Towards “Chemical” Grids

Institute of Computing Technology
Research Center for Grid and Service Computing

Yann Radenac

yann.radenac@software.ict.ac.cn

Outline

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

1. Chemical Programming
2. Chemical Grid Programming
3. Perspectives

Outline

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

1. **Chemical Programming**
2. Chemical Grid Programming
3. Perspectives

Chemical Programming

- Initial work from Jean-Pierre Banâtre and Daniel Le Métayer (1986)
- Programming model using chemistry as a metaphor:
  - data = molecules
  - computation = chemical reactions
- Last development: HOCL

Higher Order Chemical Language

- HOCL: Higher Order Chemical Language
- Based on the γ-calculus
- A HOCL program is a chemical solution of atoms $\langle A_1, \ldots, A_n \rangle$
- Atoms A_i may be:
  - Integers, strings, ... any external object
  - Tuples $A_1 : \cdots : A_k$
  - Sub-solutions
  - One-shot rules: $\mathbf{one}\ P\ \mathbf{by}\ M\ \mathbf{if}\ C$
  - N-shot rules: $\mathbf{replace}\ P\ \mathbf{by}\ M\ \mathbf{if}\ C$

Chemical programming : HOCL

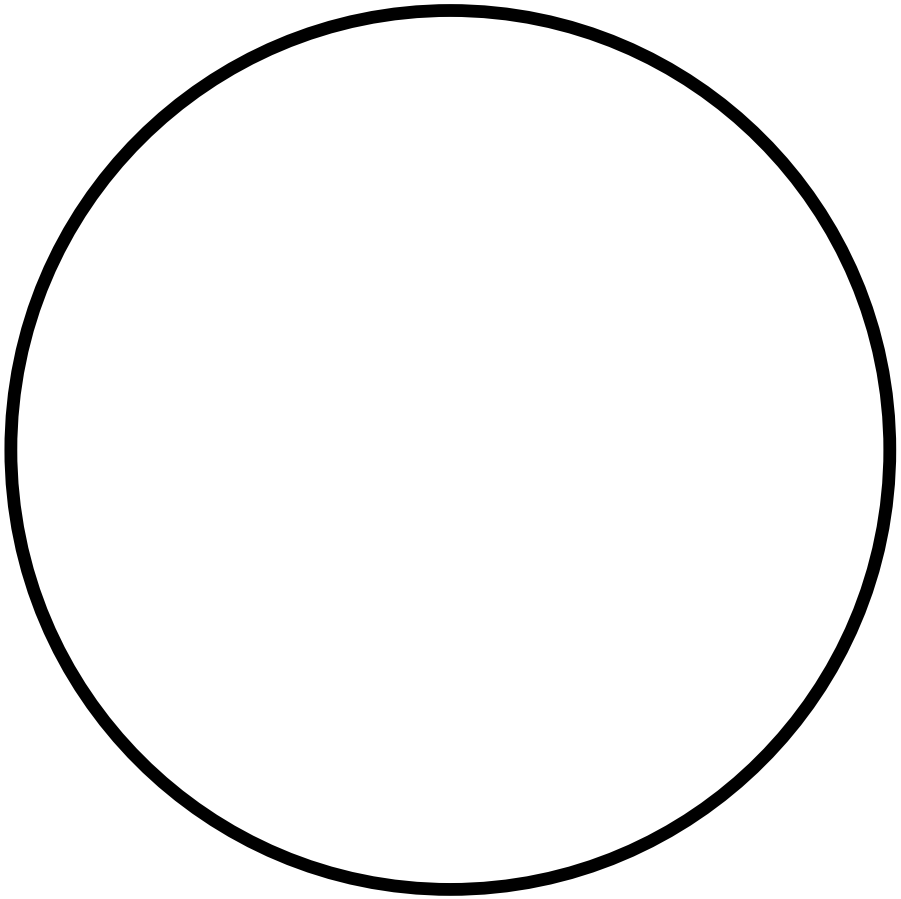
ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

Example: computing the prime numbers lower than 10

Chemical programming : HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

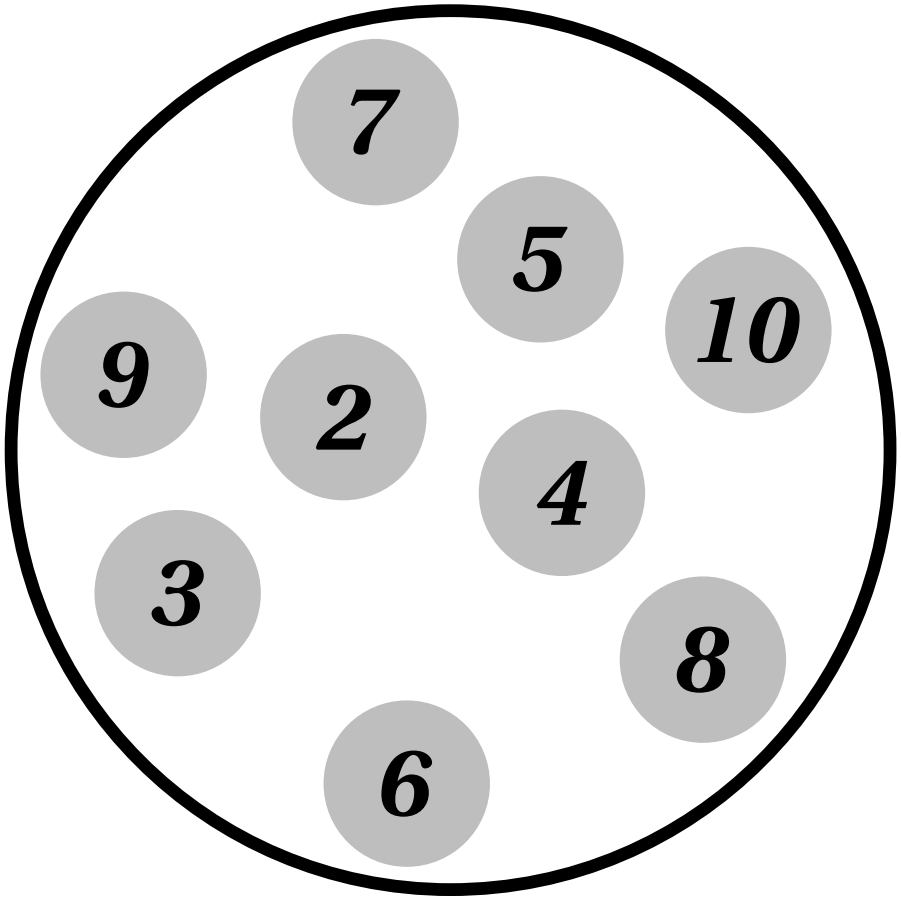
Example: computing the prime numbers lower than 10



Chemical programming : HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

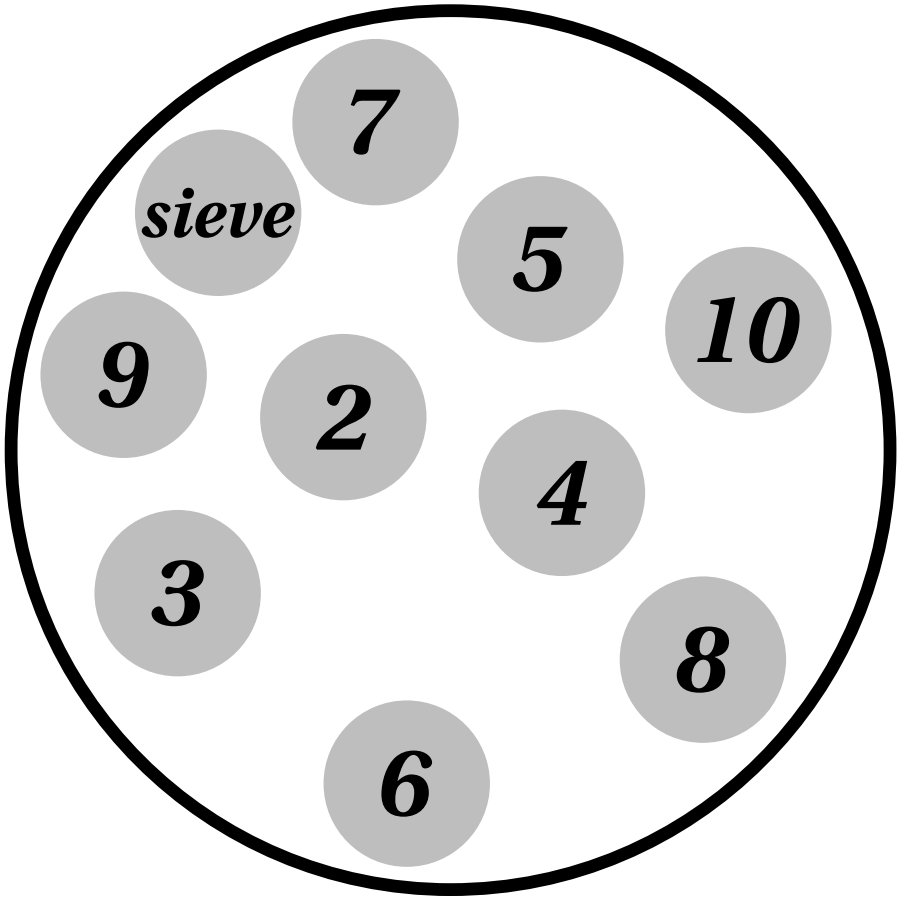
Example: computing the prime numbers lower than 10



Chemical programming : HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

Example: computing the prime numbers lower than 10

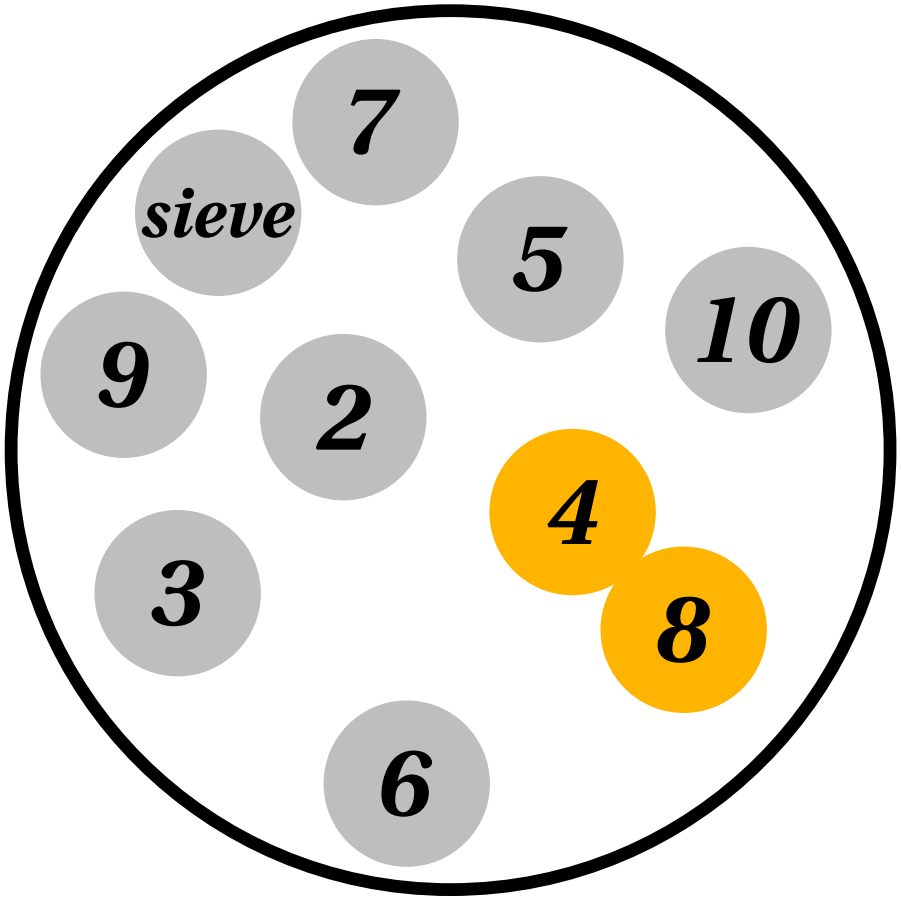


where $\text{sieve} = \textbf{replace}\, x, y\, \textbf{by}\, x\, \textbf{if}\, x \text{ div } y$

Chemical programming : HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

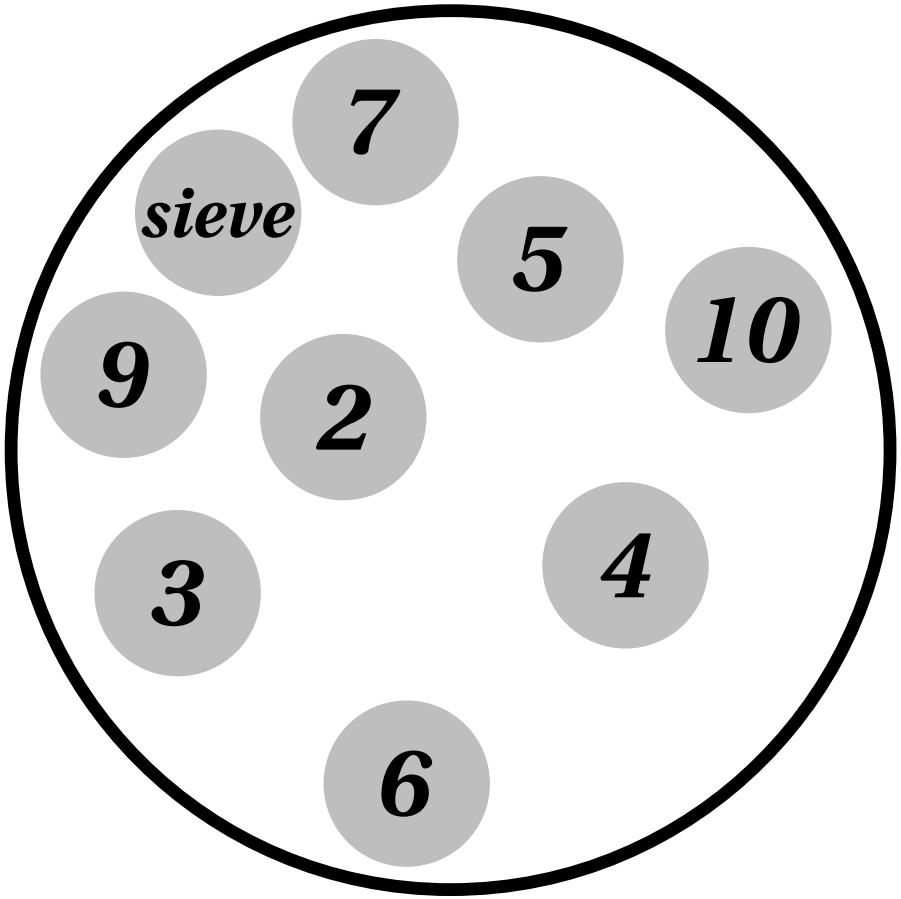
Example: computing the prime numbers lower than 10



where $\text{sieve} = \textbf{replace}\, x, y\, \textbf{by}\, x\, \textbf{if}\, x \text{ div } y$

Chemical programming : HOCL

Example: computing the prime numbers lower than 10

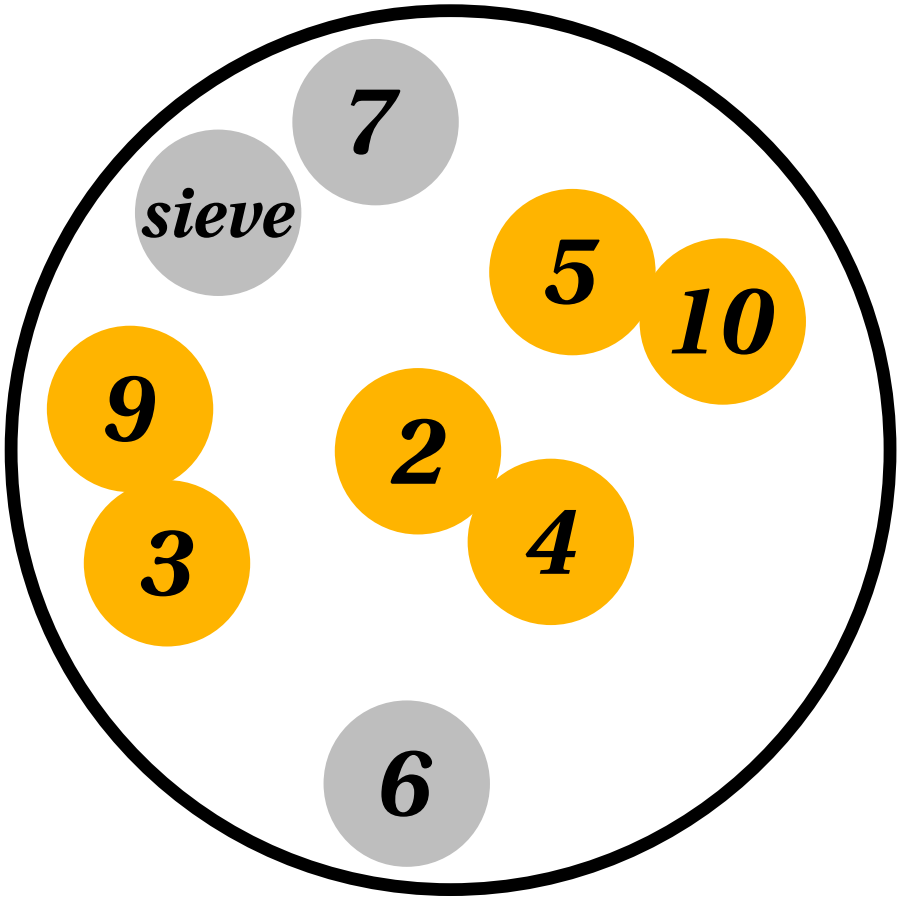


where $\text{sieve} = \textbf{replace}\, x, y\, \textbf{by}\, x\, \textbf{if}\, x \text{ div } y$

Chemical programming : HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

Example: computing the prime numbers lower than 10

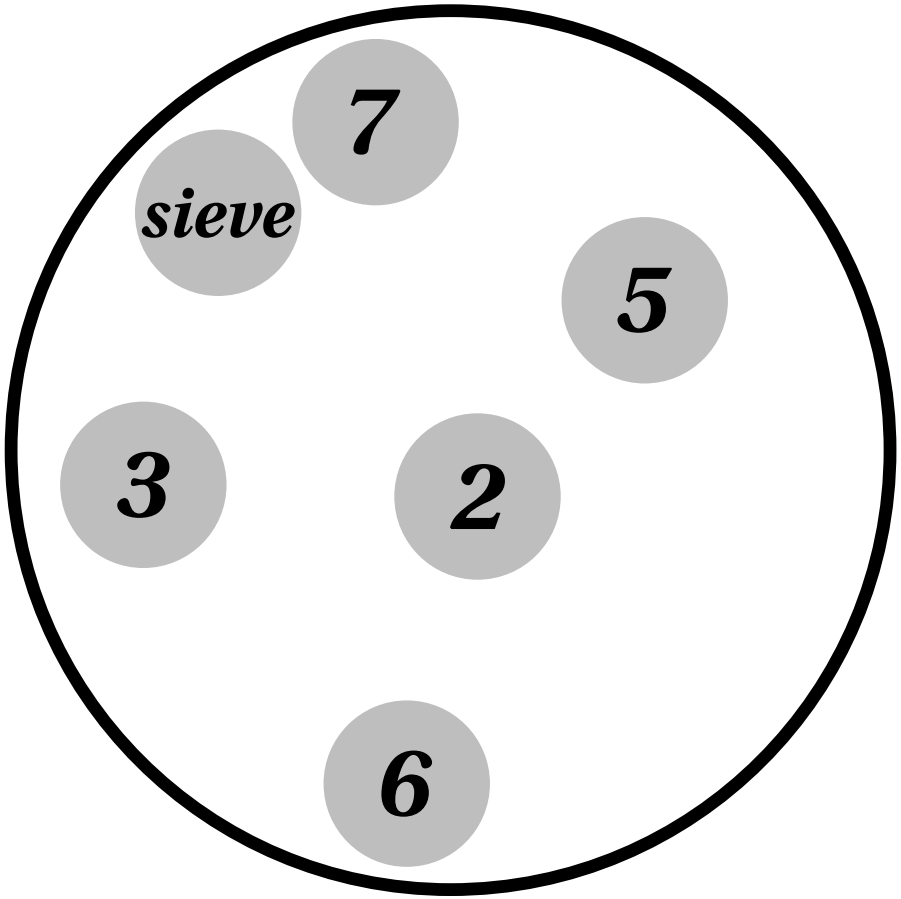


where $\text{sieve} = \textbf{replace}\, x, y\, \textbf{by}\, x\, \textbf{if}\, x \text{ div } y$

Chemical programming : HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

Example: computing the prime numbers lower than 10

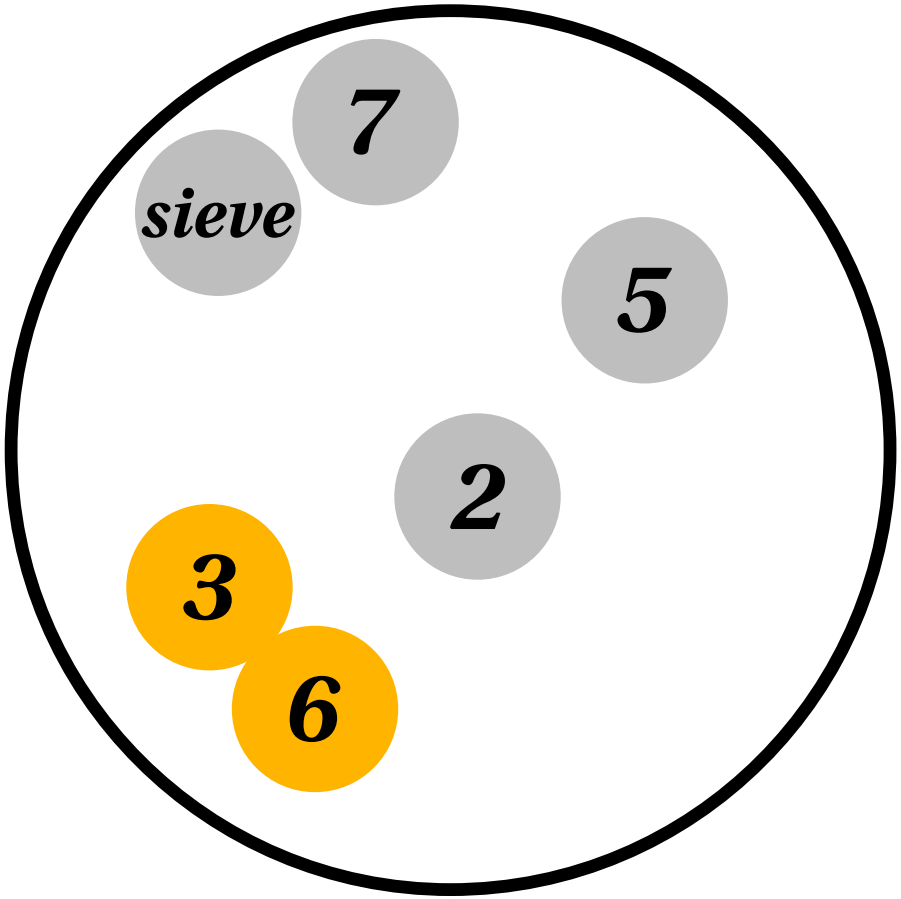


where $\text{sieve} = \textbf{replace}\, x, y\, \textbf{by}\, x\, \textbf{if}\, x \text{ div } y$

Chemical programming : HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

Example: computing the prime numbers lower than 10

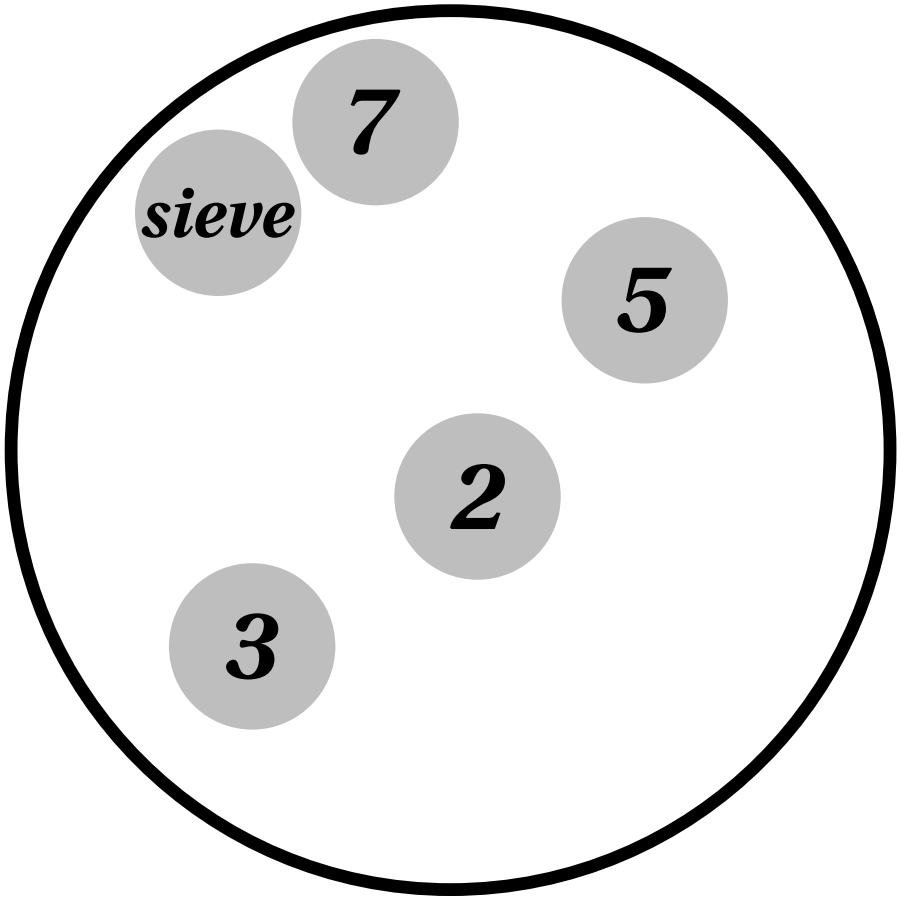


where $\text{sieve} = \textbf{replace}\, x, y\, \textbf{by}\, x\, \textbf{if}\, x \text{ div } y$

Chemical programming : HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

Example: computing the prime numbers lower than 10



where $\text{sieve} = \textbf{replace}\, x, y\, \textbf{by}\, x\, \textbf{if}\, x \text{ div } y$

Outline

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

1. Chemical Programming
2. **Chemical Grid Programming**
3. Perspectives

Grid programming with HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

- Main reasons:
  - Conventional approaches seem to fail

Grid programming with HOCL

- Main reasons:
  - Conventional approaches seem to fail
  - Very high level parallel language: no process, provides atomicity and mutual exclusion, shared memory, etc.

Grid programming with HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

- Main reasons:
  - Conventional approaches seem to fail
  - Very high level parallel language: no process, provides atomicity and mutual exclusion, shared memory, etc.
  - No central control (locality property)

Grid programming with HOCL

- Main reasons:
  - Conventional approaches seem to fail
  - Very high level parallel language: no process, provides atomicity and mutual exclusion, shared memory, etc.
  - No central control (locality property)
  - Chemical metaphor

Grid programming with HOCL

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

- Main reasons:
  - Conventional approaches seem to fail
  - Very high level parallel language: no process, provides atomicity and mutual exclusion, shared memory, etc.
  - No central control (locality property)
  - Chemical metaphor
  - Autonomic programs. . .

Autonomic property

Stabilizing

$$\langle \text{sieve}, 2, 3, 4, 5, 6, 7, 8, 9, 10 \rangle$$

$$\downarrow *$$

$$\langle \text{sieve}, 2, 3, 5, 7 \rangle$$

Autonomic property

Stabilizing

$$\langle \text{sieve}, 2, 3, 4, 5, 6, 7, 8, 9, 10 \rangle$$

$$\downarrow *$$

$$\langle \text{sieve}, 2, 3, 5, 7 \rangle$$

Perturbation

$$\langle \text{sieve}, 2, 3, 5, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 \rangle$$

Autonomic property

$$\langle \text{sieve}, 2, 3, 4, 5, 6, 7, 8, 9, 10 \rangle$$

Stabilizing

$$\downarrow *$$

$$\langle \text{sieve}, 2, 3, 5, 7 \rangle$$

Perturbation

$$\langle \text{sieve}, 2, 3, 5, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 \rangle$$

Re-stabilizing

$$\downarrow *$$

$$\langle \text{sieve}, 2, 3, 5, 7, 11, 13, 17, 19 \rangle$$

Chemical Grid Programming

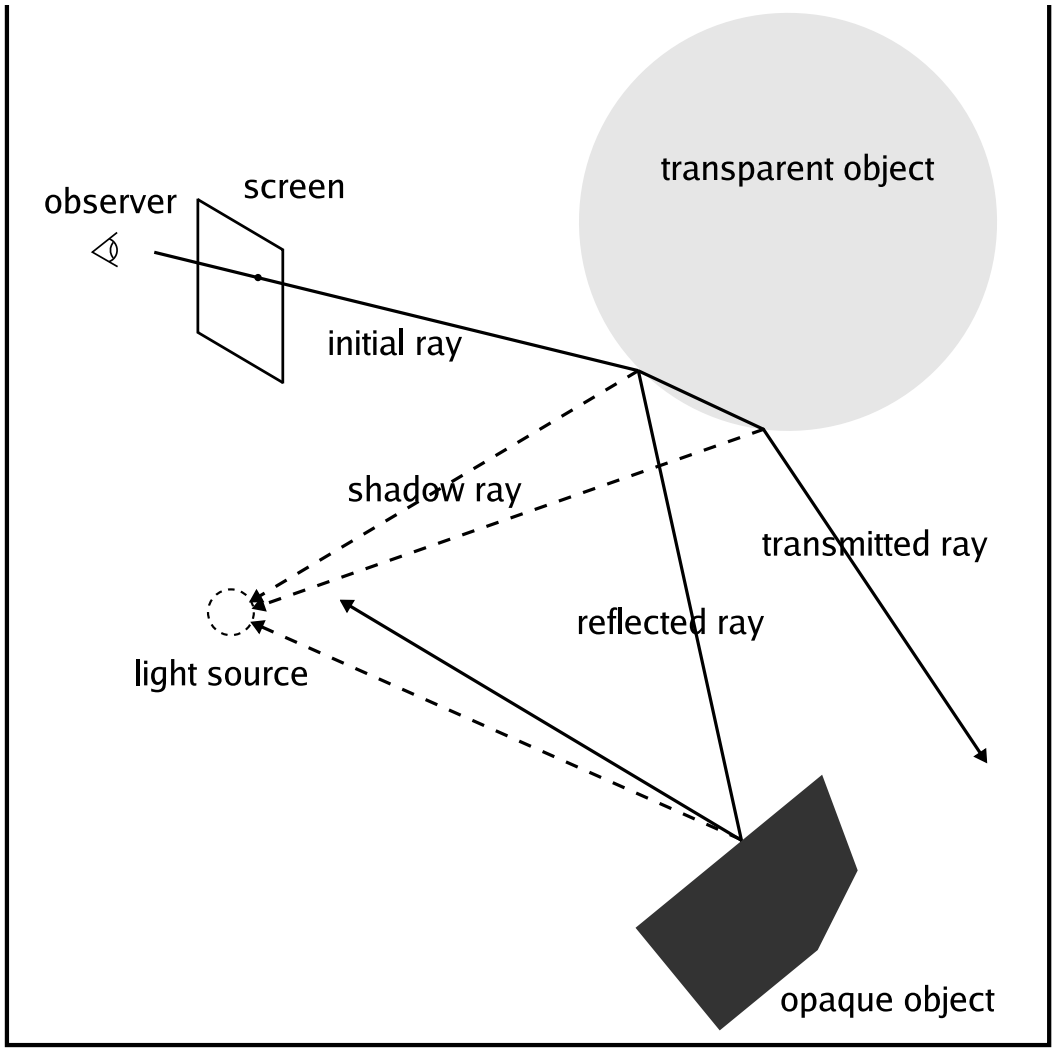
Application level: write applications with HOCL

System level: specification and interface of "chemical" grids with HOCL

Application level

- Applications are chemical programs
- No reference to any grid mechanism
- HOCL used as a coordination language

Appl. level: Ray Tracing Example



Appl. level: Ray Tracing Example

import Pixel, Ray, LightRay, Contrib, Scene.

Appl. level: Ray Tracing Example

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

$$\textbf{import}\ \text{Pixel}, \text{Ray}, \text{LightRay}, \text{Contrib}, \text{Scene}.$$
$$\langle p_1, \ldots, p_n,\ \text{renderPixel} \rangle$$

Appl. level: Ray Tracing Example

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

import Pixel, Ray, LightRay, Contrib, Scene.

$$\langle p_1, \ldots, p_n,\ \text{renderPixel} \rangle$$

renderPixel = **replace** p::Pixel

by $\langle \mathit{firstRay}(p),\ \text{renderRay},\ \text{deleteRay},$
$\text{enlighten},\ \text{sumContrib} \rangle$

Appl. level: Ray Tracing Example

ECHOGRID
EUROPEAN AND CHINESE COOPERATION ON GRID

import Pixel, Ray, LightRay, Contrib, Scene.

$$\langle p_1, \ldots, p_n, \text{ renderPixel} \rangle$$

renderPixel = **replace** p::Pixel
by $\langle \mathit{firstRay}(p), \text{ renderRay}, \text{ deleteRay}, \text{ enlighten}, \text{ sumContrib} \rangle$

renderRay = **replace** r::Ray
by Scene.$\mathit{intersectRays}(r)$
if $r.\mathit{contribution}() > \epsilon$

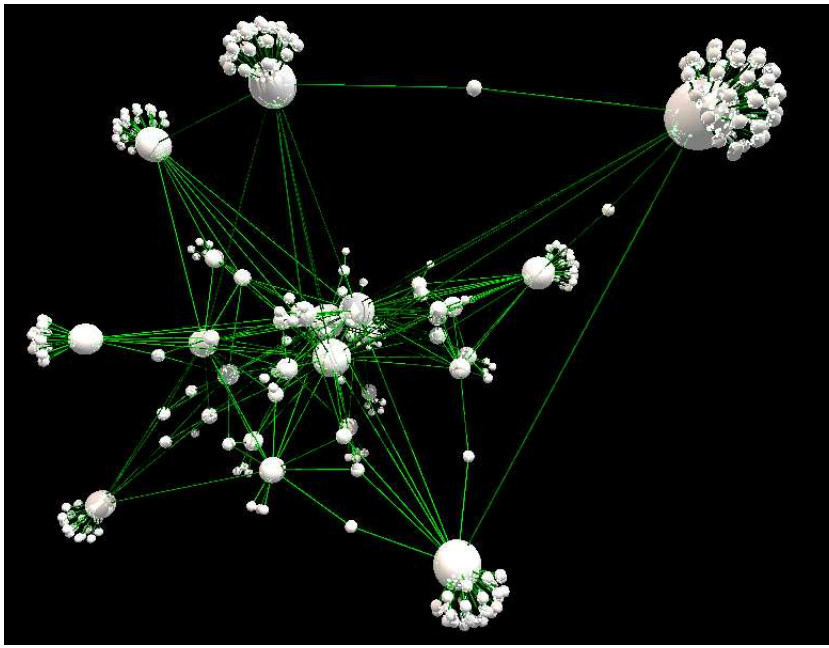
deleteRay = **replace** r::Ray, ω **by** ω **if** $r.\mathit{contribution}() \leq \epsilon$

enlighten = **replace** l::LightRay
by $l.\mathit{computeContrib}()$

sumContrib = **replace** c_1::Contrib, c_2::Contrib
by $c_1.\mathit{add}(c_2)$

System level

- Grid viewed as a chemical solution:
  - resources = molecules
  - coordination = chemical reactions
- Chemical program as:
  - a specification
  - an interface for the grid administrator



Credit: Andrew Wood, Nick Drew, Russell Beale and Bob Hendley, 1995.

Sys. level: Chemical Grid Ray Tracing

$$\langle \mathrm{R_1}{:}\langle\langle r_1\rangle,\ldots,\langle r_n,\rangle\rangle,\ \mathrm{R_2}{:}\langle\rangle,\ \ldots,\ \mathrm{R_n}{:}\langle\rangle,\ renderRules,\ \mathrm{split},\ \mathrm{merge},\ \mathrm{newRes},\ \mathrm{remRes}\rangle$$

Sys. level: Chemical Grid Ray Tracing

$$\langle \mathrm{R_1}{:}\langle\langle r_1\rangle,\ldots,\langle r_n,\rangle\rangle,\ \mathrm{R_2}{:}\langle\rangle,\ \ldots,\ \mathrm{R_n}{:}\langle\rangle,\ renderRules,\ \mathrm{split},\ \mathrm{merge},\ \mathrm{newRes},\ \mathrm{remRes}\rangle$$

$$\mathrm{split} = \mathbf{replace}\ res_1{:}\langle\langle r{::}\mathrm{Ray},\ \omega_p\rangle,\ \omega_1\rangle,\ res_2{:}\langle\omega_2\rangle \\ \mathbf{by}\ res_1{:}\langle\omega_1\rangle,\ res_2{:}\langle\langle r,\ \omega_p\rangle,\ \omega_2\rangle \\ \mathbf{if}\ res_1.load() >> res_2.load()$$

Sys. level: Chemical Grid Ray Tracing

$$\langle \mathrm{R}_1{:}\langle\langle r_1\rangle,\ldots,\langle r_n,\rangle\rangle,\ \mathrm{R}_2{:}\langle\rangle,\ \ldots,\ \mathrm{R}_\mathrm{n}{:}\langle\rangle,\ renderRules,\ \mathrm{split},\ \mathrm{merge},\ \mathrm{newRes},\ \mathrm{remRes}\rangle$$

$$\mathrm{split} = \mathbf{replace}\ res_1{:}\langle\langle r{::}\mathrm{Ray},\ \omega_p\rangle,\ \omega_1\rangle,\ res_2{:}\langle\omega_2\rangle \ \mathbf{by}\ res_1{:}\langle\omega_1\rangle,\ res_2{:}\langle\langle r,\ \omega_p\rangle,\ \omega_2\rangle \ \mathbf{if}\ res_1.load() >> res_2.load()$$

$$\mathrm{merge} = \mathbf{replace}\ res_1{:}\langle\omega_1\rangle,\ res_2{:}\langle\langle c{::}\mathrm{Contrib}\rangle,\ \omega_2\rangle \ \mathbf{by}\ res_1{:}\langle\omega_1,\ \langle c\rangle\rangle,\ res_2{:}\langle\omega_2\rangle \ \mathbf{if}\ res_1 < res_2$$

Outline

1. Chemical Programming
2. Chemical Grid Programming
3. **Perspectives**

Perspectives

- Currently EchoGrid Fellow at the ICT
- Initial work:

  Application level: implement HOCL using GSML

谢谢

Thank You!