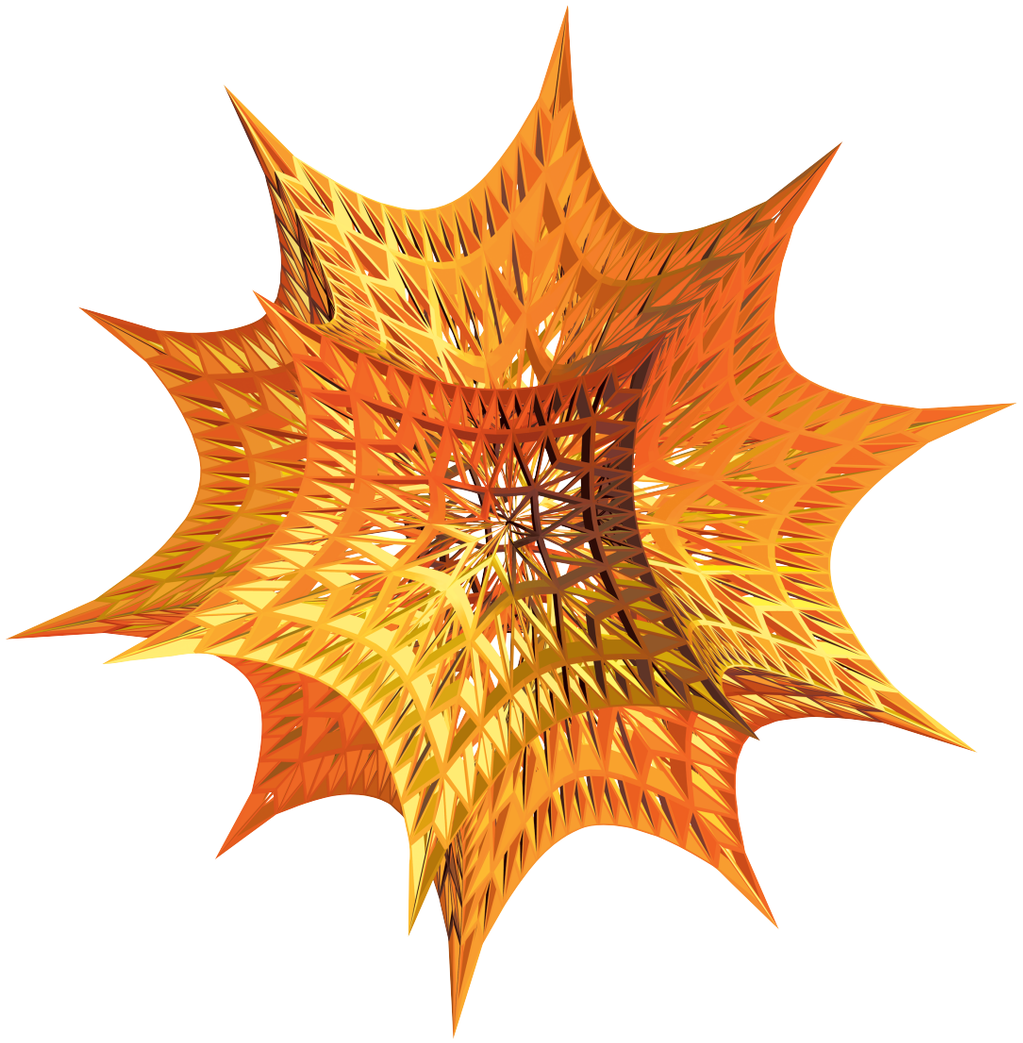


WOLFRAMRESEARCH
MAKERS OF MATHEMATICA®

MATHEMATICA® 5.2

&

MATHEMATICA® PARALLEL COMPUTING TOOLKIT

Párhuzamos programozás Mathematica-val

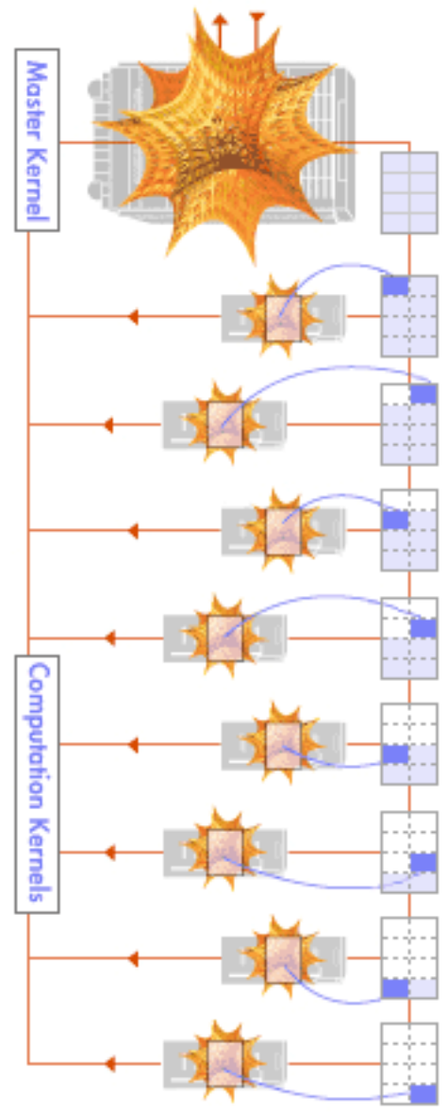
Tóth Gyula

PhD hallgató

MTA SZFKI – BME TTK Elméleti Fizika Tanszék

Bevezetés

- Párhuzamos programozás
  - Funkcionális párhuzamosítás
  - Adatpárhuzamosítás
- Optimalizálás
  - szinkronizálás – leggyengébb láncszem
  - számításigény / kommunikációigény
- Protokollok
  - alacsony szinten: LAM/MPI (c protokollok)
  - fejlett szoftverek csomagjai: Mathematica PCT

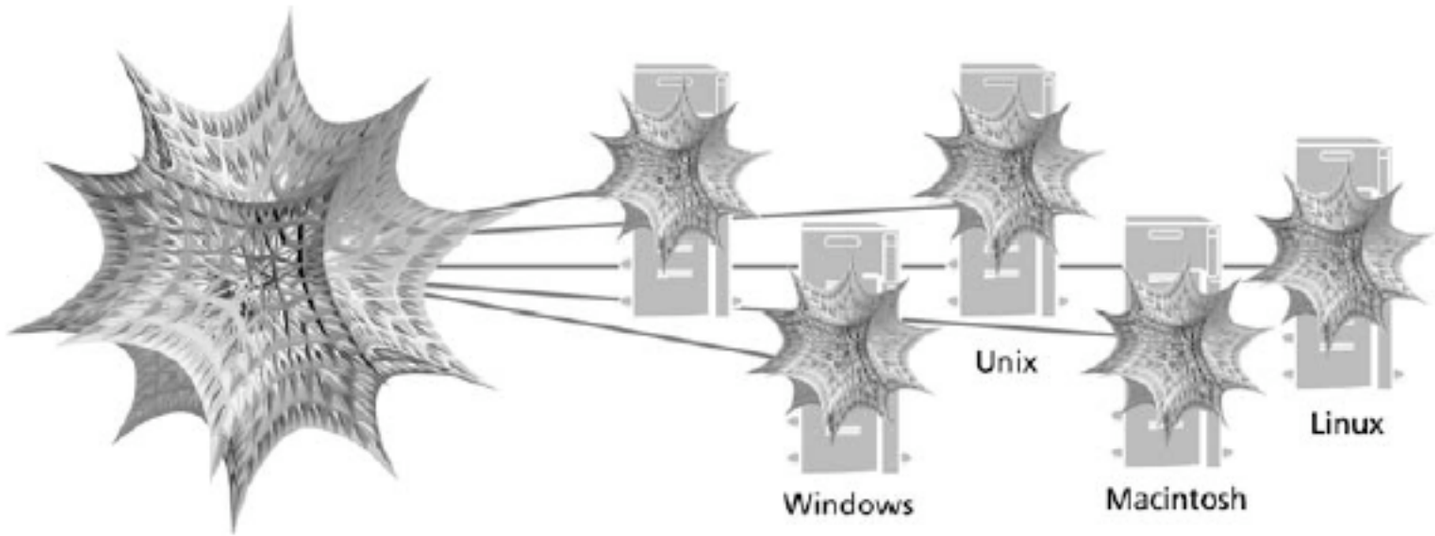


A PCT működése

- FrontEnd – Master Kernel & Slave Kernels

protocols:

ssh / rsh



- Indítási módok
  - **Aktív mód**: a szerveren futó FrontEnd-ből a master kernelen keresztül
  - **Passzív mód:** manuális indítás a slave-eken, majd csatlakozás hozzájuk a szerverről

Kommunikáció: szabad port v. Standard I/O

Tesztkörnyezet

40x dual core + 80x Athlon64

24x dual processor Xeon Quad

A PCT indítása

WOLFRAMRESEARCH
MAKERS OF MATHEMATICA®

PCT betöltése

```
Needs["Parallel`Debug`"]
```

```
Needs["Parallel`"]
```

```
Parallel Computing Toolkit 2.0 (November 11, 2004)
```

```
Created by Roman E. Maeder
```

slave kernelek indítása

```
machines = Import["machinesA.txt", "Lines"];
```

```
LaunchSlave[#, "ssh `1` /usr/local/bin/math -mathlink",
   ConnectionType → LinkLaunch] & /@ machines
```

```
{slave1[node21], slave2[node22], slave3[node23], slave4[node24], slave5[node25],
 slave6[node26], slave7[node27], slave8[node28], slave9[node29], slave10[node30]}
```

slave kernelek ellenőrzése

```
TableForm[
 RemoteEvaluate[{$ProcessorID, $MachineName, $SystemID, $ProcessID, $Version}],
 TableHeadings → {None, {"ID", "host", "OS", "process", "Mathematica Version"}}]
```

ID	host	OS	process	Mathematica Version
1	node21	Linux-x86-64	26179	5.2 for Linux x86 (64 bit) (June 20
2	node22	Linux-x86-64	24525	5.2 for Linux x86 (64 bit) (June 20
3	node23	Linux-x86-64	24608	5.2 for Linux x86 (64 bit) (June 20
4	node24	Linux-x86-64	24064	5.2 for Linux x86 (64 bit) (June 20
5	node25	Linux-x86-64	24923	5.2 for Linux x86 (64 bit) (June 20
6	node26	Linux-x86-64	22139	5.2 for Linux x86 (64 bit) (June 20
7	node27	Linux-x86-64	22095	5.2 for Linux x86 (64 bit) (June 20
8	node28	Linux-x86-64	22300	5.2 for Linux x86 (64 bit) (June 20
9	node29	Linux-x86-64	22081	5.2 for Linux x86 (64 bit) (June 20
10	node30	Linux-x86-64	22056	5.2 for Linux x86 (64 bit) (June 20

kernelek bezárása

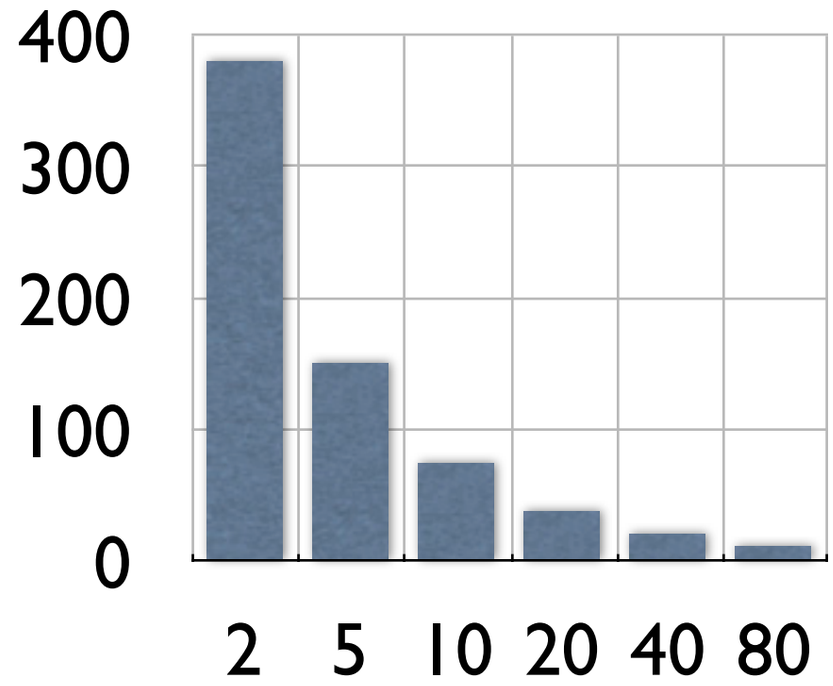
```
CloseSlaves[];
```

Sebességi teszt

WOLFRAMRESEARCH
MAKERS OF MATHEMATICA

Ponthalmaz átmérője

$$d = \max\left\{\left\|r_i - r_j\right\|; i, j = 1 \ldots n\right\}$$




```
pointset = Table[{Random[], Random[], Random[]}, {10000}];

d[x_List, y_List] := √(Plus @@ (y - x)^2)

diameterparallel2[pts_List] :=
 Module[{pointset = pts, djpar},
  djpar[j_] := With[{pj = pointset[[j]]}, Max[d[#, pj] & /@ Drop[pointset, j]]];
  ExportEnvironment[pointset, djpar, d];
  With[{n = Length[pointset]},
   res = Max[ParallelTable[Max[{djpar[j], djpar[n - j]}], {j, 1, Floor[n / 2]}]]
  ];
  RemoteEvaluate[Clear[pointset, djpar]];
  res
 ]

AbsoluteTiming[diameterparallel2[pointset]]

{21.071954 Second, 1.66908}
```

Soros futási idő:

740s

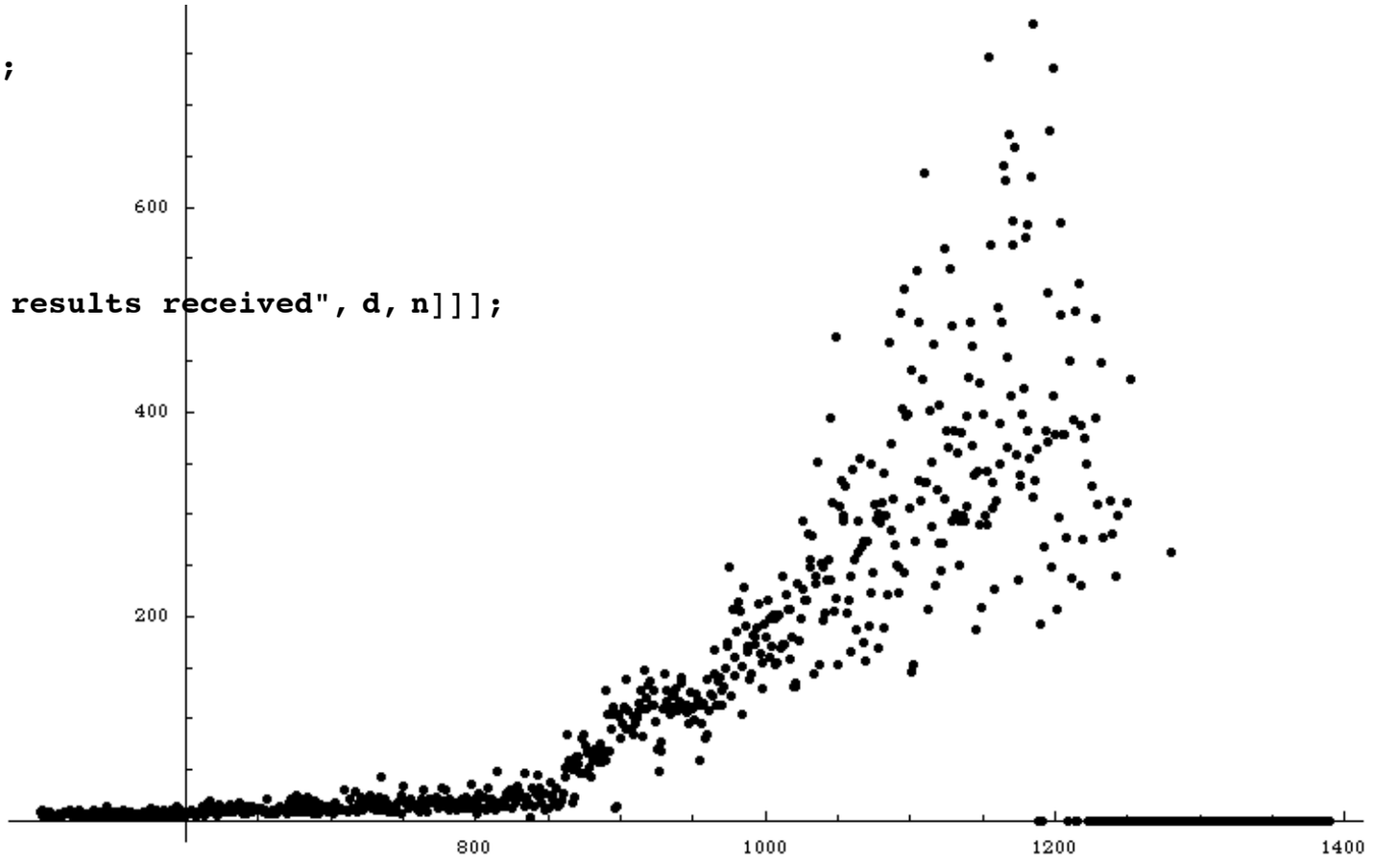
PCT programozás

WOLFRAMRESEARCH
MAKERS OF MATHEMATICA®

Polinomok irreducibilitása: $p_{n,i} = \sum_{i=1}^{n-1} i x^{i-1}$

ParallelTable
vs.
szinkronizálás

```
test[d_] :=
  Module[{res, x},
   res = Timing[Length[FactorList[poly[d, x]]] - 1];
   {res[[1]] /. Second → 1, d, res[[2]]}
  ];

d0 = d = 500; ids = {}; n = 0; results[_] := {0, 0};
CheckAbort[
  While[True,
   While[$QueueLength ≤ 1,
    AppendTo[ids, Queue[{d}, test[d]]];
    If[Mod[d, Length[$Slaves]] == 0,
     Print[StringForm["Degree `1` queued, `2` results received", d, n]]];
    d++;
    If[! QueueRun[], Break[]];
   ];
   If[Length[ids] > 0,
    {res, id, ids} = WaitOne[ids];
    n++;
    {time, deg, nf} = res;
    results[deg] = {time, nf};
    If[nf > 1, Print[res];];
   ];
  ],
  Print["aborting..."];
  ResetSlaves[]
 ];
```



PCT programozás

WOLFRAMRESEARCH
MAKERS OF MATHEMATICA®

Kaszkád programozása

f@@expr rekurzív módban

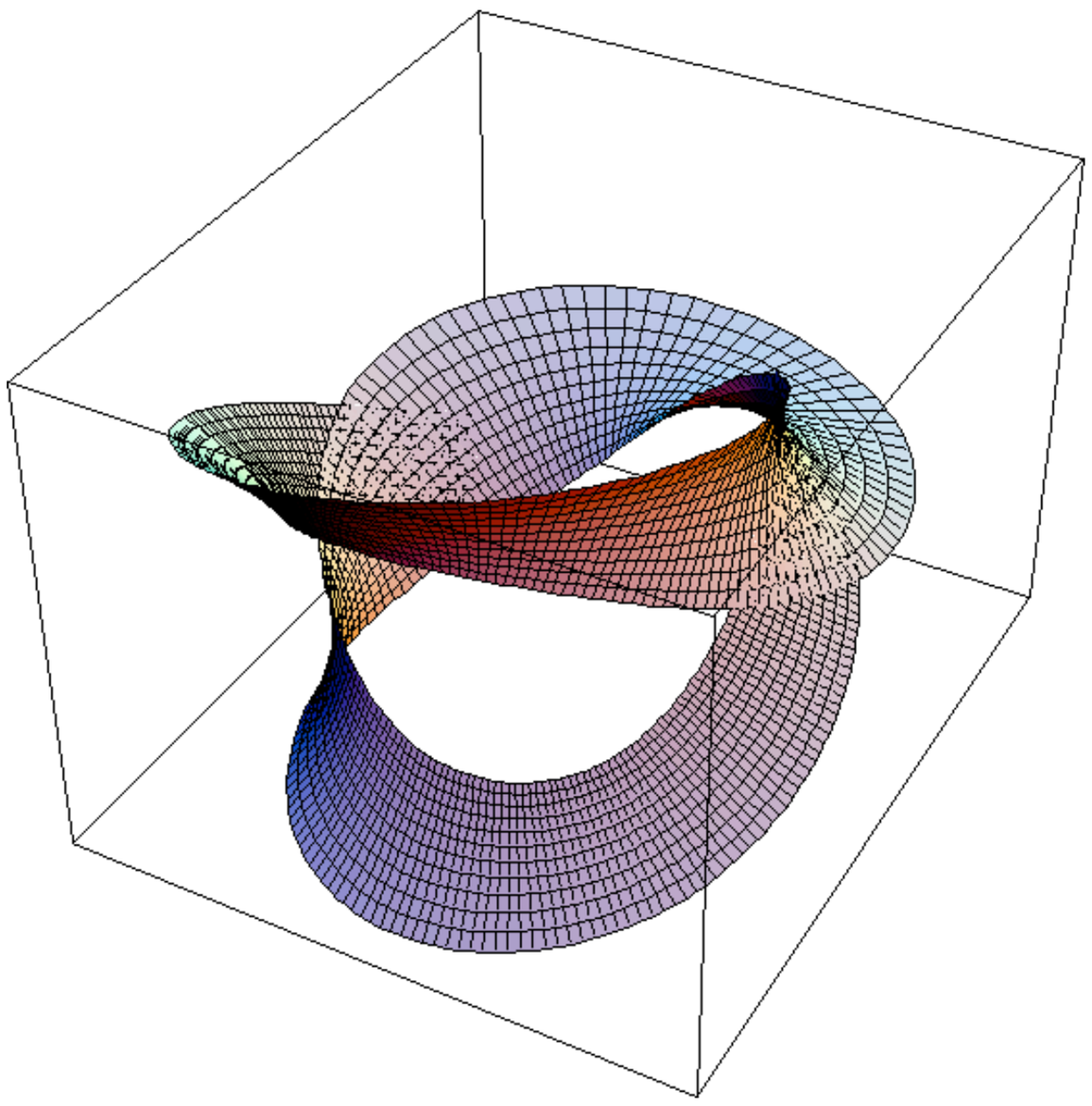
```
Cascade[f_, expr0_] /; Length[expr0] > 2 * Length[$Slaves] :=
 Cascade[f,
  ParallelEvaluate[expr0, f @@ # &, Head[expr0]]]
Cascade[f_, expr0_] /; Length[expr0] > 0 :=
 Module[{expr = expr0},
  While[Length[expr] > 1,
   expr = Wait[Composition[Queue, f] @@@ Partition[expr, 2, 2, 1, {}]];];
  First[expr]]

SetOptions[$DebugObject, Trace → {SendReceive}];

Cascade[Plus, {1, 2, 3, 4, 5}]

SendReceive: Sending to slave1[node21]: pid1[1 + 2] (q=1)

SendReceive: Sending to slave2[node22]: pid2[3 + 4] (q=1)

SendReceive: Sending to slave3[node23]: pid3[+5] (q=1)

SendReceive: Receiving from slave1[node21]: pid1[3] (q=0)

SendReceive: Receiving from slave2[node22]: pid2[7] (q=0)

SendReceive: Receiving from slave3[node23]: pid3[5] (q=0)

SendReceive: Sending to slave1[node21]: pid4[3 + 7] (q=1)

SendReceive: Sending to slave2[node22]: pid5[+5] (q=1)
```

```
Cascade[Plus, Range[4 * Length[$Slaves] + 1]]

SendReceive: Sending to slave1[node21]: (Plus @@ #1 &)[{1, 2, 3, 4, 5}] (q=1)

SendReceive: Sending to slave2[node22]: (Plus @@ #1 &)[{6, 7, 8, 9}] (q=1)

SendReceive: Sending to slave3[node23]: (Plus @@ #1 &)[{10, 11, 12, 13}] (q=1)

SendReceive: Receiving from slave1[node21]: 15 (q=0)

SendReceive: Receiving from slave2[node22]: 30 (q=0)

SendReceive: Receiving from slave3[node23]: 46 (q=0)

SendReceive: Sending to slave1[node21]: pid1[15 + 30] (q=1)

SendReceive: Sending to slave2[node22]: pid2[+46] (q=1)

SendReceive: Receiving from slave1[node21]: pid1[45] (q=0)

SendReceive: Receiving from slave2[node22]: pid2[46] (q=0)
```

PCT programozás

WOLFRAMRESEARCH
MAKERS OF MATHEMATICA®

Grafika és animáció



ParallelAnimate
ParallelPlot3D
ParallelDensityPlot
ParallelParametricPlot3D
ParallelContourPlot

nagyfelbontású
parametrizált felület:
– soros: 0.53s
– 20PC: 0.012s
a kirajzolás ideje: ~1perc!

korlát: videókártya!

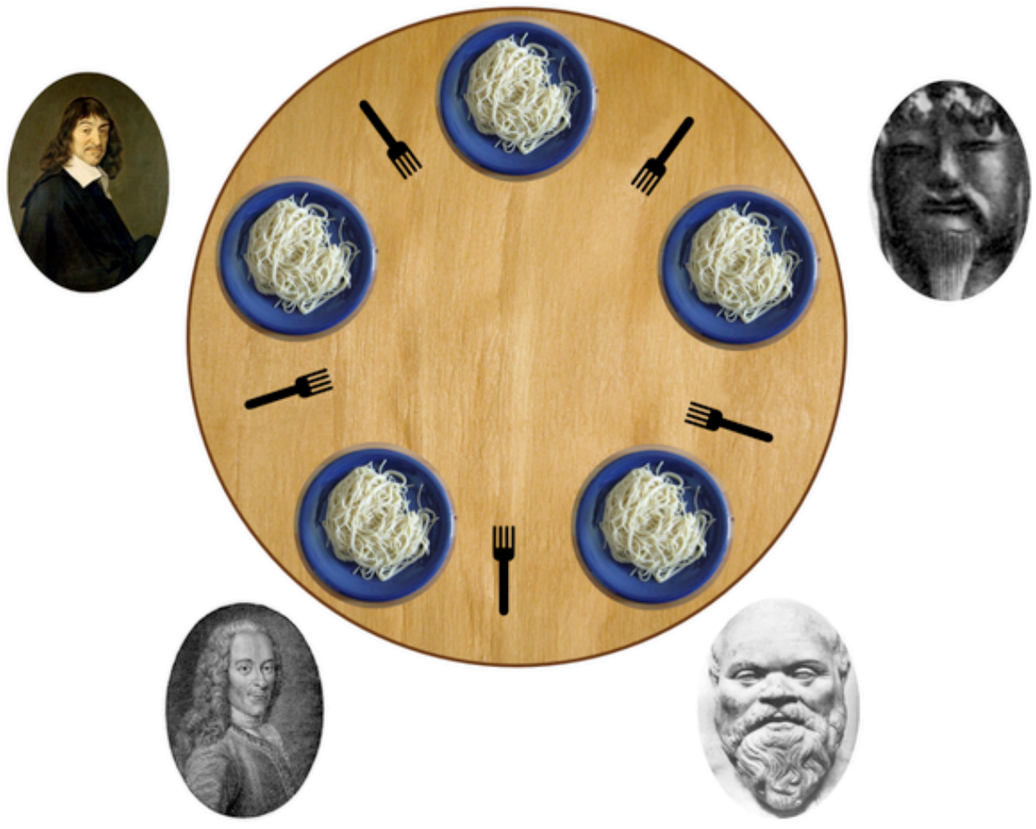
PCT programozás

WOLFRAMRESEARCH
MAKERS OF MATHEMATICA®

Megosztott erőforrások, szinkronizálás

```
SharedDownValues[{forks}]
philo[i_, n_] :=
 With[{f1 = i, f2 = Mod[i, n] + 1},
  While[True,
   Print[i, " thinking"]; Pause[10 Random[]];
   Print[i, " hungry"];
   While[TestAndSet[forks[f1], i] =!= i ||
     TestAndSet[forks[f2], i] =!= i,
    Print[i, " *angry*"]; Pause[Random[]]];
   Print[i, " eating"]; Pause[2 Random[]];
   forks[f1] = Null; forks[f2] = Null;]]
ExportEnvironment[philo];
Clear[philo]

philos =
  With[{n = Length[$Slaves]},
   Table[philo[i, n], {i, n}]];

ParallelDispatch[Evaluate[philos]]
```

ExportEnvironment vs. SharedDownVariable



PCT programozás

WOLFRAMRESEARCH
MAKERS OF MATHEMATICA®

Párhuzamos próbálkozás

ParallelTry

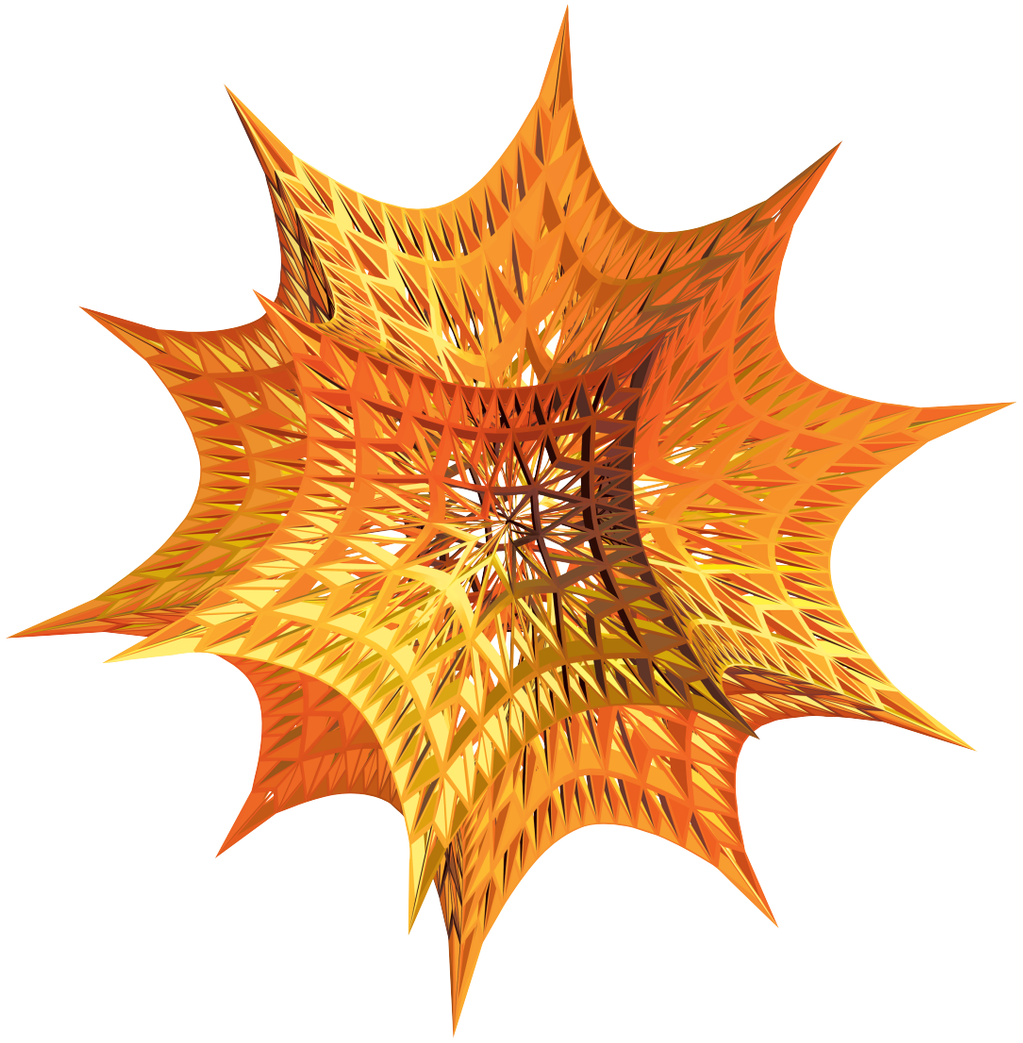
```
SetAttributes[parallelTry, HoldRest]
parallelTry[tf_, exprs__] :=
 Module[{res = tf, id, ids},
  ids = Queue /@ Unevaluated[{exprs}];
  CheckAbort[
   While[res == tf && Length[ids] > 0, (* try one more *)
    {res, id, ids} = WaitOne[ids]
   ],
   res = $Aborted;
  ];
  res
 ]

SetAttributes[ParallelTry, HoldAll]
ParallelTry[exprs__] := parallelTry[$Failed, exprs]

res = ParallelTry @@ Join @@ exprs;
```

- prímszámkeresés
- véletlen polinomok faktorizálhatósága
- stb.

Összefoglalás

- Párhuzamos programok alapelvei
- A PCT működése
  - Felépítés, indítási módok
  - Egyszerű teszt a sebességre
- Programozás PCT-vel, párhuzamos algoritmusok
  - számítási idő kihasználása
  - kaszkád kiértékelés
  - grafika
  - megosztott erőforrások
  - ParallelTry

WOLFRAMRESEARCH
MAKERS OF MATHEMATICA®



MATHEMATICA® 5.2

&

MATHEMATICA® PARALLEL COMPUTING TOOLKIT

Köszönöm a figyelmet!