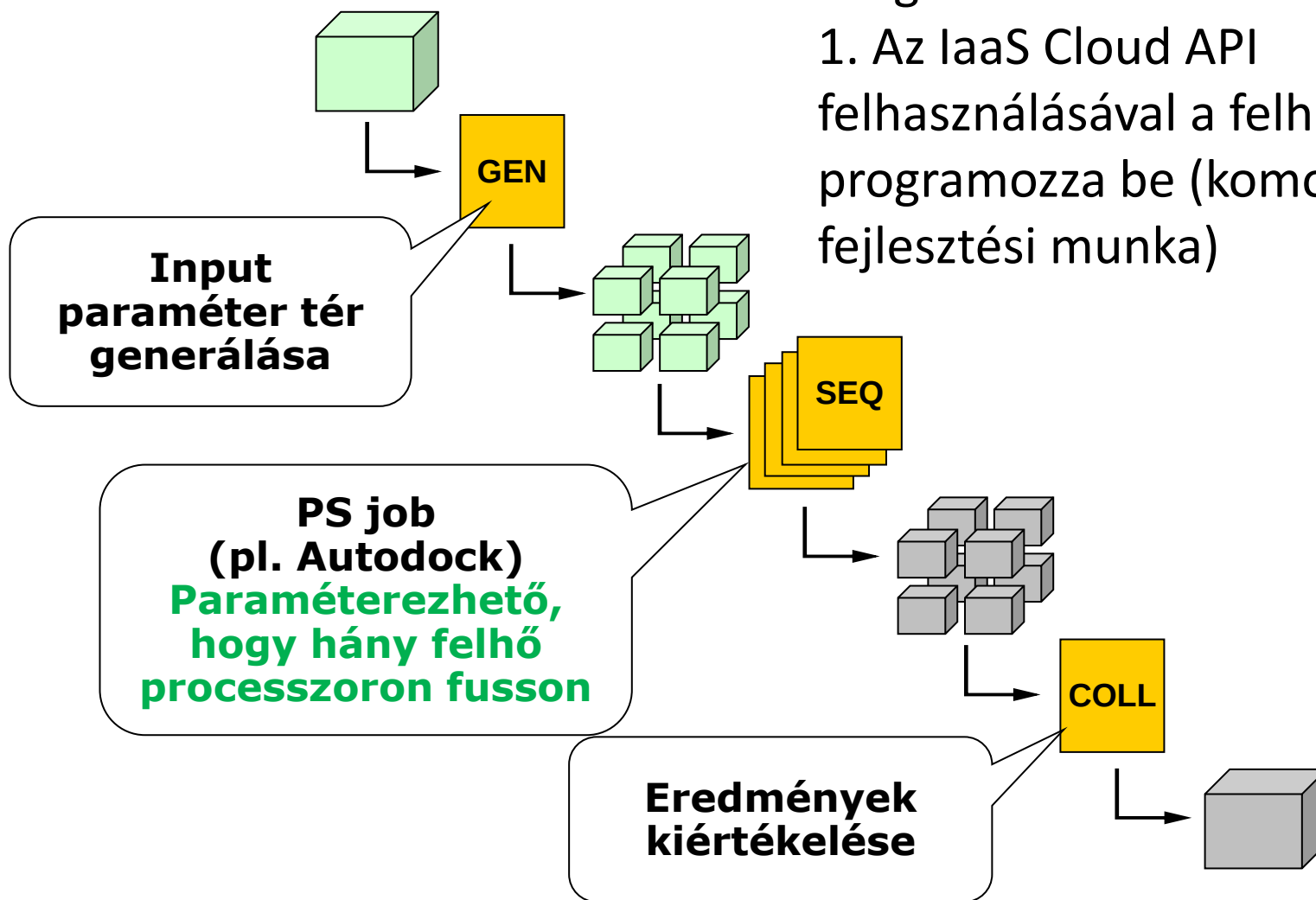
ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs
1	m1.tiny	512	1	0		1
2	m1.small	2048	20	0		1
3	m1.medium	4096	40	0		2
4	m1.large	8192	80	0		4
5	m1.xlarge	16384	160	0		8

- Ezek közül a tiny-t nem támogatjuk, mert az túl kicsi.

1.2 Paraméter söprés (PS) alkalmazások végrehajtása HTC párhuzamosítással

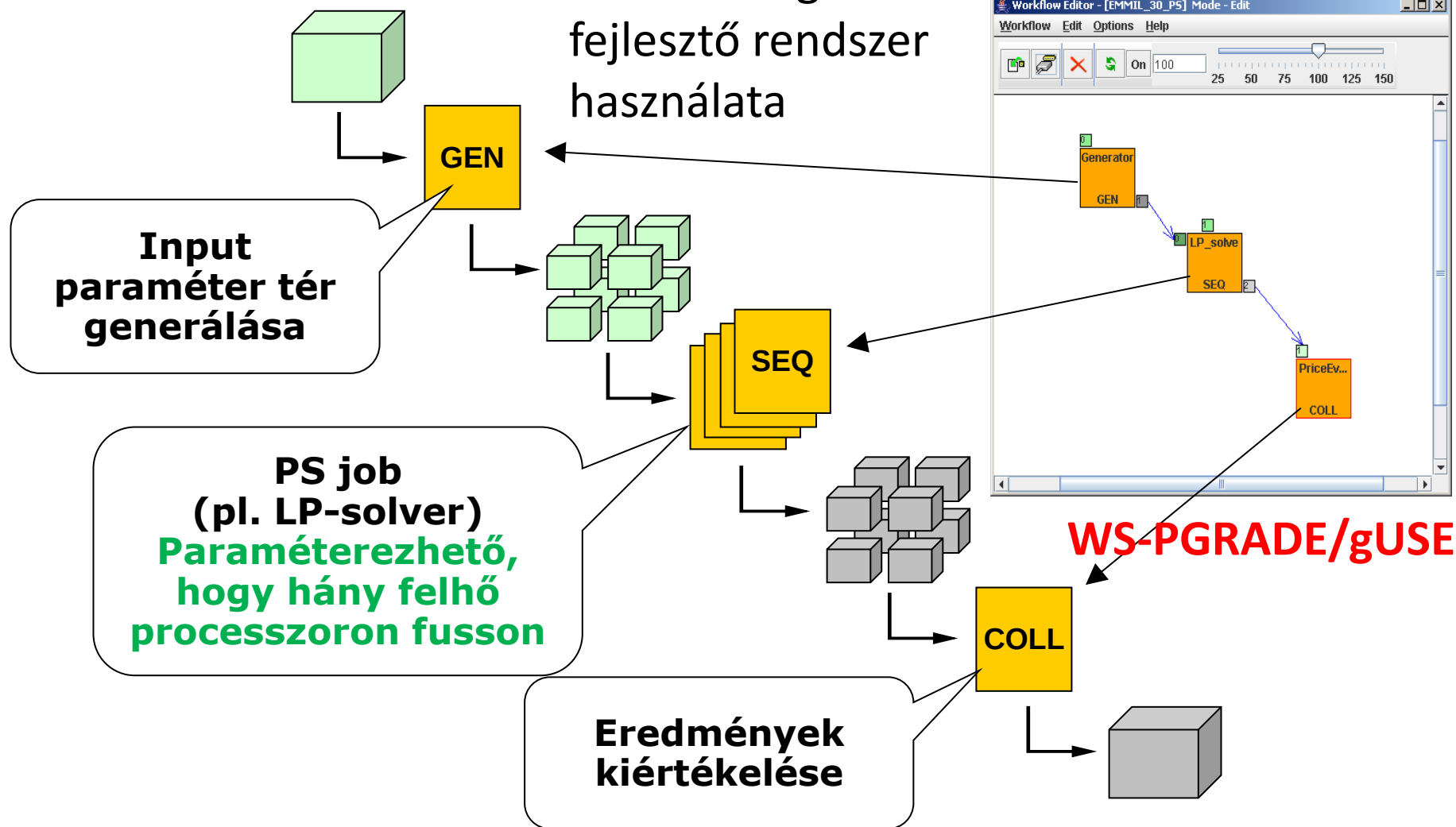
Programozására 2 lehetőség:

1. Az IaaS Cloud API felhasználásával a felhasználó programozza be (komoly fejlesztési munka)



1.2 Paraméter söprés (PS) alkalmazások végrehajtása HTC párhuzamosítással

2. lehetőség: PaaS fejlesztő rendszer használata



2.1 MPI alkalmazás végrehajtása



- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő **MPI futtató rendszert** tartalmazó Linux vagy Windows gép gyors létrehozása a felhőben
 - lemezképet adunk, amiből a kívánt gép indítható (ld. Demo)
 - Négyféle mérettel lehet ezeket elindítani (small és medium ajánlott teszteléshez, large és xlarge ajánlott futtatáshoz)
 2. MPI alkalmazás telepítése a létrehozott gépen vagy beküldése a gépre
 3. A létrehozott gépről elérni a tárolót, ami a feldolgozandó adatot tárolja
 4. Futtatni az alkalmazást a létrehozott gépen
- Ez már egy komplexebb IaaS alkalmazás, aminek a lemezképét később fogjuk közzé tenni az MTA Cloud web lapon

2.2 OpenMP alkalmazás végrehajtása



- Igények és kielégítésük az MTA Felhőben:
 1. Nagyon hasonló az MPI-hoz
- Ez is egy komplexebb IaaS alkalmazás, aminek a támogatását később fogjuk közzé tenni az MTA Cloud web lapon

3.1 Docker alkalmazás végrehajtása párhuzamosítás nélkül

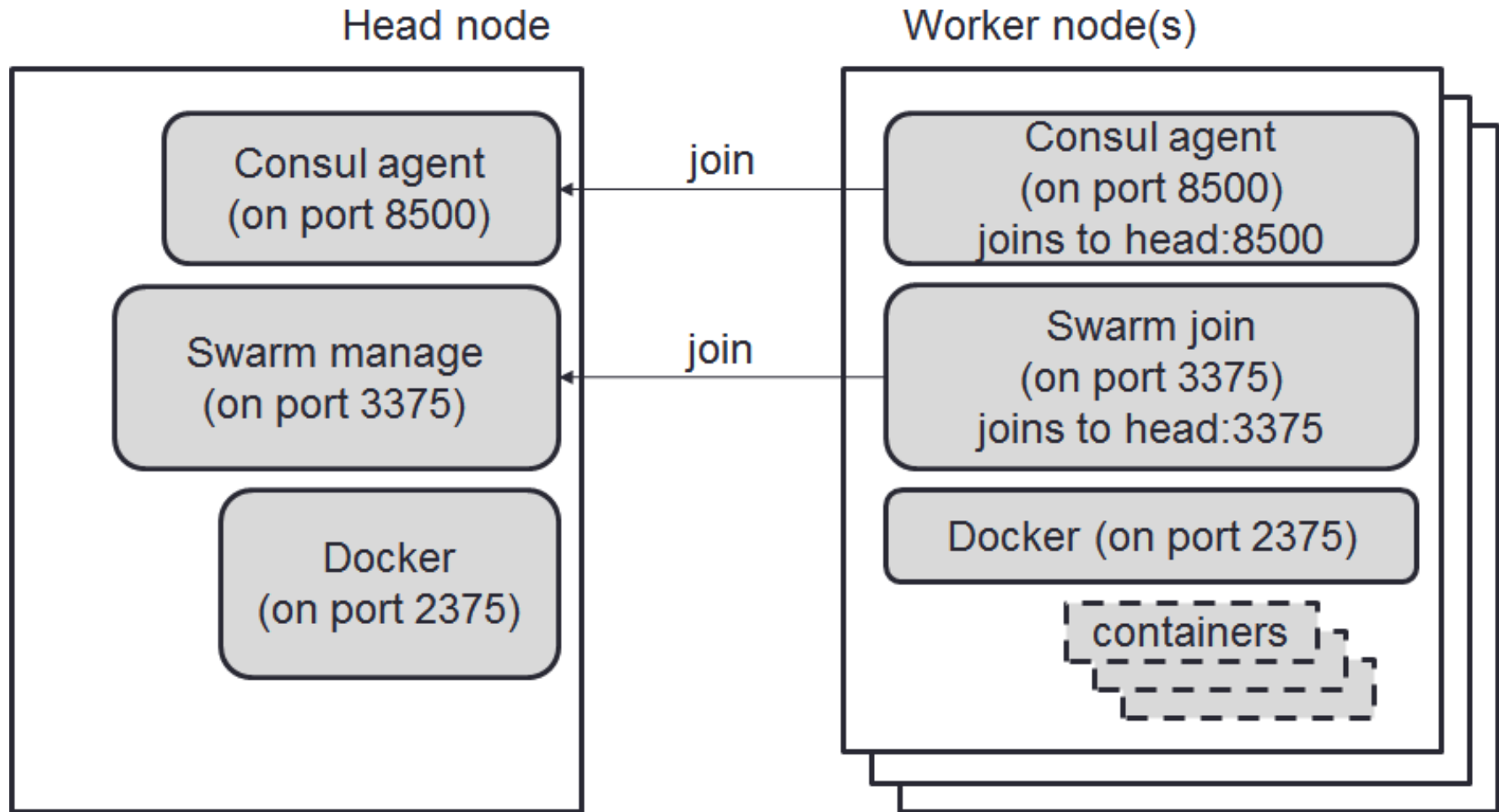


- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő, **docker konténert tartalmazó** Linux vagy Windows gép gyors létrehozása a felhőben
 - Linux és Windows lemezképeket adunk, amikből a kívánt gépek indíthatóak
 - Négyféle mérettel lehet ezeket elindítani
 2. szekvenciális alkalmazás telepítése a docker konténerben
 3. A docker konténerből elérni a tárolót, ami a feldolgozandó adatot tárolja
 4. Futtatni az alkalmazást a létrehozott gépen
- Tipikus IaaS alkalmazás, ld. demo és gyakorlat
- **Ez az, amit ma megtanítunk használni**

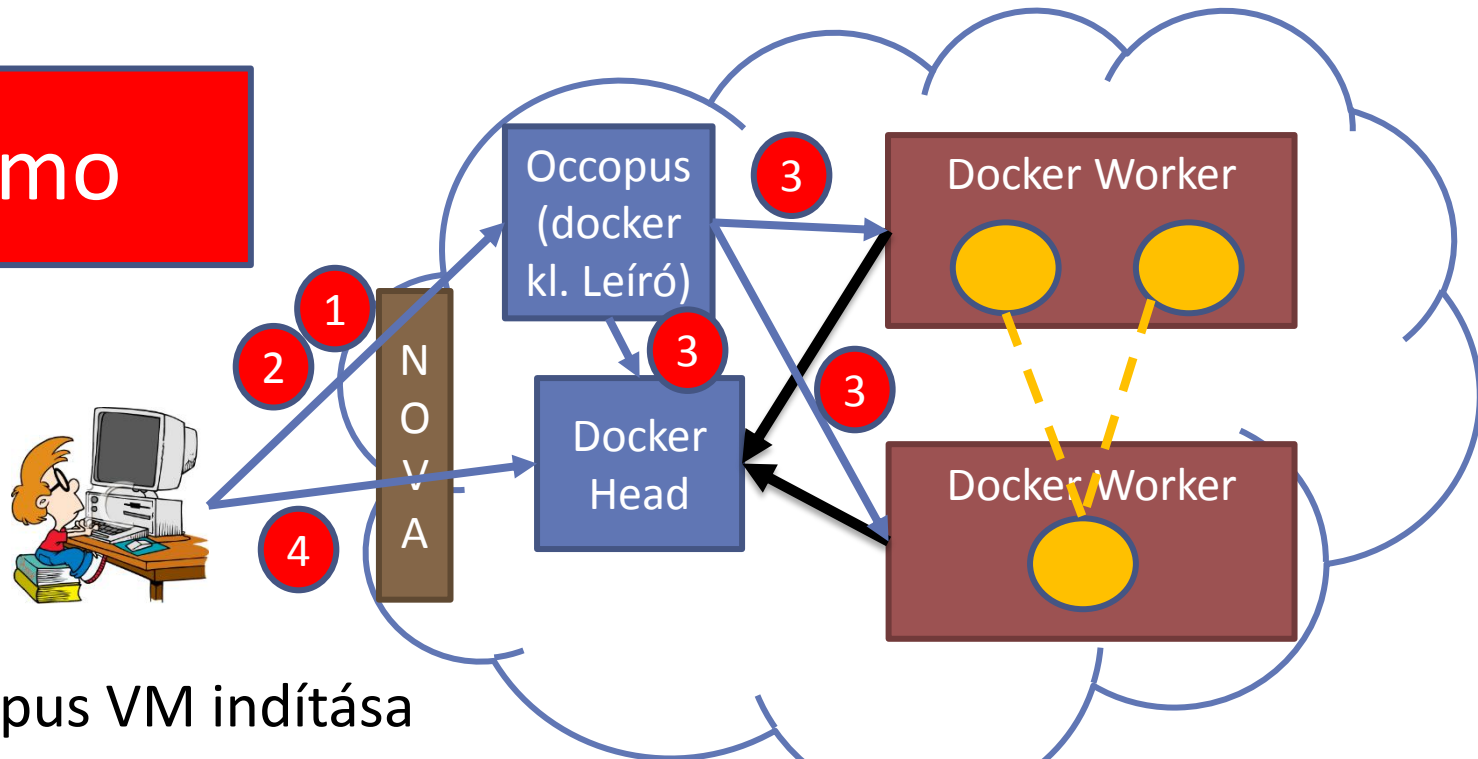
docker klaszteren

- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő, **docker klaszter** létrehozása a felhőben
 - A szükséges lemezképeket, Occopus szervízt és Occopus fájl leírókat adjuk
 2. Docker alkalmazás telepítése a docker konténerben
 3. A docker konténerből elérni a tárolót, ami a feldolgozandó adatot tárolja
 4. Futtatni az alkalmazást a létrehozott docker klaszteren
- Komplex IaaS alkalmazás, **ld. demo**

Docker klaszter felépítése



Ld. demo



1. lépés: Occopus VM indítása

2. lépés: Belépés az Occopus VM-re és a Docker Klaszter Leíróval az Occopus indítása

3. lépés: Occopus automatikusan létrehozza a Docker Klasztert a felhőben

4. lépés: Belépés a Docker Klaszterbe és használata

4. Mapreduce/Hadoop alkalmazás végrehajtása



- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő Hadoop klaszter létrehozása a felhőben
 - Verzió 1: WS-PGRADE-ben workflow-ként definiálva (EGI FedCloud-on megy, de MTA Cloud-ra még adaptálni kell)
 - Verzió 2: Occopus segítségével (fejlesztés alatt)
 2. mapreduce alkalmazás végrehajtása a létrehozott Hadoop klaszteren
 - Verzió 1: WS-PGRADE-ben workflow-ként definiálva
 - Verzió 2: Occopus segítségével
- Tipikus PaaS alkalmazás

WS-PGRADE/gUSE segítségével

- **deploy**

- Port 0: Cluster configuration file
- Port 1: Software bundle (e.g. python interface, scripts)
- Port 2: Cluster configuration bundle (Channel)



- **exe**

- Port 0: Cluster configuration bundle (Channel)
- Port 1: Job configuration file
- Port 2: Data bundle (e.g. local input, job jar)
- Port 3: Cluster configuration bundle (Channel)
- Port 4: Job output, if the output selected is local



- **destroy**

- Port 0: Cluster configuration bundle (Channel)



5.1 Egyedi pipeline alkalmazás végrehajtása



- Két opció:
 - IaaS API segítségével a felhasználó programozza be
 - PaaS eszköz segítségével (pl. WS-PGRADE/gUSE)
- 1. Opció: IaaS API használata
- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő Linux vagy Windows gépek gyors létrehozása a felhőben
 2. Ezeket a pipeline sorrendnek megfelelően egymás felé láthatóvá és elérhetővé kell tenni
 3. szekvenciális alkalmazás telepítése a létrehozott gépeken
 4. A létrehozott gépekről elérni a tároló(ka)t, ami(k) a feldolgozandó adatot tárolja(ják)
 5. Futtatni az alkalmazást a létrehozott gépeken
- Tipikus komplex IaaS alkalmazás -> érdemes Occopus-t használni

5.1 Egyedi pipeline alkalmazás végrehajtása



- 2. Opció
 - PaaS eszköz segítségével (WS-PGRADE/gUSE)
- Igények és kielégítésük az MTA Felhőben WS-PGRADE/gUSE esetén:
 1. Az alkalmazás pipeline definiálása, mint workflow
 2. Az alkalmazást a bemenő adatokkal beküldeni a felhőbe
- Tipikus PaaS alkalmazás -> érdemes WS-PGRADE/gUSE-t használni
- Például a korábban látott mapreduce/hadoop alkalmazás ilyen pipeline workflow-ként is használható a WS-PGRADE/gUSE-ban

WS-PGRADE/gUSE segítségével

- **deploy**

- Port 0: Cluster configuration file
- Port 1: Software bundle (e.g. python interface, scripts)
- Port 2: Cluster configuration bundle (Channel)

- **exe**

- Port 0: Cluster configuration bundle (Channel)
- Port 1: Job configuration file
- Port 2: Data bundle (e.g. local input, job jar)
- Port 3: Cluster configuration bundle (Channel)
- Port 4: Job output, if the output selected is local

- **destroy**

- Port 0: Cluster configuration bundle (Channel)



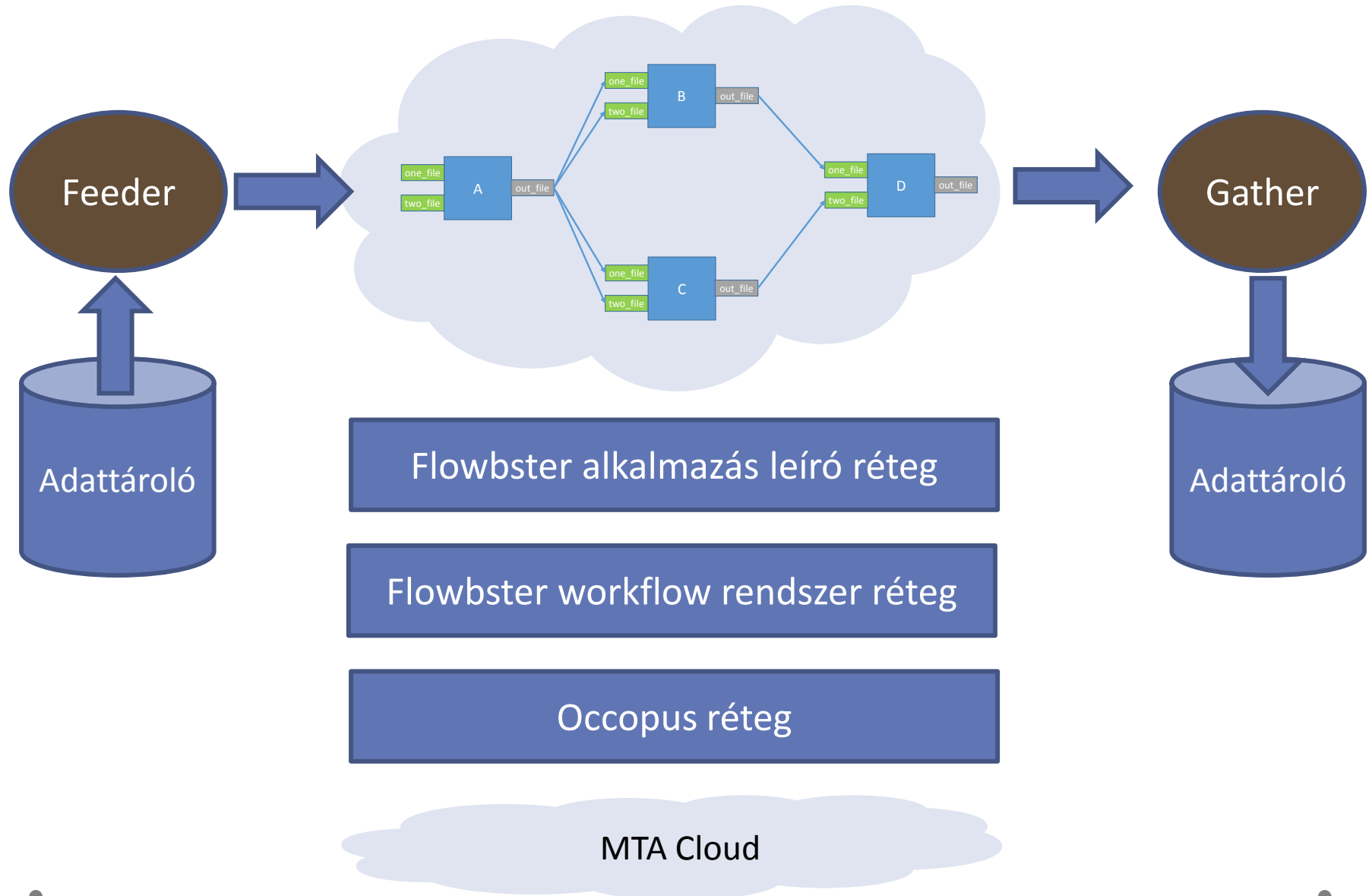
- Két opció:
 - IaaS API segítségével a felhasználó programozza be
 - PaaS eszköz segítségével (Flowbster, ami Occopus-ra épül)
- 1. Opció: IaaS API használata
- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő Linux vagy Windows gépek gyors létrehozása a felhőben
 2. Ezeket a pipeline sorrendnek megfelelően egymás felé láthatóvá és elérhetővé kell tenni
 3. szekvenciális alkalmazás telepítése a létrehozott gépeken
 4. Átfolytatni az adatokat a létrehozott pipeline-ba kapcsolt gépeken
- Tipikus komplex IaaS alkalmazás -> érdemes Occopus-t használni

5.2 Adat folyam pipeline alkalmazás végrehajtása



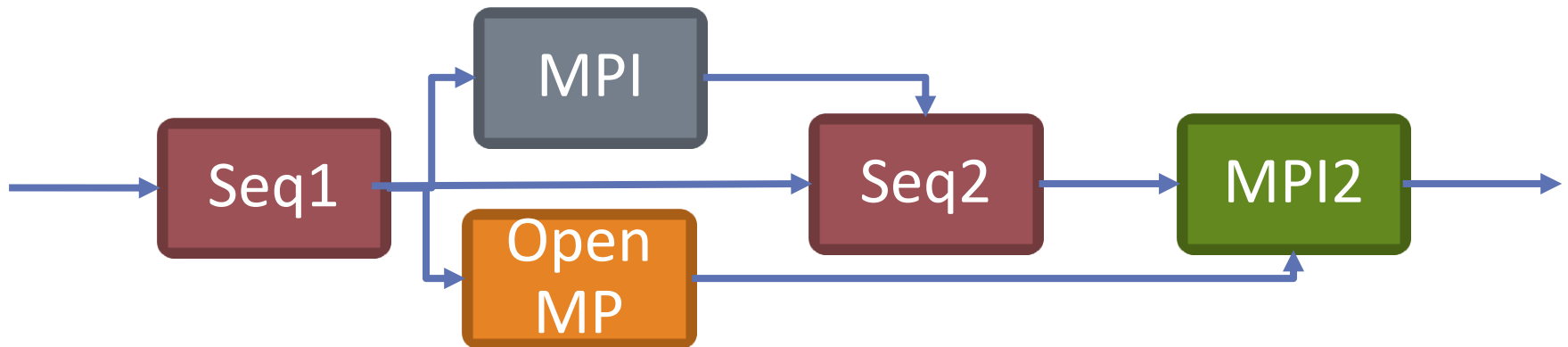
- 2. opció:
 - PaaS eszköz segítségével (**Flowbster**, ami Occopus-ra épül)

Flowbster koncepció



- 2. opció:
 - PaaS eszköz segítségével (**Flowbster**, ami Occopus-ra épül)
- Igények és kielégítésük az MTA Felhőben **Flowbster** esetén:
 1. Flowbster (pipeline) node-ok és kapcsolatuk definiálása a Flowbster alkalmazás leíró réteg segítségével. Két opció:
 1. Szövegesen Occopusnak megfelelő fájl leíró nyelven (Occopus tudást igényel)
 2. Grafikusan (semmilyen előismeret nem kell hozzá) – fejlesztés alatt
 2. szekvenciális alkalmazások definiálása a pipeline node-ok számára (Flowbster node-ok konfigurálása)
 3. Szükséges virtuális gépek indítása a felhőben és összekapcsolásuk a Flowbster definíció szerint – Occopus végzi el
 4. Átfolytatni az adatokat a létrehozott pipeline-ba kapcsolt gépeken

6. Workflow alkalmazás végrehajtása



6. Workflow alkalmazás végrehajtása



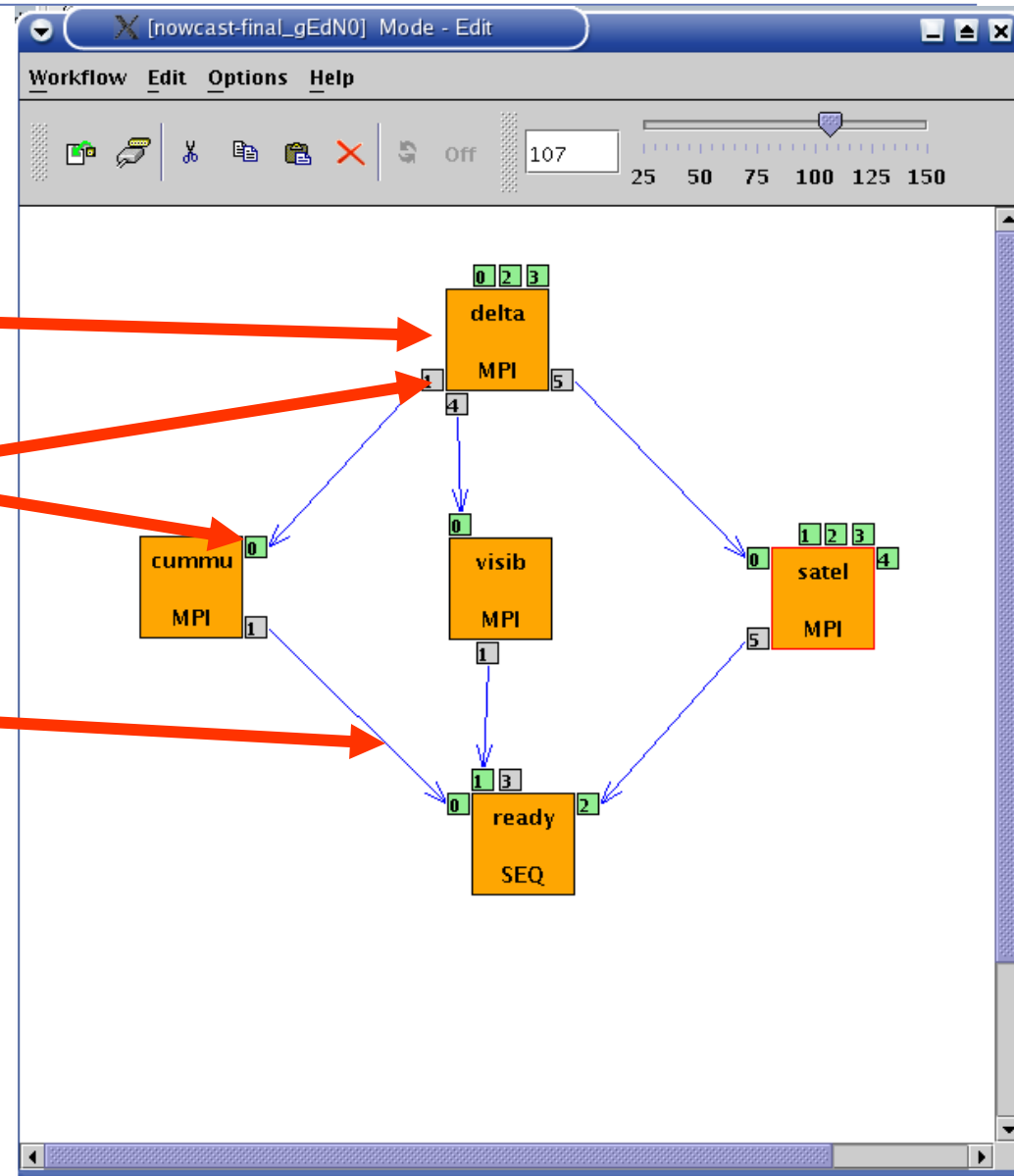
- Két opció:
 - IaaS API segítségével a felhasználó programozza be
 - PaaS eszköz segítségével (WS-PGRADE/gUSE)
- 1. Opció: IaaS API használata
- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő Linux vagy Windows gépek gyors létrehozása a felhőben
 2. Ezeket a workflow sorrendnek megfelelően egymás felé láthatóvá és elérhetővé kell tenni
 3. szekvenciális alkalmazás telepítése a létrehozott gépeken
 4. A létrehozott gépekről elérni a tároló(ka)t, ami(k) a feldolgozandó adatot tárolja(ják)
 5. Futtatni az alkalmazást a létrehozott gépeken
- Tipikus komplex IaaS alkalmazás -> érdemes Occopus-t használni

6. Workflow alkalmazás végrehajtása



- 2. opció:
 - PaaS eszköz segítségével (**WS-PGRADE/gUSE**)
- Igények és kielégítésük az MTA Felhőben **WS-PGRADE/gUSE** esetén:
 1. Az alkalmazás definiálása, mint workflow
 2. Az alkalmazást a bemenő adatokkal beküldeni a felhőbe
- Tipikus PaaS alkalmazás -> érdemes WS-PGRADE/gUSE-t használni

- **Irányított aciklikus gráf,** ahol
 - A node-ok **job-okat, szervízeket, vagy beágyazott workflow-t** reprezentálnak
 - A port-ok input/output fájlakat reprezentálnak
 - Az élek a node-ok közötti adatmozgatást és függőséget reprezentálják
- **Workflow szemantika:**
 - Egy node tüzelhet, azaz a hozzá tartozó job végrehajtható, ha minden input fájl megérkezett



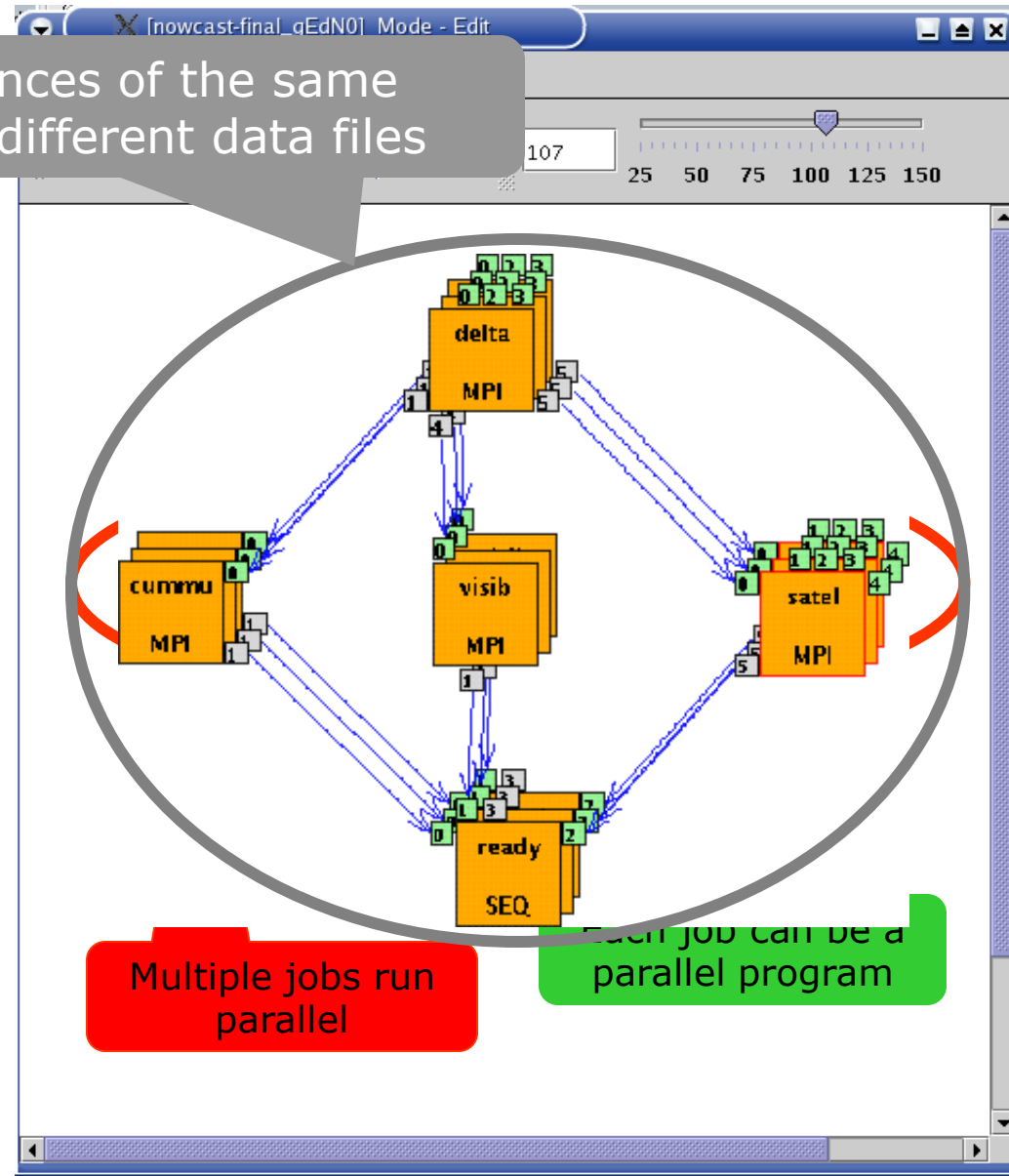
Három szintű párhuzamosság

Multiple instances of the same workflow with different data files

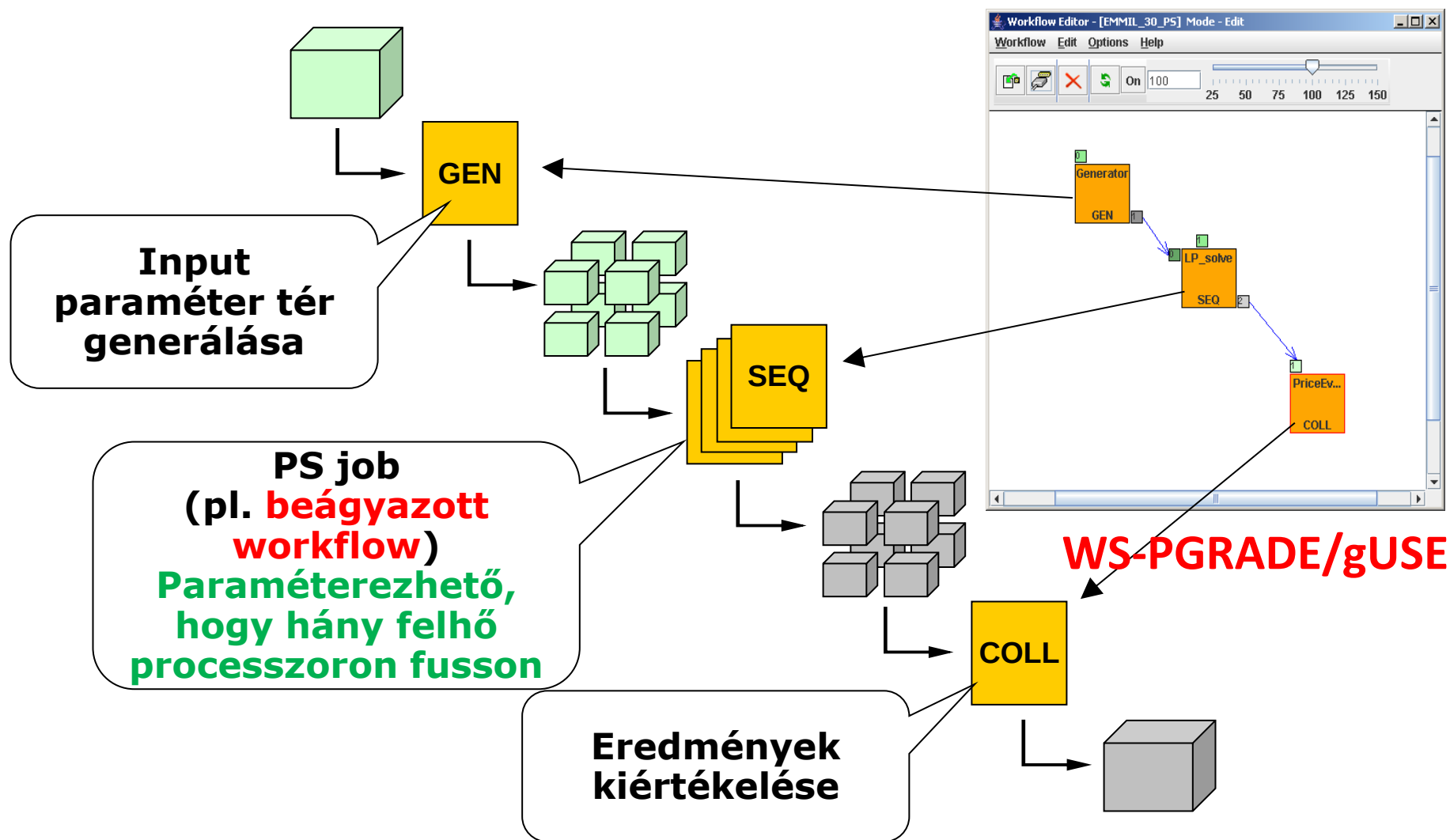
– Parallel execution inside a workflow node

– Parallel execution among workflow nodes

– Parameter study execution of the workflow

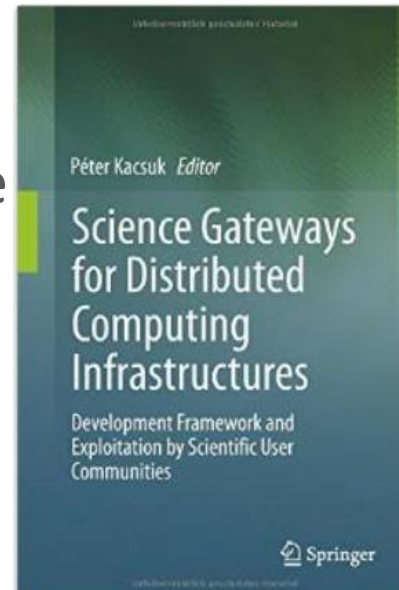


Paraméter söprés (PS) párhuzamosítás



WS-PGRADE/gUSE használata

- A sok-felhő rendszerre az alkalmazások fejlesztését és futtatását támogatja
- Ennek segítségével különböző tudományterületekre specializált SaaS szolgáltatásokat lehet kifejleszteni (pl. Autodock gateway kémiai és gyógyszerkutatási alkalmazásokhoz)



- Felhasználása európai projektekben:

- **Több mint 30 projekt** használja
- **EGI FedCloud**
- **DRIHM** – meteorológiai és vízgazdálkodási alkalmazások
- **VERGE** – földtudományok
- **CloudSME** – vállalati és gyártástechnológiai szimulációk



- A **standard OpenStack IaaS támogatás** készen áll, július 1-től béta verzióban elérhető a szolgáltatás
- Bonyolultabb IaaS VM szerkezetek (pl. docker klaszter, pipeline, stb.) létrehozásához érdemes az **Occopus** szolgáltatást használni, ami július 1-től béta verzióban elérhető szolgáltatás
- Magasabb szintű alkalmazásokhoz érdemes **PaaS eszközöket** alkalmazni:
 - **WS-PGRADE/gUSE**: sokféle felhővel megy már, de az MTA Felhőhöz még adaptálni kell (várhatóan az októberi átadásra menni fog)
 - **Flowbster**: parancssoros verzió rendelkezésre áll, a grafikus verzió várhatóan az októberi átadásra menni fog