www.CactusCode.org

Supporting Efficient Execution in Heterogeneous Distributed Computing Environments with Cactus and Globus

Gabrielle Allen*, Thomas Dramlitsch*, Ian Foster†, Nicolas Karonis‡, Matei Ripeanu#, Ed Seidel*, Brian Toonen†

* Max-Planck-Institut für Gravitationsphysik
†Argonne National Labs
‡Northern Illinois University
#University of Chicago

www.CactusCode.org

This talk is about

- **Large scale distributed computing**
  what,why & recent experiments, results
- **Short review of problems of executing codes in grid environments**
  (networks, algorithms, infrastructure etc.)
- **Introducing a framework for distributed computing**
  how **CACTUS**, **GLOBUS** and **MPICH-G2** together form a complete set of tools to for easy execution of codes in grid environments
- **The status of distributed computing**
  where we are, what we can do now

Major Problems of Metacomputing

•Heterogeneity
different operating systems, different queue systems, different authentication schemes, different processors/processor speeds

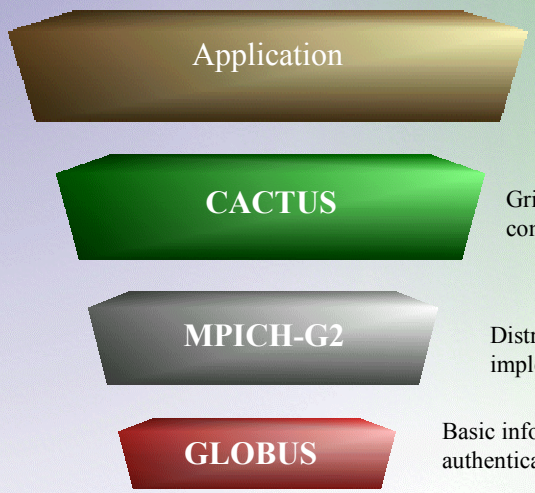
•Networks
wide area networks are getting faster every day, but are still orders of magnitude slower than intra-machine networks of supercomputers

•Algorithms
Most parallel codes use communication schemes, processor distributions and algorithms which are written for single machine execution (i. e. *unaware* of the nature of a grid environment)

(see sc95,sc98,may 2001,now)

the globus project™ www.globus.org
www.CactusCode.org

Layered structure of the framework



Numerical application, unaware of the grid

Grid-aware parallelizing- and communication-algorithms

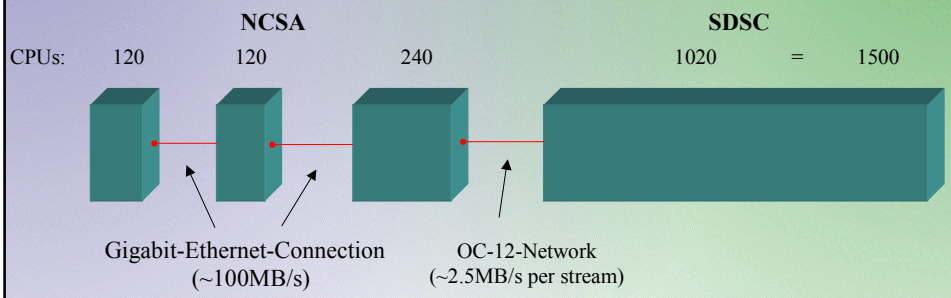
Distributed high-performance implementation of MPI

Basic information about job, infrastructure, authentication, queues, resources, etc.

the globus project
www.globus.org

www.CactusCode.org

First test: Distributed Teraflop Computing (DTF)



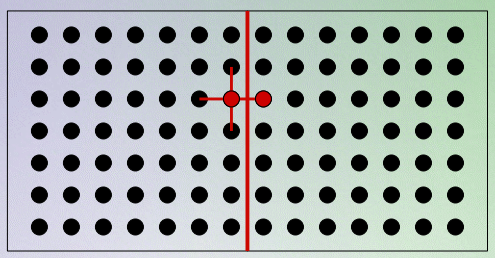
The code computed the evolution of gravitational waves, according to Einstein's theory of general relativity.

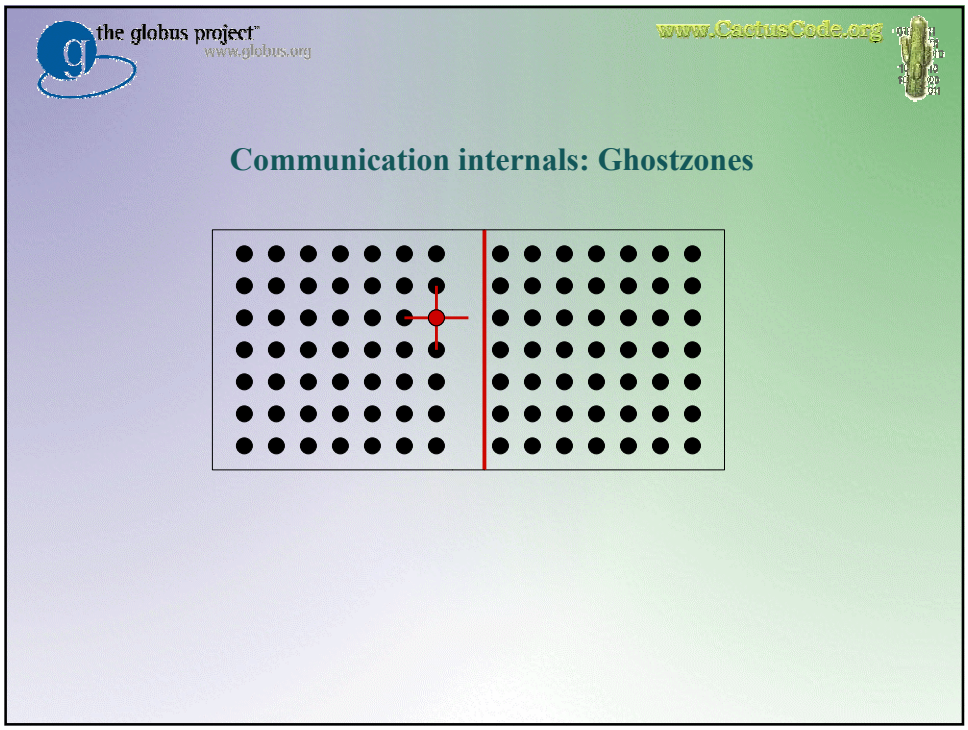
The setup included all major problems: multiple sites/authentication, heterogenity, slow networks, different queue systems, MPI-implementations ...

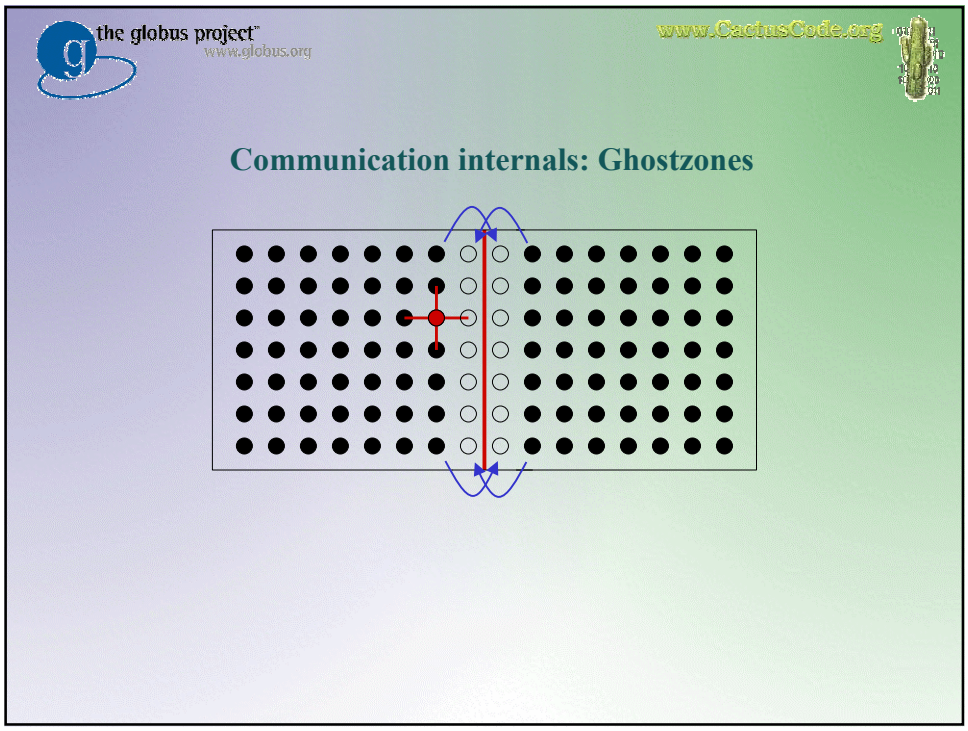
the globus project
www.globus.org

www.CactusCode.org

Communication internals: Ghostzones



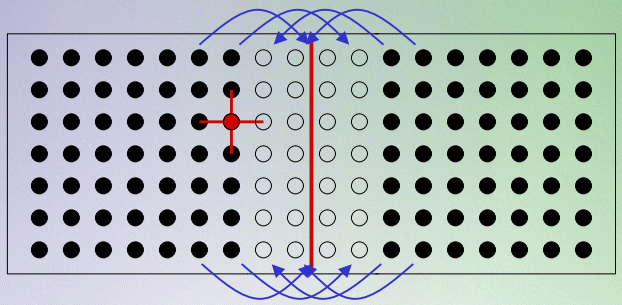

the globus project
www.globus.org
www.CactusCode.org
Communication internals: Ghostzones


the globus project
www.globus.org
www.CactusCode.org
Communication internals: Ghostzones

the globus project
www.globus.org

www.CactusCode.org

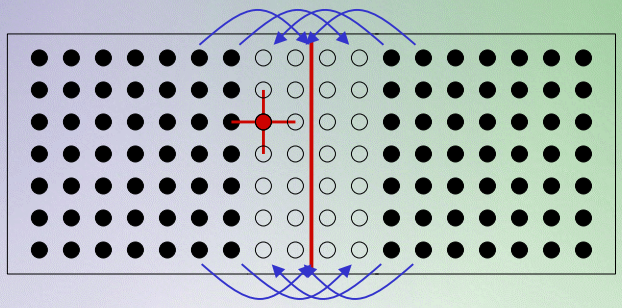
Communication internals: Ghostzones



the globus project
www.globus.org

www.CactusCode.org

Communication internals: Ghostzones

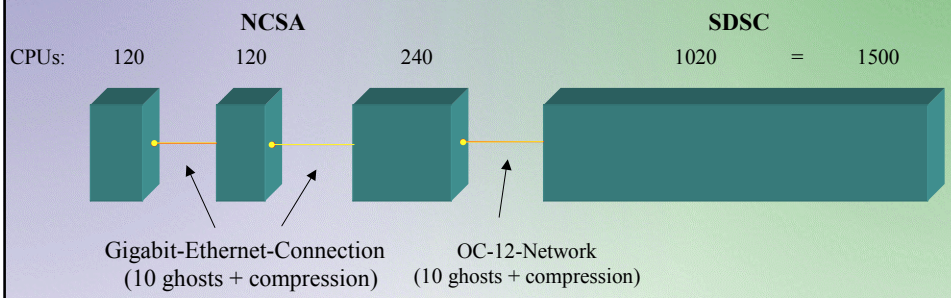


In the DTF run we used a ghostzone size of **10**

the globus project
www.globus.org

www.CactusCode.org

DTF Setup



Eficciency: 63% for 1500 CPU run and
88% for 1140 CPU run

Without ghostzone + compression: **~15%**

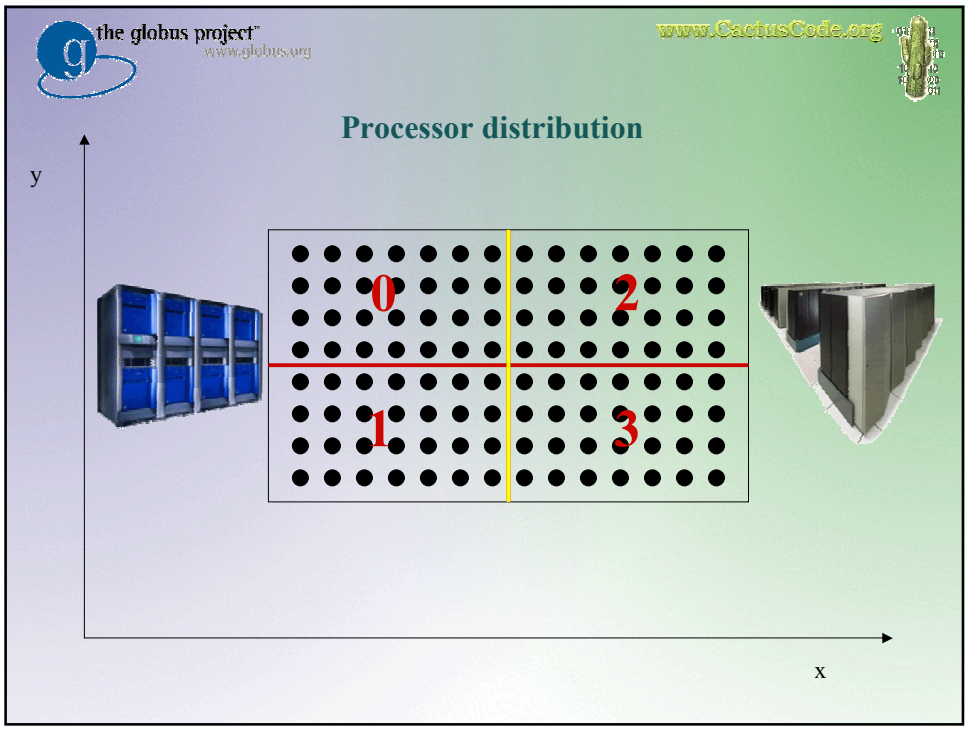
the globus project
www.globus.org

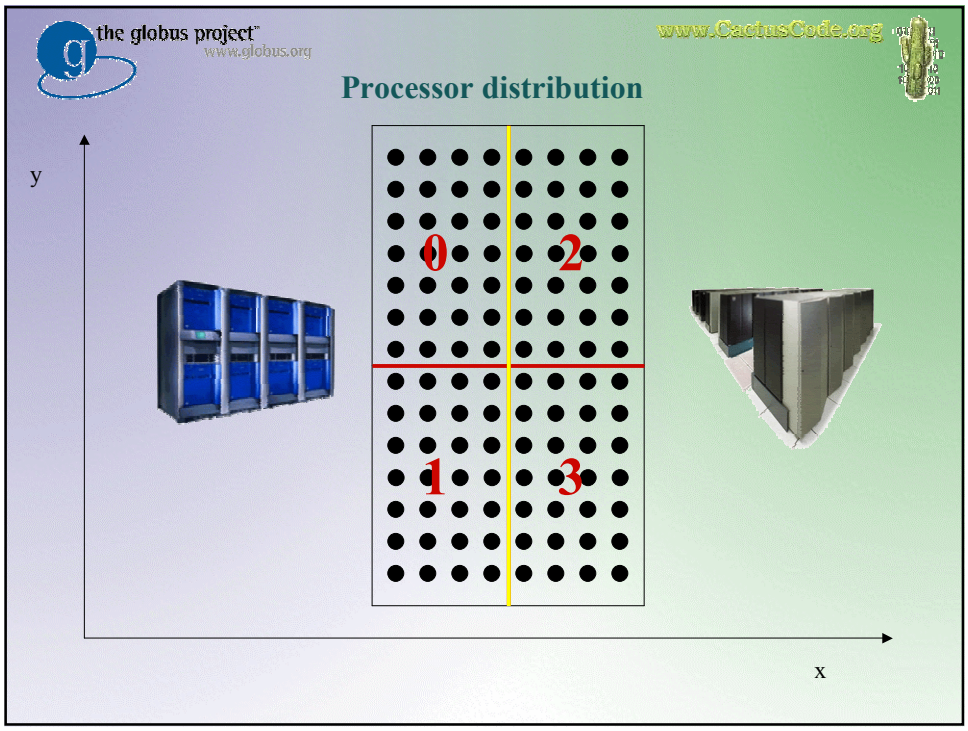
www.CactusCode.org

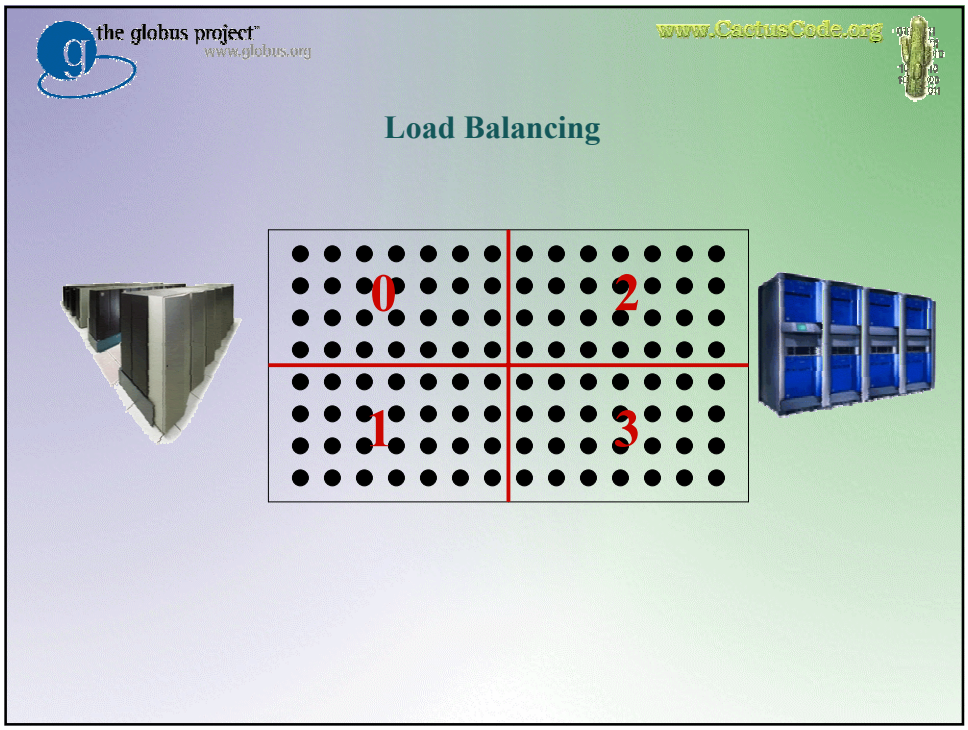
What we learnt from the DTF run

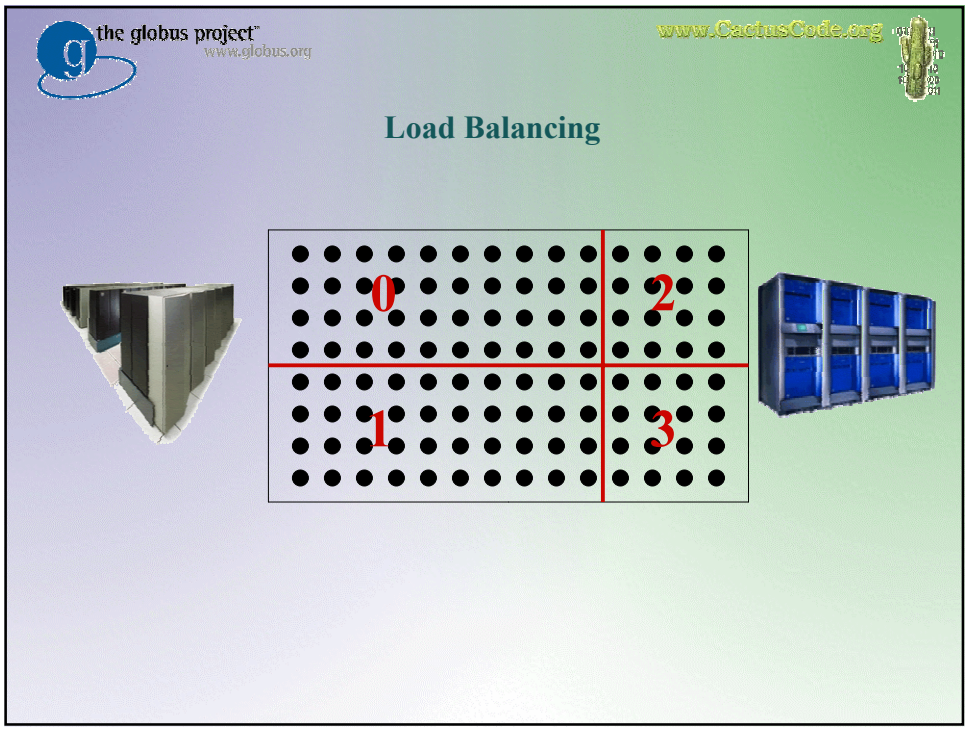
- Large scale distributed computing is possible with cactus,globus and mpich-g2
- Applying simple communication tricks improves efficiency a lot
- But: finding out best processor topology, where to compress, where to increase ghostsizes, how to loadbalance etc. goes **far beyond** what the user is willing to do
- configuration was not “fault-tolerant”
- Thus: we need a code which **automatically** and **dynamically** adapts itself to the given grid environment

And that’s what we have done


the globus project
www.globus.org
www.CactusCode.org
Processor distribution
y
0
2
1
3
x


the globus project
www.globus.org
www.CactusCode.org
Processor distribution
y
0
2
1
3
x


the globus project
www.globus.org
www.CactusCode.org
Load Balancing
0
2
1
3

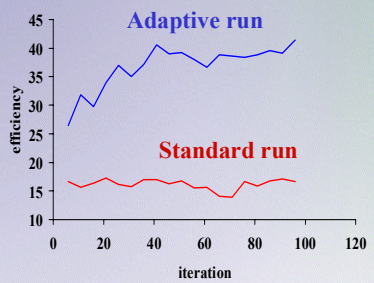

the globus project
www.globus.org
www.CactusCode.org
Load Balancing
0
2
1
3

the globus project
www.globus.org

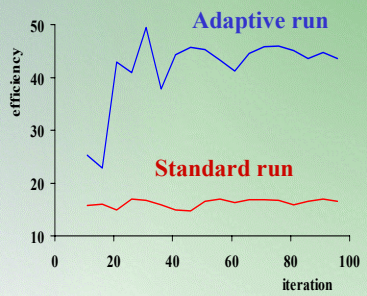
www.CactusCode.org

Adaptive Techniques

4+4 processor transatlantic run



8+8 processor NCSA+Washu



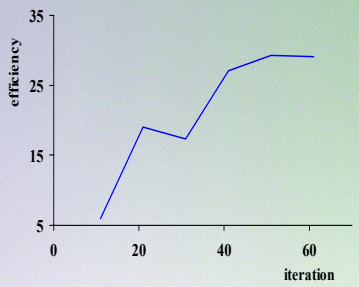
Runs here are "latest" physics-codes: many functions to synchronize , non-trivial data sets, non-communication-optimized algorithms on the application level

DTF run could be launched right away, with almost no preparation!

the globus project
www.globus.org

www.CactusCode.org

128+128 run btw. NCSA and SDSC yesterday



256 processors run, using **unoptimized** and **latest** fortran codes.

Launched from a portal & gained efficiency improvements of factor 6 out of the box!!

Improvements btw. April 2001 and now

- Processor distribution/topologies are set up in a way that communication over the WAN is always minimal
- Loadbalancing: fully automatic
- Ghostzones and compression: dynamically adaptive during the run, and only where needed
- To achieve all this, we consequently used globus (DUROC api)
- Now Fault-tolerant

Conclusion

- Executing codes in a metacomputing environment is becoming as easy as executing codes on a single machine with CACTUS, GLOBUS and MPICH-G2
- A much higher efficiency is **automatically** achieved during the run through **dynamical adaptation**
- incredible improvements between SC95 and now
- Together with the usage of portals and resource brokers the user will be able to take full advantage of the grid