int.eu.grid

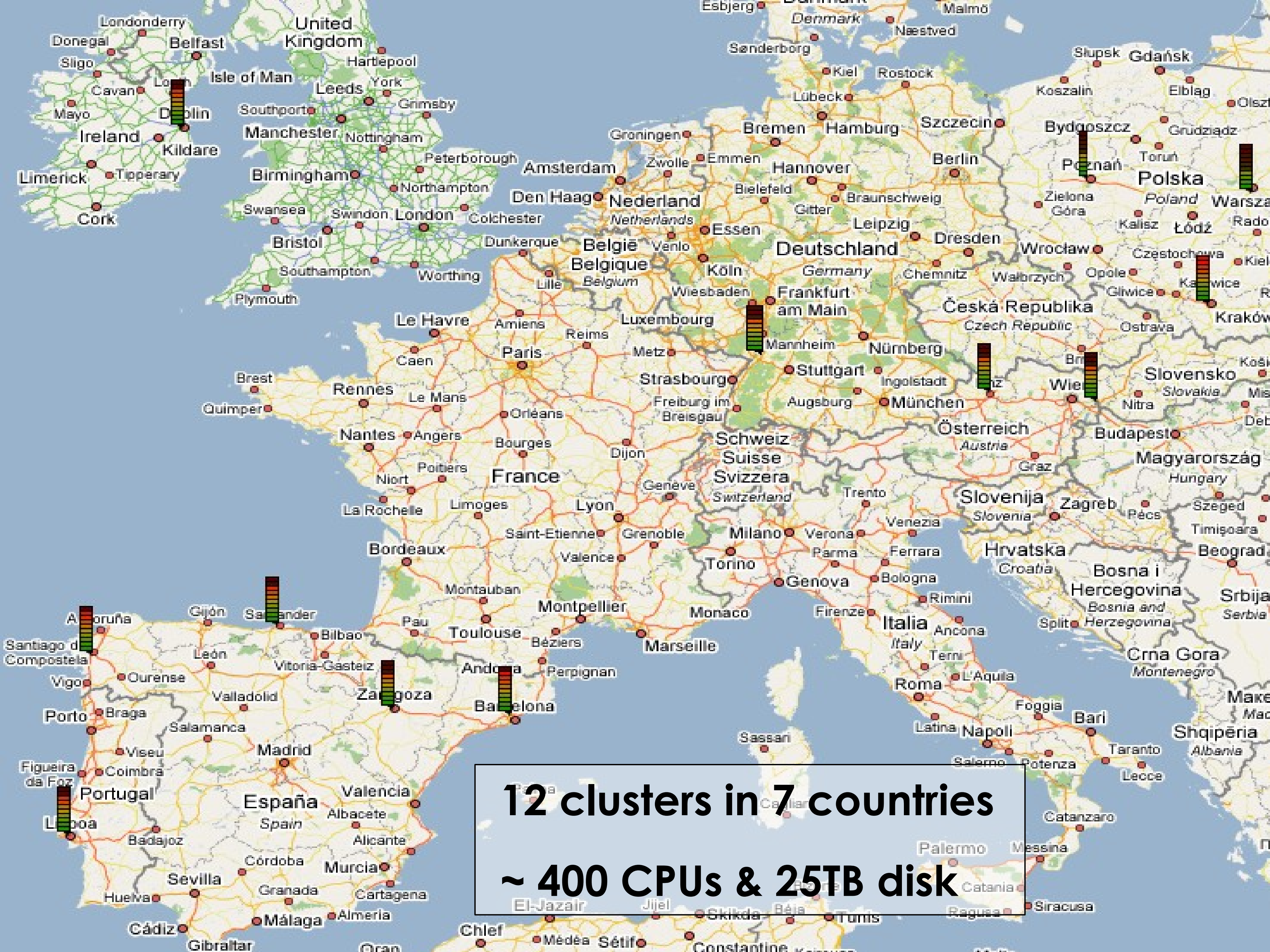
http://www.interactive-grid.eu

Ultrasound-CT on Interactive European Grid

Marcus Hardt
Forschungszentrum Karlsruhe

The grid

- “The” grid
  - Interactive European Grid (int.eu.grid)
  - 2 Year Project (May'06 - April'08)
  - ~20 people
  - Mission
    - 100% gLite compatible
    - MPI for the grid
    - Bring grid to new user communities
    - Improve useability
  - 5 Applications
    - Fusion
    - Medicine (USCT, Brain)
    - Environment
    - Astrophysics

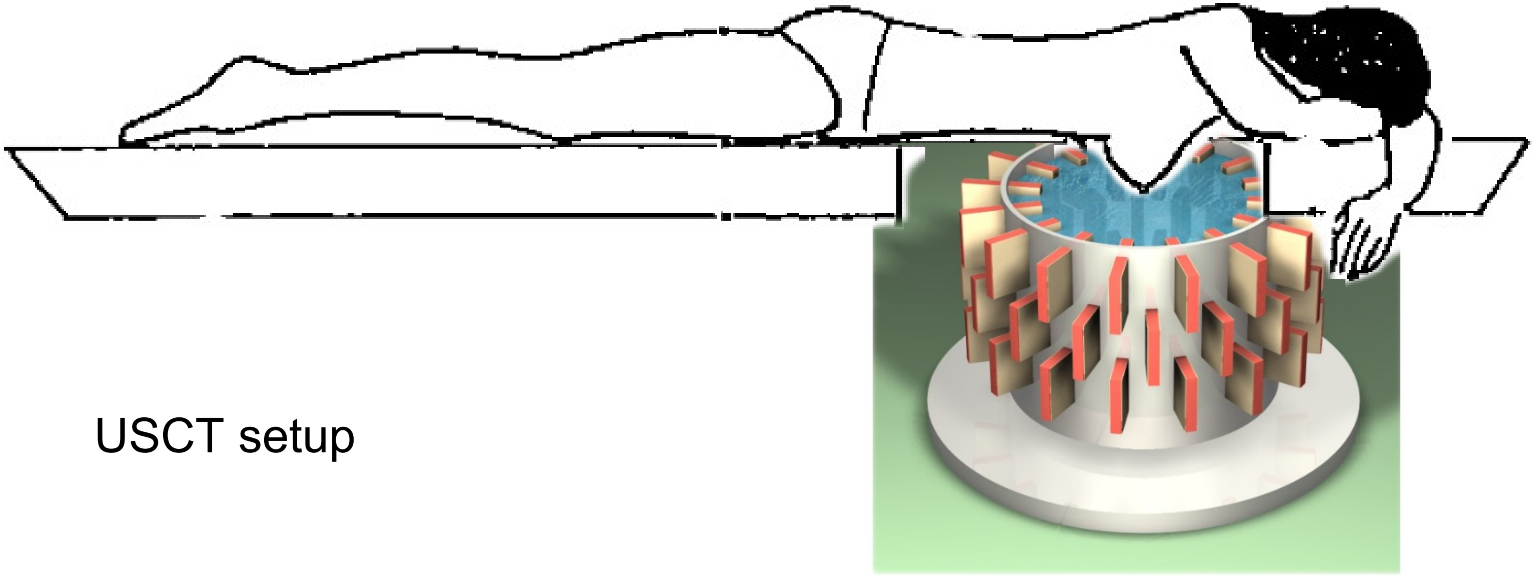


12 clusters in 7 countries

~ 400 CPUs & 25TB disk

The application

- The application: Ultrasound CT (USCT)
  - New method for medical imaging
  - Application: Breast cancer diagnosis



USCT setup

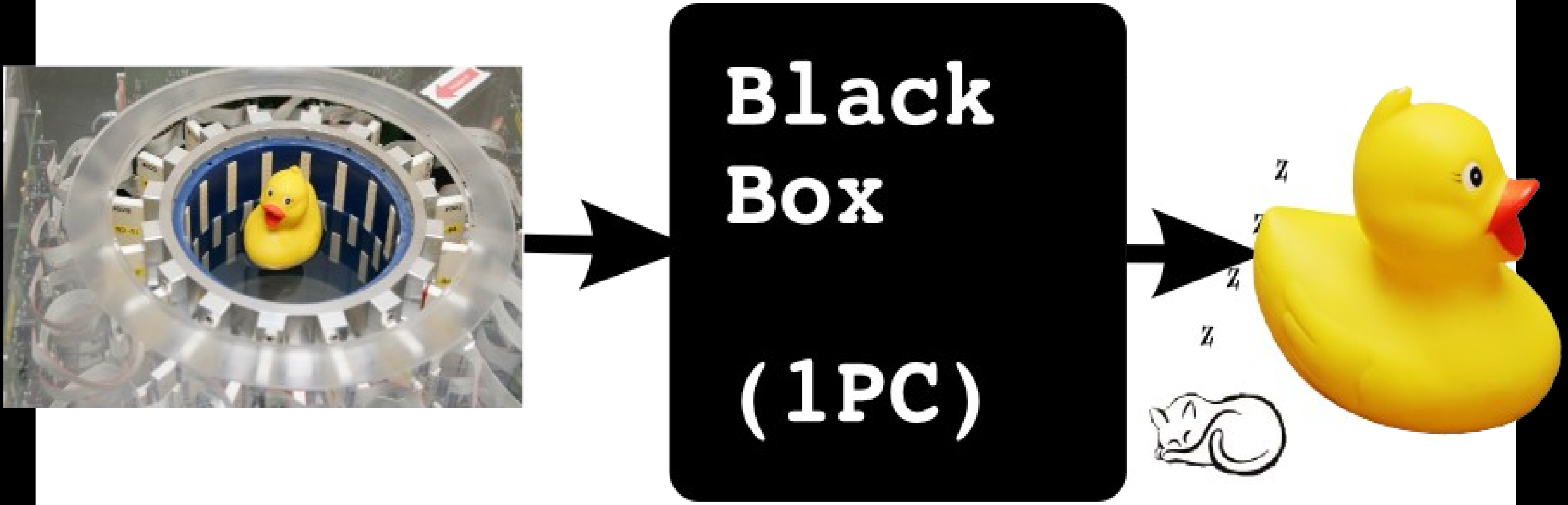
USCT Reconstruction := “Black Box”

- Algorithm:
  - Based on ellipsoidal backprojection (SAFT)
  - Converts ultrasound signals to 3D volume graphics
    - Input: ~ 20GB
    - Output: ~ 8GB
  - Computing time:

•	4096^2(2D)	...	128^2x100	...	4096^2 x 3410
<=>	1hour	...	1.5 Months	...	150 Years

- Matlab
  - Problem solving environment
    - similar to Maple, Mathematica, Scilab ...
  - Strategic development platform
  - Not easy to “submit matlab to the grid”

USCT Reconstruction

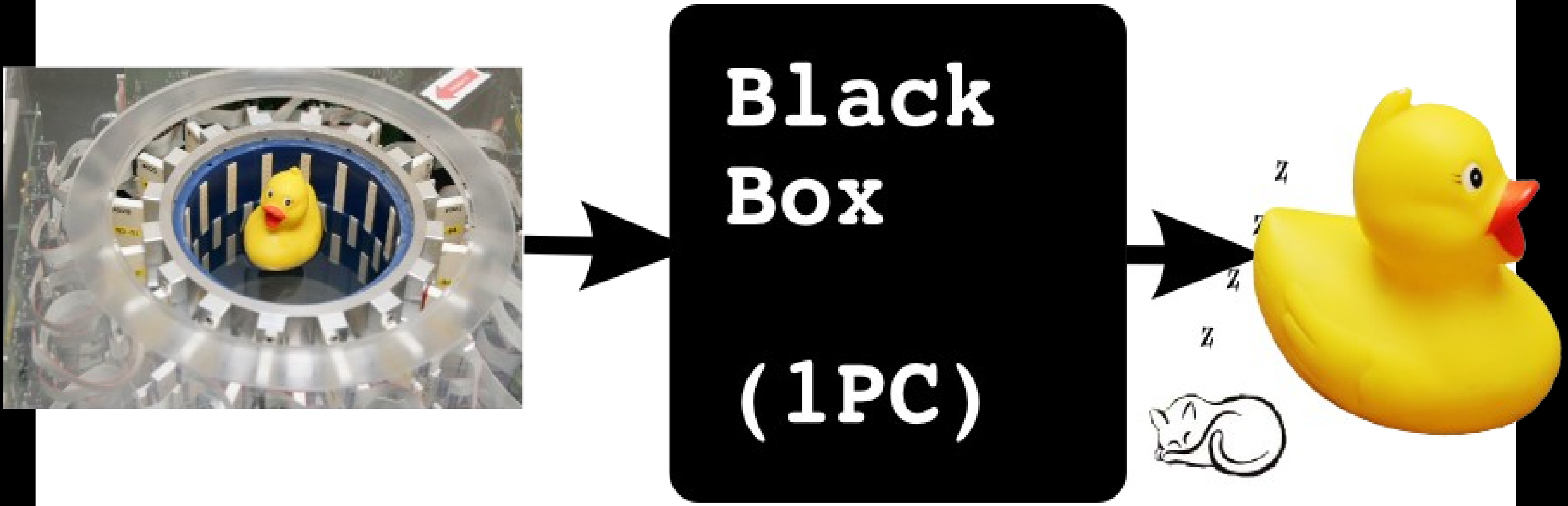


Black Box

(1PC)

- Computation takes long (days, weeks, years)

USCT Reconstruction

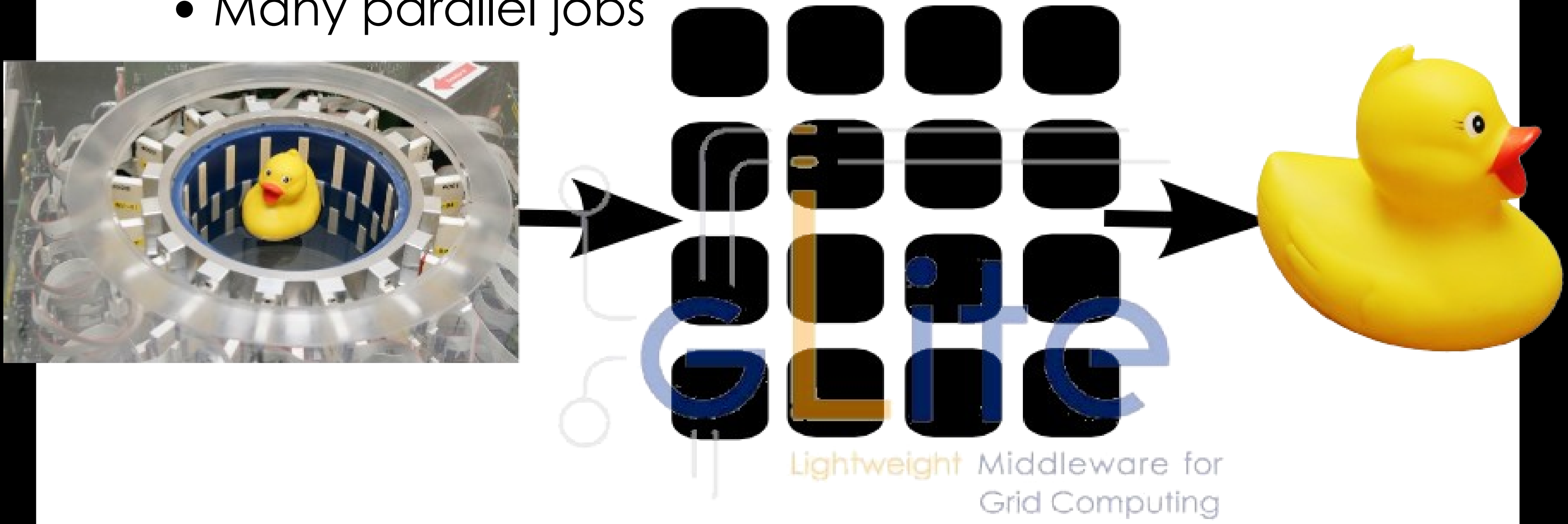

Black Box

(1PC)

- Computation takes long (days, weeks, years)
- **Goal:**
  - **Seamless, interactive, grid access**
  - **from Matlab**

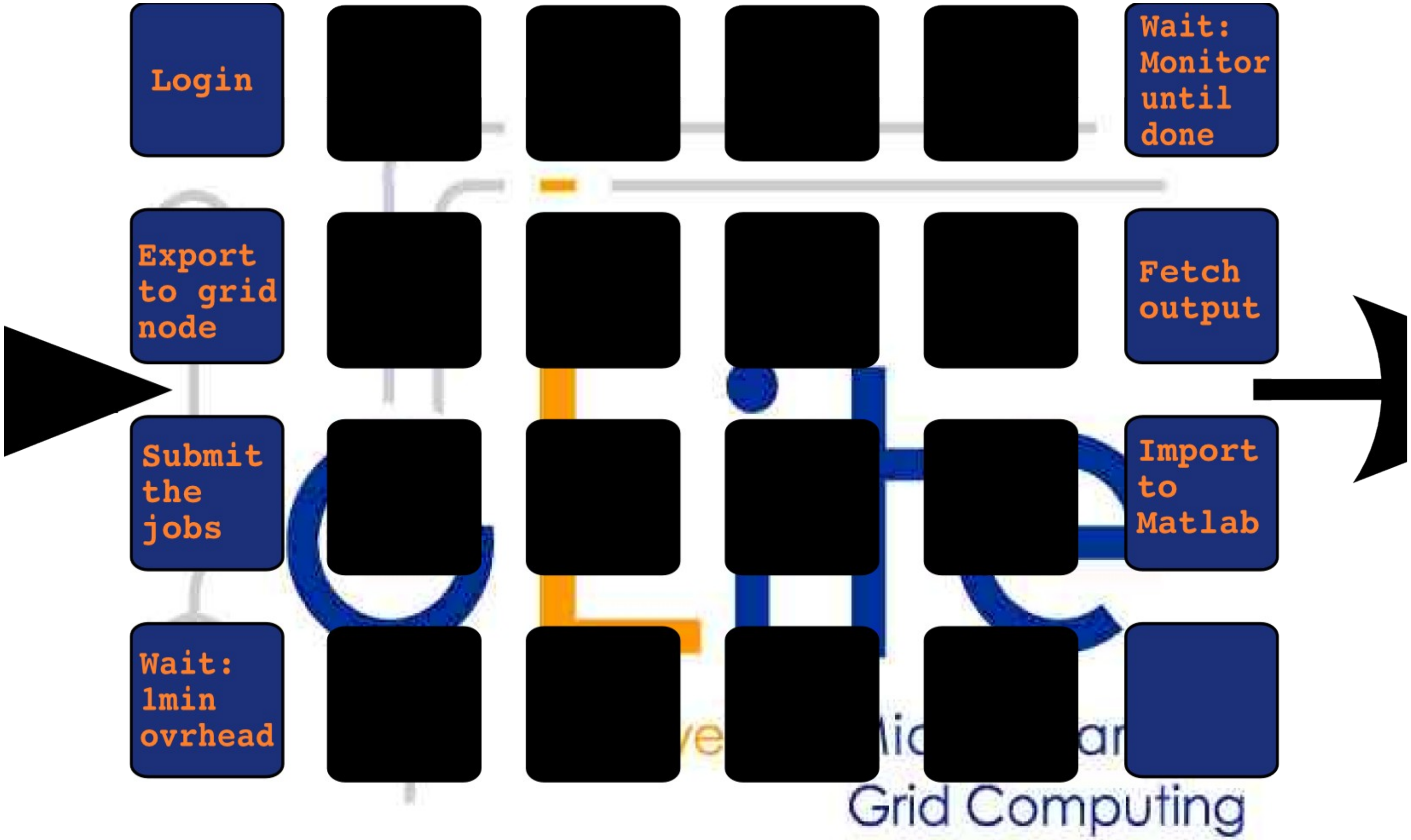
Using the grid

- Simple approach to parallel execution:
  - Partitioning of data
  - Many parallel jobs



Using the grid

- Lets take a close look



Login

Export to grid node

Submit the jobs

Wait: 1min ovrhead

Wait: Monitor until done

Fetch output

Import to Matlab

Using the grid

- Goal:
  - Seamless
  - interactive
  - grid access
  - from Matlab

Using the grid

- Goal:
  - Seamless ✗
  - interactive ✗
  - grid access ✓
  - from Matlab ✗

Using the grid

- Goal:
  - Seamless ✗
  - interactive ✗
  - grid access ✓
  - from Matlab ✗

Usability-test:

=> The users will run away

What's missing?

- **Goal:**
  - **Seamless** ✗
  - **interactive** ✗
  - **grid access** ✓
  - **from Matlab** ✗

- Seamless
  - User might not know if he uses the grid
- Interactive
  - No overhead (< 10 s)
  - No manual data movement
- From Matlab
  - Run Matlab-functions remotely

Improving Grid Access with RPC

- GridSolve
  - Developed at ICL, University Tennessee, Knoxville
  - Implements an RPC client/server solution
  - Client interface for Java, C, Fortran, Matlab, Octave
  - Easy to use:

```
y=problem(x)  <=>  y=gs_call('problem', x)
```

    - Transport input parameters to remote side
    - Execute “problem”
    - Transport result back

=> Reduce complexity of the grid to one function call

How to do it?

1. Integrate GridSolve with glite
2. Make Matlab run on glite

Goal:

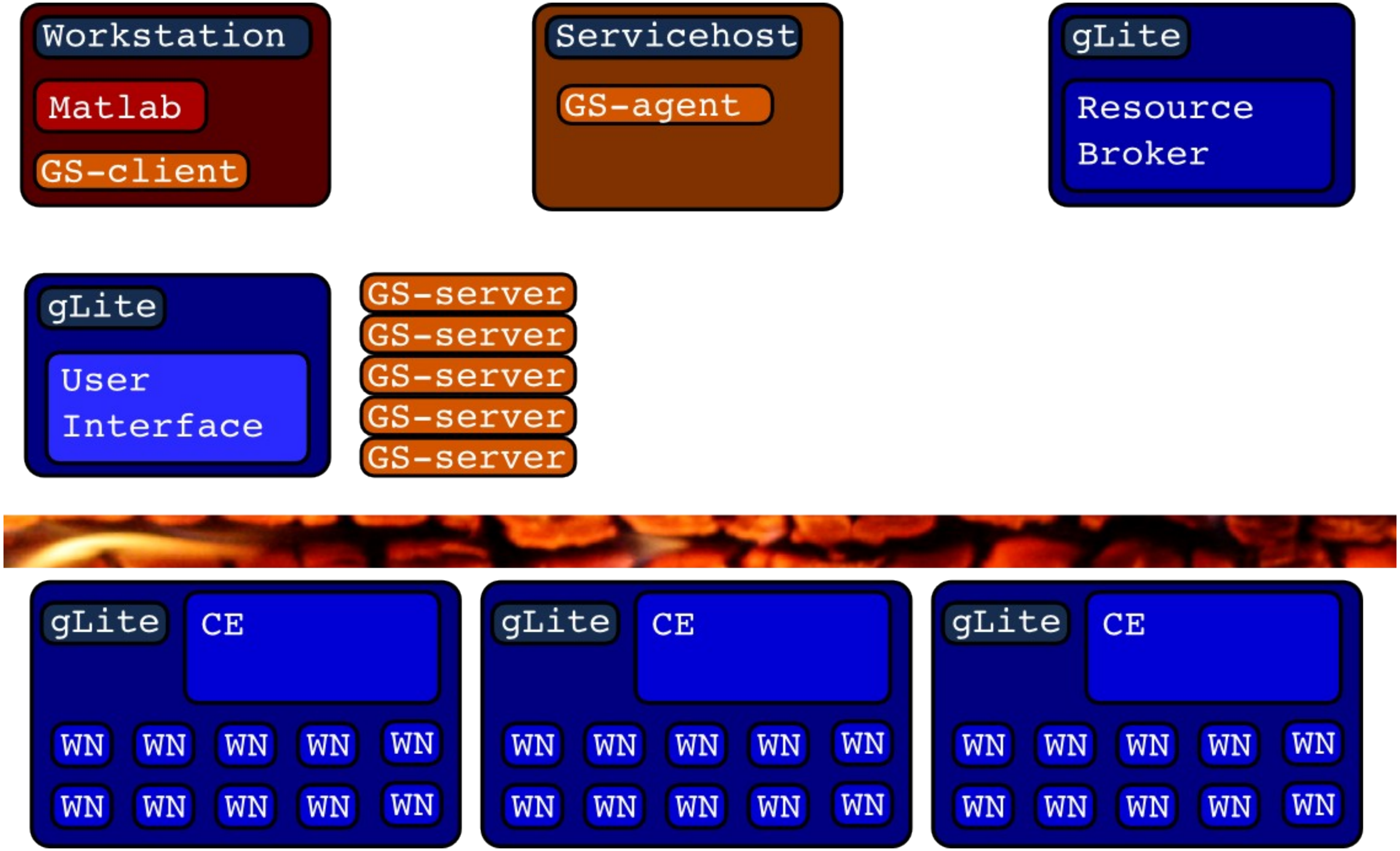
- Seamless ✗
- interactive ✗
- grid access ✓
- from Matlab ✗

Reduce complexity of the grid to one function call

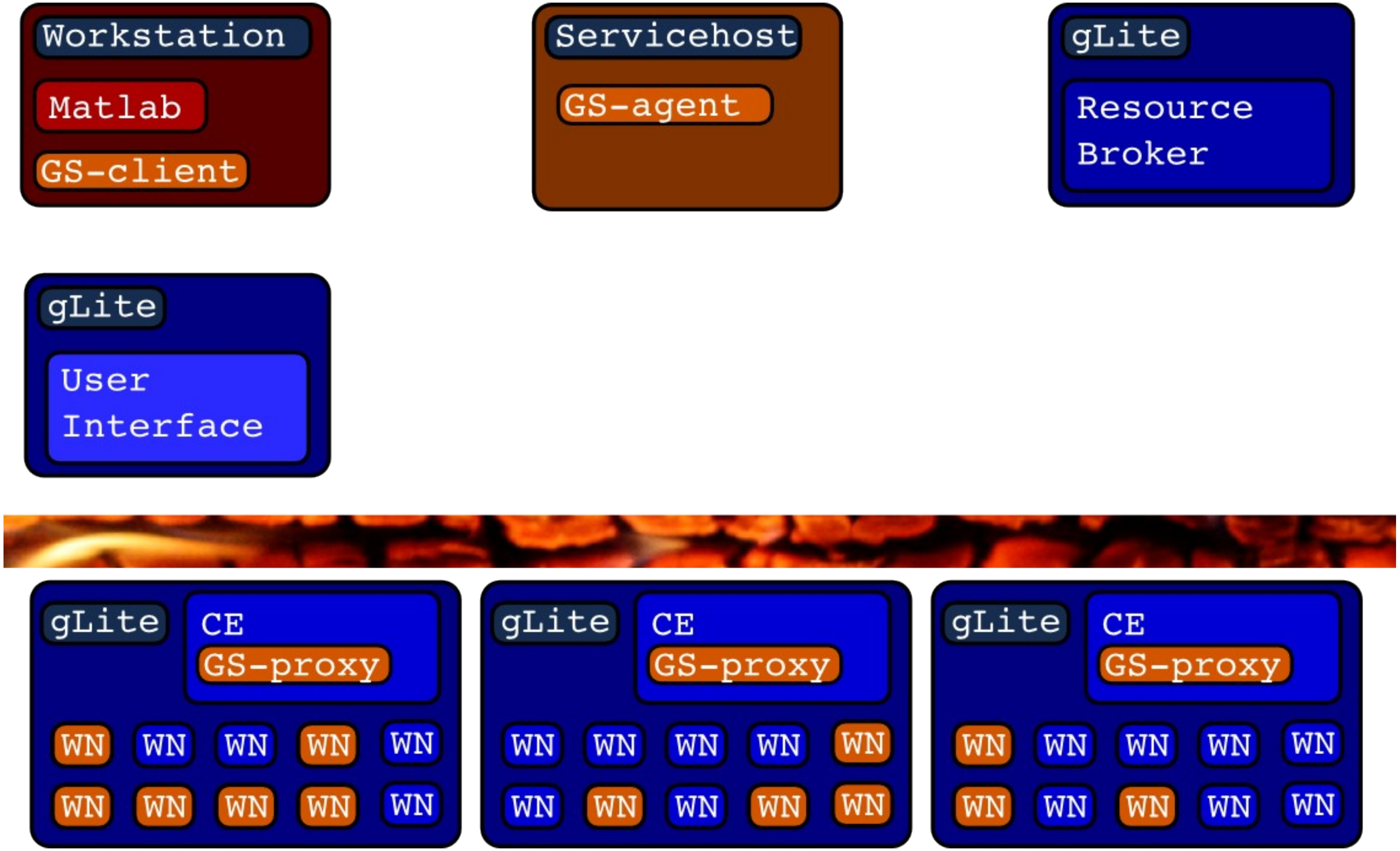
GridSolve(GS)/gLite Integration

- Create GS-Service hosts (GS-agent + GS-proxy)
- Encapsulate GS-server into gLite-job
  - Deployment
    - Compile GS
    - Compile + Link remote procedures (RPCs)
    - Package everything
    - Install GS + RPCs on Workernode (WN)
- Ensure network connectivity
  - GS-Server <=> GS-Proxy <=> GS-agent <=> GS-client

GridSolve startup on gLite



GridSolve ready for action



Matlab/gLite integration

- Matlab Compiler (toolbox)

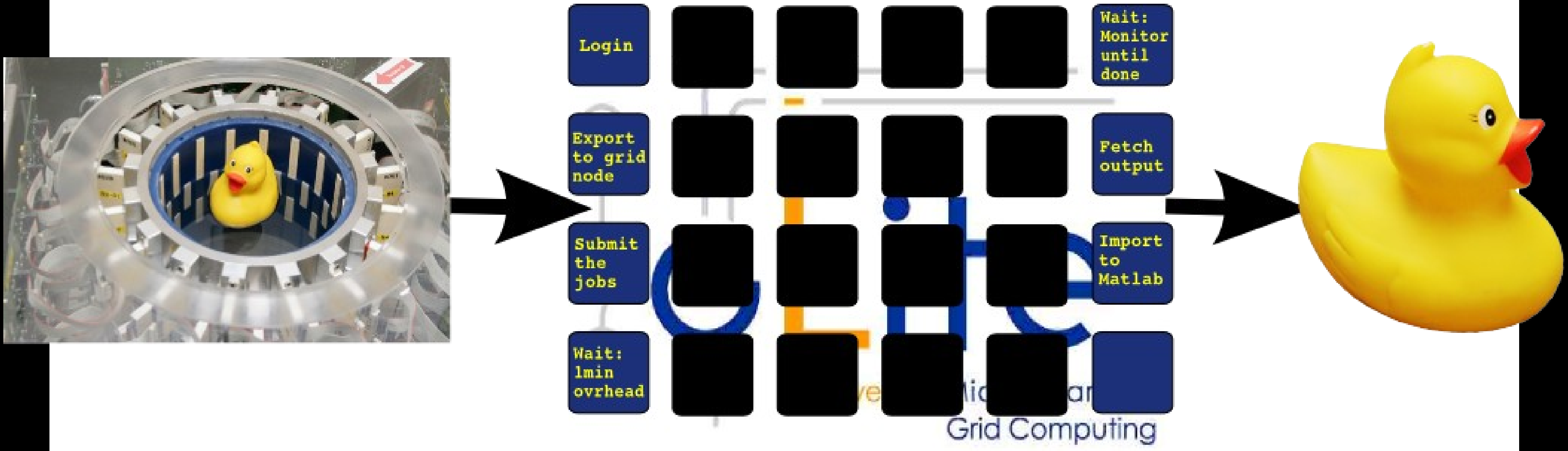
=> Matlab Compiler Runtime (MCR)

  - Install on the fly (as part of glite-job)
  - Fix linux glibc version incompatibility
    - Install new glibc on the fly

- Usability enhancement
  - Access GridSolve from Matlab
  - Point Matlab to service hosts
  - Support for RPC creation
    - Compilation/Linking/Deployment

Putting things together

- RPC with GridSolve
- On top of int.eu.grid/gLite
- Using Matlab functionality



Login

Export to grid node

Submit the jobs

Wait: 1min ovrhead

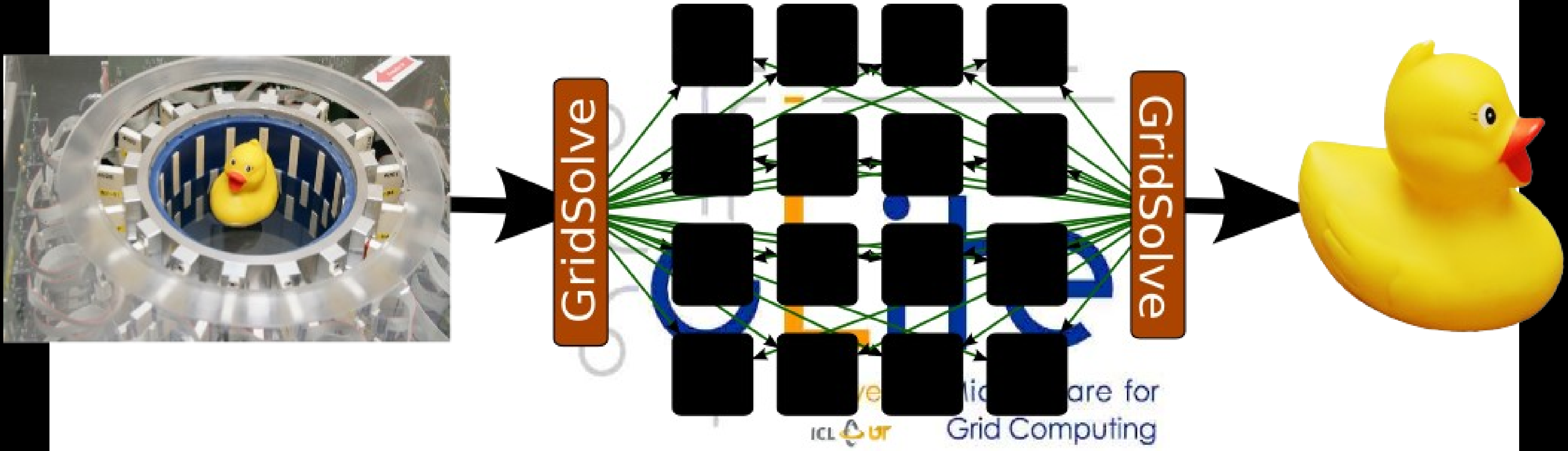
Wait: Monitor until done

Fetch output

Import to Matlab

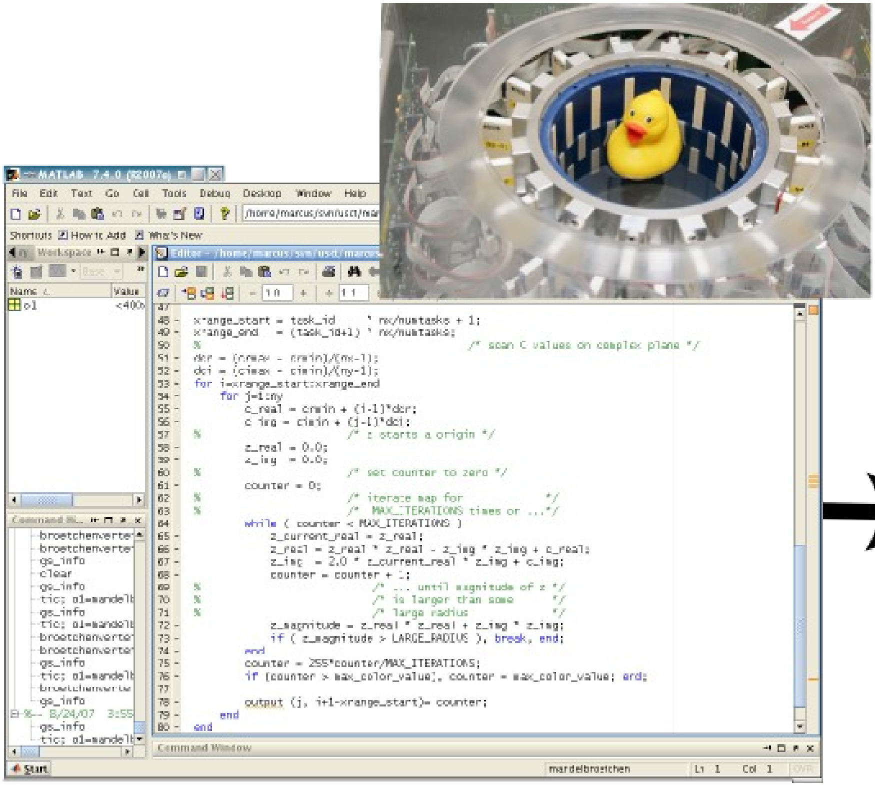
Putting things together

- RPC with GridSolve
- On top of int.eu.grid/gLite
- Using Matlab functionality



Demonstration

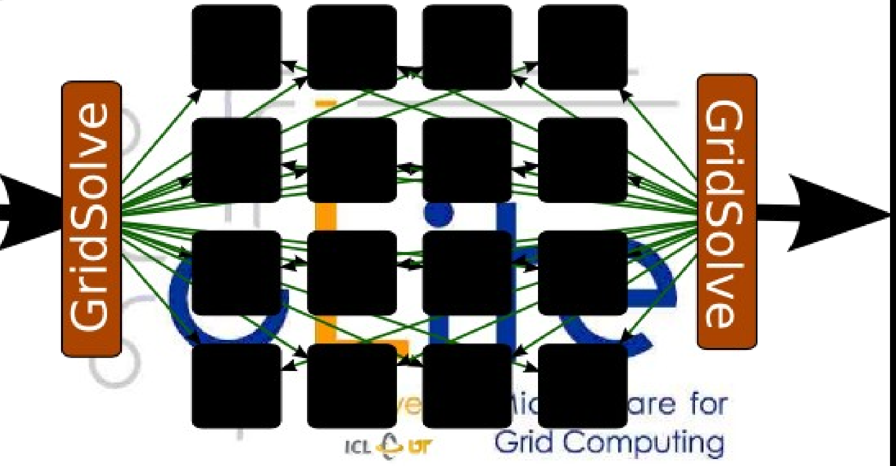
- USCT is a complex Application
  => Currently: Simulation as proof of concept



MATLAB 7.4.0 (R2007a)
File Edit Text Go Cell Tools Debug Desktop Window Help
/home/marcus/svn/usct/mar
Shortcuts How to Add What's New
Workspace
Name Value
o1 <400x
Editor - /home/marcus/svn/usct/marcus
1.0
1.1
Command History
Command Window
Start
mandelbroetchen
Ln 1 Col 1 OVR

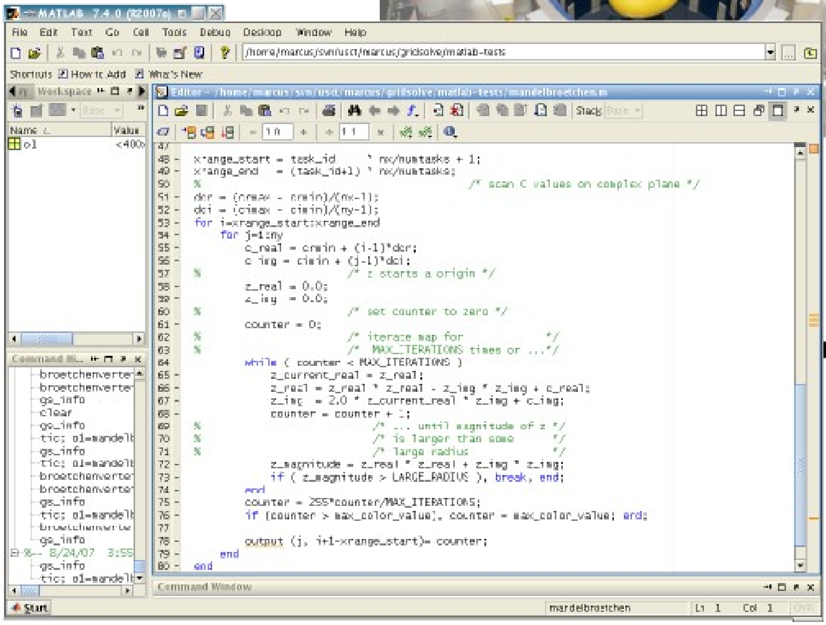
```
x range_start = task_id     * nx/numtasks + 1;
x range_end   = (task_id+1) * nx/numtasks;
%                                          /* scan C values on complex plane */
dcr = (crmax - crmin)/(nx-1);
dci = (cimax - cimin)/(ny-1);
for i=xrange_start:xrange_end
    for j=1:ny
        c_real = crmin + (i-1)*dcr;
        c_img = cimin + (j-1)*dci;
%                      /* z starts a origin */
        z_real = 0.0;
        z_img  = 0.0;
%                      /* set counter to zero */
        counter = 0;
%                      /* iterate map for             */
%                      /*  MAX_ITERATIONS times or ...*/
        while ( counter < MAX_ITERATIONS )
            z_current_real = z_real;
            z_real = z_real * z_real - z_img * z_img + c_real;
            z_img  = 2.0 * z_current_real * z_img + c_img;
            counter = counter + 1;
%                          /* ... until magnitude of z */
%                          /* is larger than some      */
%                          /* large radius             */
            z_magnitude = z_real * z_real + z_img * z_img;
            if ( z_magnitude > LARGE_RADIUS ), break, end;
        end
        counter = 255*counter/MAX_ITERATIONS;
        if (counter > max_color_value), counter = max_color_value; end;

        output (j, i+1-xrange_start)= counter;
    end
end
```

```
broetchenverte
broetchenverte
gs_info
clear
gs_info
tic; o1=mandelb
gs_info
tic; o1=mandelb
broetchenverte
broetchenverte
gs_info
tic; o1=mandelb
broetchenverte
gs_info
%-- 8/24/07  3:55
gs_info
tic; o1=mandelb
```

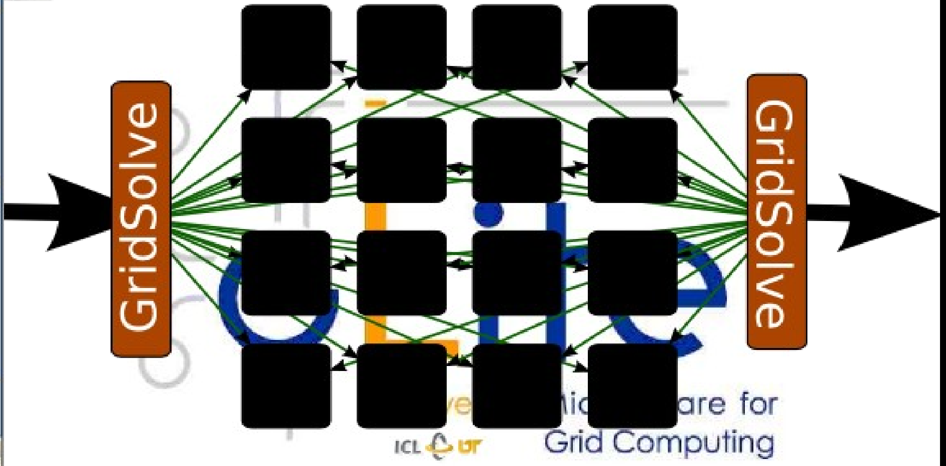


Simulation

- Simulation: Mandelbrot fractal
- Using the same infrastructure


MATLAB 7.4.0 (R2007a)
File Edit Text Go Cell Tools Debug Desktop Window Help
/home/marcus/svn/usct/marcus/gridsolve/matlab-tests
Shortcuts How to Add What's New
Workspace
Name Value
o1 <400x
Editor - /home/marcus/svn/usct/marcus/gridsolve/matlab-tests/mandelbroetchen.m
Stack Base
Command History
Command Window
Start
mandelbroetchen Ln 1 Col 1 OVR

```
xrange_start = task_id     * nx/numtasks + 1;
xrange_end   = (task_id+1) * nx/numtasks;
%                                          /* scan C values on complex plane */
dcr = (crmax - crmin)/(nx-1);
dci = (cimax - cimin)/(ny-1);
for i=xrange_start:xrange_end
    for j=1:ny
        c_real = crmin + (i-1)*dcr;
        c_img = cimin + (j-1)*dci;
%                         /* z starts a origin */
        z_real = 0.0;
        z_img  = 0.0;
%                         /* set counter to zero */
        counter = 0;
%                         /* iterate map for             */
%                         /*  MAX_ITERATIONS times or ...*/
        while ( counter < MAX_ITERATIONS )
            z_current_real = z_real;
            z_real = z_real * z_real - z_img * z_img + c_real;
            z_img  = 2.0 * z_current_real * z_img + c_img;
            counter = counter + 1;
%                             /* ... until magnitude of z */
%                             /* is larger than some      */
%                             /* large radius             */
            z_magnitude = z_real * z_real + z_img * z_img;
            if ( z_magnitude > LARGE_RADIUS ), break, end;
        end
        counter = 255*counter/MAX_ITERATIONS;
        if (counter > max_color_value), counter = max_color_value; end;

        output (j, i+1-xrange_start)= counter;
    end
end
```

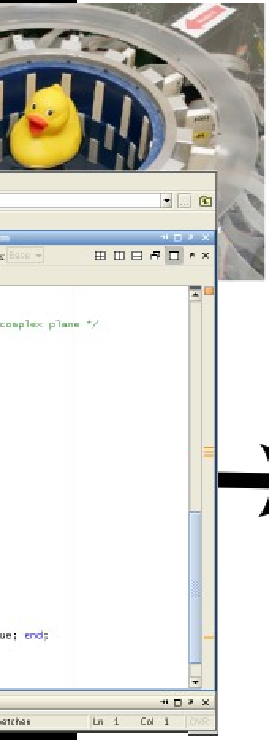


Life-Demo

- Movie of the life demonstration:
  - **`http://marcus.hardt-it.de/grid4matlab`**
- **Real** life demo on int.eu.grid
  - Talk to me (any time during the breaks)

Result

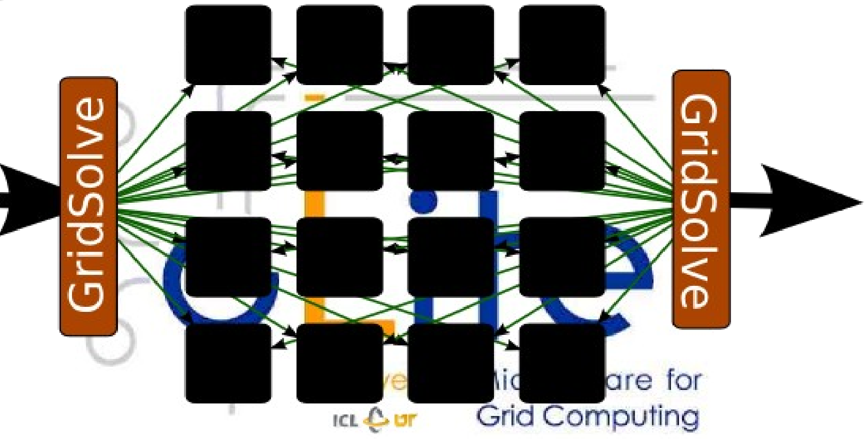
- Simulation works
- Reasonable speedup (7x on 8 machines)

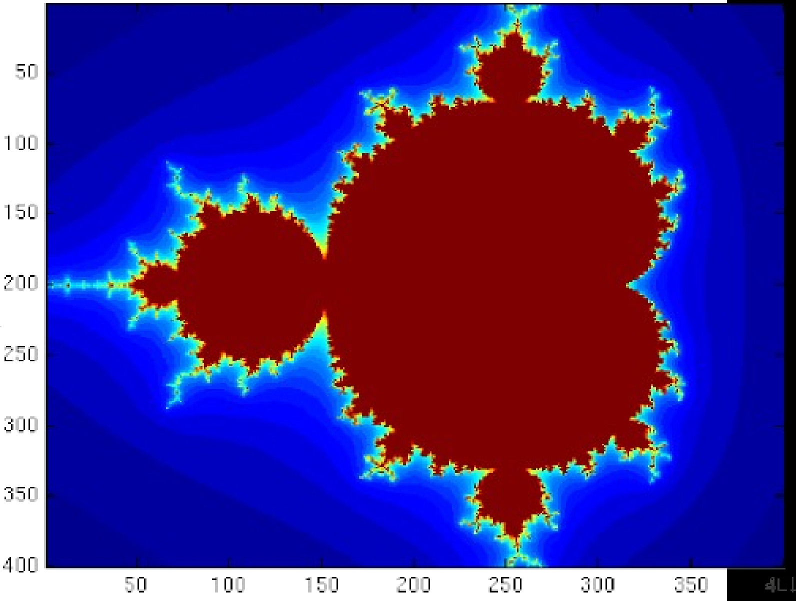


complex plane */

ue; end;

Ln 1 Col 1





Current status

- We can
  - Convert Matlab functions to run on the grid
    - Involved hands-on work
  - Run simple simulations in our infrastructure
  - Use the grid from matlab...

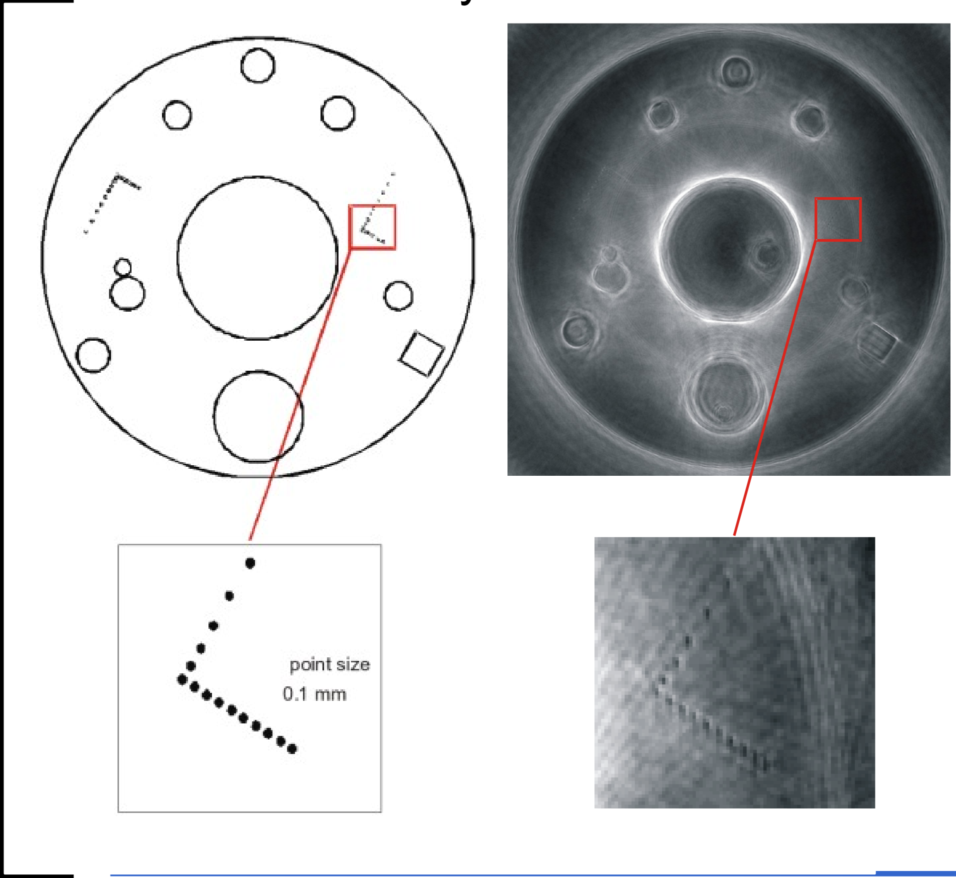
    ... for hand-tuned functions
- We want to...
  - Use real code
    - Cope with the data (20 GB in, 8 GB out)
    - Identify Bottlenecks
  - Automatically send Matlab functions to the grid
    - Reduce hands-on work
  - Data Handling (Future)
  - GFAL + gLite-DICOM

interactive grid

USCT Images

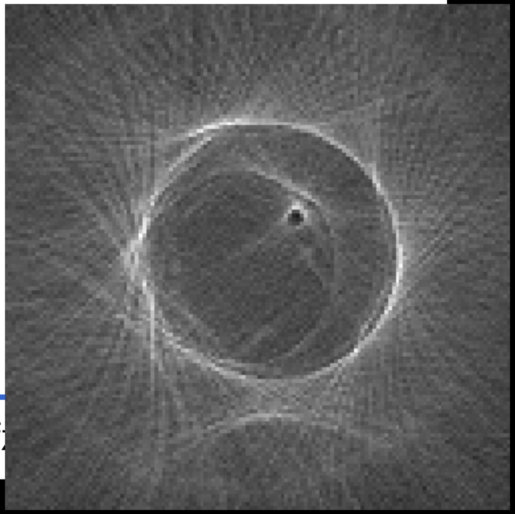
First results with old USCT:

- 0.1 mm Nylon threads visible



Current results with new hardware:

- EGG & Yolk visible
- 3D imaging



Improving grid access

- Idea: Remote Procedure Calls (RPC)
  - Submit daemon(s) as glite job(s)
  - Integrate client into Matlab
  - Connect to daemon(s) from client
    - Call remote procedures from client
    - Transfer input/output parameters
- Advantages:
  - “glite-submit-penalty” only for startup
  - Interactive answer via direct network connection
- Disadvantages:
  - Implement an RPC solution....