THE UBC vn BEOWULF CLUSTER

Doc/VN/index.html

- CFI Proposal & Funding
- Construction
- Operation
- Future Plans
- Numerical Relativity
- Software Infrastructure for Parallel Computations

Matthew W. Choptuik, UBC & CIAR

Vancouver Linux User's Group Meeting

Burnaby BC, September 25, 2000

Supported by NSERC, CIAR, CFI and NSF PHY9722068

Standard prefix: laplace.physics.ubc.ca:/People/matt/

The Canadian Foundation For Innovation

www.innovation.ca

The CFI was established by the federal government with an up-front investment of $800 million. This principal amount and accrued interest will enable the Foundation to contribute, on average, about $180 million annually over five years to research infrastructure projects. The CFI targets its investment at key needs in the areas of health, environment, science, and engineering. The Foundation operates on the principle that its investments are made in partnership with the private and voluntary sectors as well as with provincial governments. The Foundation contributes 40% of total eligible project costs. On this basis, funding for the total investment by the Foundation and its partners should exceed $2 billion.

- Several programs, including *On-going New Opportunities*
  - Eligibility: First tenure track position in Canada.
  - HPC Potential at UBC from CFI: $\sim$ 750–1000 K / yr

The UBC vn PIII/Linux Cluster
Design: Take 1

- 280K CFI On-going New Opps. App., 4/29/99 (UBC)
  `Doc/CFI.april99/index.html`
  - Affleck (Phys. & Astro.)
  - Ascher (Comp. Sc.)
  - Choptuik* (Phys. & Astro.)
  - Patey* (Chem.)
  - Salcudean* (Mech. Eng.)
  - Thachuk* (Chem.)
  - Unruh (Phys. & Astro.)

- Patterned after Patey/Thachuk's machine (currently 23 compute nodes and one front-end, roughly half done), asks for
  - 64 $\times$ Dual 450 Mhz PIII/512 Mb/10 Gb (no CD ROM, keyboard, mouse, monitor) "compute nodes" 220K
  - 2 $\times$ Dual 450 Mhz PIII/512 Mb with additional peripherals "front-end nodes" 10K
  - 1 $\times$ HP-4000M Switch with 4 expansion modules $\rightarrow$ 72 (!) 100FDX ports (3.6 Gb/s back-plane) 7K
  - 13 (!) $\times$ APC Smart-UPS 1400 14K

The UBC vn PIII/Linux Cluster
Design: Take 2

- 650K CFI On-going New Opps. App., 9/15/99 (CFI) `Doc/CFI/index.html`
  - Affleck (Phys. & Astro.)
  - Ascher (Comp. Sc.)
  - Bushe* (Mech. Eng.)
  - Choptuik* (Phys. & Astro.)
  - Patey* (Chem.)
  - Salcudean* (Mech. Eng.)
  - Thachuk* (Chem.)
  - Unruh (Phys. & Astro.)
- ASKS FOR "Cluster 1" *AND*
- "Cluster 2" (focus on coarse-grained parallelism)
  - 48 $\times$ Single 600Mhz Alpha/2 Mb/256 Mb/10 Gb 230K
  - Myrinet (1000 Mb) Switch solution 32K
  - 8 $\times$ APC Smart-UPS 1400 9K
- CFI funding for full amount (40% of 650K) awarded in December 1999, matching BCKDF funding (40%) awarded June 2000, UBC funds remaining 20% from Bluman fund.

The UBC vn PIII/Linux Cluster Acquisition

- 280K for vn advanced against future CFI funding 8/27/99
- 9/99–10/99 spent evaluating machines, finding good location, setting up bid details with Purchasing
- Request for bid sent out 10/7 with closing date 11/2, equipment to be delivered 16 nodes per week
- Vendors: *Varsity*, *UBC Bookstore*, *AE*
- WHAT WE PURCHASED
  - 64 compute: 2 x 450Mhz PIII/512 Mb/10 Gb IDE 180k (Memory: 44% of total compute node cost)
  - 3 front-ends: 2 x 450Mhz PIII/512 Mb/34 Gb SCSI 20K
  - 1 $\times$ HP-4000M Switch: 7K
  - 4 $\times$ APC Matrix 3000M with 8 PDUs: 19K
- Total expenditures to date: 260K
- Computer room (Klinck/Old CS) annual "rent": 7K
  Currently split three ways: Klawe, Bushe, Choptuik

The UBC vn PIII/Linux Cluster
Assembly

- Assembly & Software Installation Team
  - Jason Ventrella
  - Inaki Olabarrieta
  - Choptuik
  - Unruh
- At vendor (near UBC)
  - BIOS settings
  - "Everything" (!) install of Mandrake 6.1
  - Network configuration including IP address assignment
- At our site (Main Machine Room, Old CS Bldg)
  - Plug node in, attach to network, power up
  - Secondary software installation (remote)

The UBC vn PIII/Linux Cluster
Additional Software

- MPI (Message Passing Interface) (`mpich` Version 1.1.2)
- PGI Compiler suite—gives us `F90` capability and yields significant performance improvement over `GNU` compilers for "complex" `f77` code (typically a factor or 2–2.5 or even more)
- Various public-domain scientific computation packages, `LAPACK`, `ODEPACK`, `HDF/netCDF`, $\cdots$

The UBC vn PIII/Linux Cluster
Hardware Problems

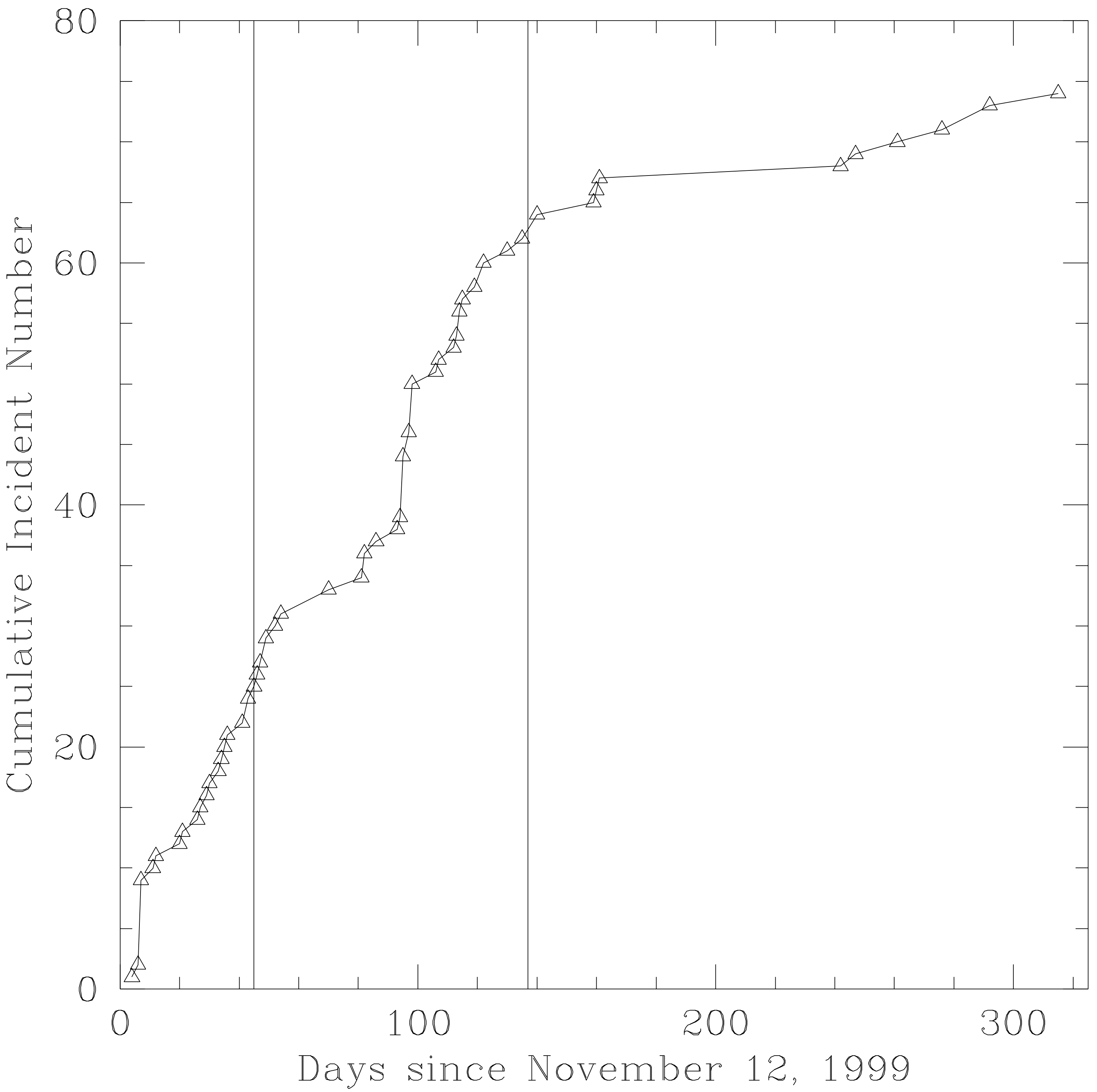
- Hardware shake-down, $\sim$ 6 problem nodes identified in first few weeks of operation: power supplies, bad memory, second processor not recognized
  - Power supplies
  - Bad memory modules
  - Second processor not recognized
- Re-seated one network card in March
- No other hardware problems

The UBC vn PIII/Linux Cluster
Software Problems

- Kernel issues
  - Driver for Intel network card required updating
    Rebuild of kernel required, completed Dec 24, 1999
  - `mv` of file on NFS mounted filesystem would cause node to hang. Rebuild, update of kernel required, completed Mar 20, 1999.
- Currently running 2.2.14 kernels

- About 75 total "incidents" / "crashes" thus far.
- Entire cluster has *never* been taken down.

vn.physics.ubc.ca: Incident (Crash) Summary



The UBC vn PIII/Linux Cluster

Sample Applications

- "shell-level" parallelism
  - Ethan Honda (UT Austin grad stud): detailed parameter space survey of "oscillons" (typically 40 + processes)
  - Roman Petryk (UBC grad stud): quantum gravity inspired calculations (typically 40 + processes)

- MPI-based parallelism
  - Luis Lehner (UT Austin postdoc), Mijan Huq (Penn State postdoc): 3D black hole calculations
    (81 x 81 x 81 spends 11% in communications. Could do 161 x 161 x 161)
  - Roman Baranowski, UBC Chemistry postdoc (MD Simulations)
  - Lothar Buchmann, TRIUMF Research Scientist (Nuclear Physics)

The UBC vn PIII/Linux Cluster

The anarchy queueing system

```
vnfe1 % ruptime | grep -v down | grep -v vnfe | sort -n +6

vn10 up 9+11:34, 0 users, load 0.00, 0.00, 0.00
vn11 up 9+11:34, 0 users, load 0.00, 0.00, 0.00
vn13 up 9+11:34, 0 users, load 0.00, 0.00, 0.00
vn15 up 9+11:31, 0 users, load 0.00, 0.00, 0.00
vn20 up 9+11:32, 0 users, load 0.00, 0.00, 0.00
vn21 up 9+11:32, 0 users, load 0.00, 0.00, 0.00
vn22 up 9+11:32, 0 users, load 0.00, 0.00, 0.00
vn23 up 9+11:28, 0 users, load 0.00, 0.00, 0.00
vn24 up 9+11:28, 0 users, load 0.00, 0.00, 0.00
vn26 up 9+11:29, 0 users, load 0.00, 0.00, 0.00
vn35 up 9+11:27, 0 users, load 0.00, 0.00, 0.00
vn39 up 9+11:27, 0 users, load 0.00, 0.00, 0.00
vn40 up 9+11:28, 0 users, load 0.00, 0.00, 0.00
vn41 up 9+11:28, 0 users, load 0.00, 0.00, 0.00
vn42 up 9+11:28, 0 users, load 0.00, 0.00, 0.00
vn43 up 4+01:51, 0 users, load 0.00, 0.00, 0.00
vn44 up 4+22:16, 0 users, load 0.00, 0.00, 0.00
vn8 up 9+11:34, 0 users, load 0.00, 0.00, 0.00
vn9 up 9+11:34, 0 users, load 0.00, 0.00, 0.00
vn33 up 9+11:26, 0 users, load 0.97, 0.91, 0.82
vn38 up 8+17:48, 0 users, load 1.82, 1.91, 1.89
                         .
                         .
                         .
vn53 up 4+21:31, 0 users, load 2.27, 2.20, 2.08
```

The UBC vn PIII/Linux Cluster

Usage Summary

- Usage from 01/01/2000 through 01/19/2000
  - 440 000 CPU hours of possible 800 000 "Raw" usage 55%
  - Figure closer to 75% over the past two months—cluster typically fully saturated during peak hours.
- Selected user amounts

USER	CPU Hrs Used	% of Tot. Avail.
Roman Baranowski, UBC Chem	98 000	12%
Suqin Dong, UBC Mech Eng	52 000	6.5%
Christopher Daub, UBC Chem	49 000	6.1%
Trevor Stocki, TRIUMF	37 000	4.7%
Lothar Buchmann, TRIUMF	36 000	4.5%
Jordana Tzenova, UBC Zoology	28 000	3.5%
Ethan Honda, UBC/UT Austin Phys	20 000	2.4%
⋮		
Matthew Case, SFU Phys	10 000	1.2%
⋮		

- 23 users with usage over 1000 CPU-hours

FUTURE PLANS
"Phase 2 Cluster"

- Second cluster from remaining CFI/BCKDF/UBC funding ($\sim$ 300K)
  - Projected installation: Before March 2001
  - 48 $\times$ Single 600Mhz Alpha/2 Mb/256 Mb/10 Gb
  - 1000 Mb interconnect, low latency

"Phase 1 Upgrade"

- Can upgrade motherboard to 800 Mhz:
  - 800 Mhz, \$310 per processor, \$400 per 450-equiv, 0.78 times more cluster for \$43K (0.16 of original cost)
  - 700 Mhz, \$220 per processor, \$400 per 450-equiv, 0.56 times more cluster for \$29K (0.11 of original cost)
  - 550 Mhz, \$110 per processor, \$500 per 450-equiv, 0.22 times more cluster for \$15K (0.058 of original cost)

FUTURE PLANS

“Phase 3 Cluster, ⋯”

Doc/CFI00/

- Plan to submit BC-wide proposal for next CFI Innovation Fund competition, CFI deadline sometime next spring?
- Will involve researchers at UBC, SFU, U Vic, TRIUMF among others.
- Preliminary budget 10 000 000 total
- Rough specs:
  - 1 TeraFlop (1 000 000 MegaFlop) computation rate
  - 1 Terabyte RAM
  - 500 Terabyte Disk
- Would have at least 100 times capacity of current cluster.
- Possible participation/partnering with one of NSF supercomputing centres?

Numerical Relativity
Goals

Simulation of geometry of space-time without and with sources
Simulation of the gravitational field without and with sources

- Astrophysically relevant, dynamical, gravitational-radiation-producing spacetimes of particular interest,
  Must solve field equations in 3 space-dimensions plus time

- Physical Requirements for Efficient Radiation
  - (Large) masses confined to regions comparable in size to their Schwarzschild radii, R_S:

$$R_S = \frac{2G}{c^2} M$$

$$\frac{2G}{c^2} = 1.5 \times 10^{-27} \, \frac{\mathrm{m}}{\mathrm{kg}} = 3.0 \, \frac{\mathrm{km}}{M_\odot}$$

$$G = 6.67 \times 10^{-11} \mathrm{N\ m^2/kg^2} \qquad c = 3.00 \times 10^8 \mathrm{m/s}$$

    R_S for Earth is about 1 cm!

  - Internal redistribution of significant fraction of energy at speeds approaching speed of light, c

LIGO Site 1: Hanford WA

(http://www.ligo-wa.caltech.edu/)

LIGO Site 2: Livingston LA

(http://www.ligo-la.caltech.edu/)

Numerical Relativity
Goals

- Ideal Candidates—"Compact Binaries"
  - Black hole–black hole binary (for BH, $R = R_S$)
  - Black hole–neutron star binary
  - Neutron star–neutron star binary

- Not-so-astrophysically relevant but physically motivated model problems also of interest, focus of my past research
  - No experimental GR
  - Possibility for "computational laboratories"
  - Good vehicle for infrastructure & algorithm development

Typical Model Problem

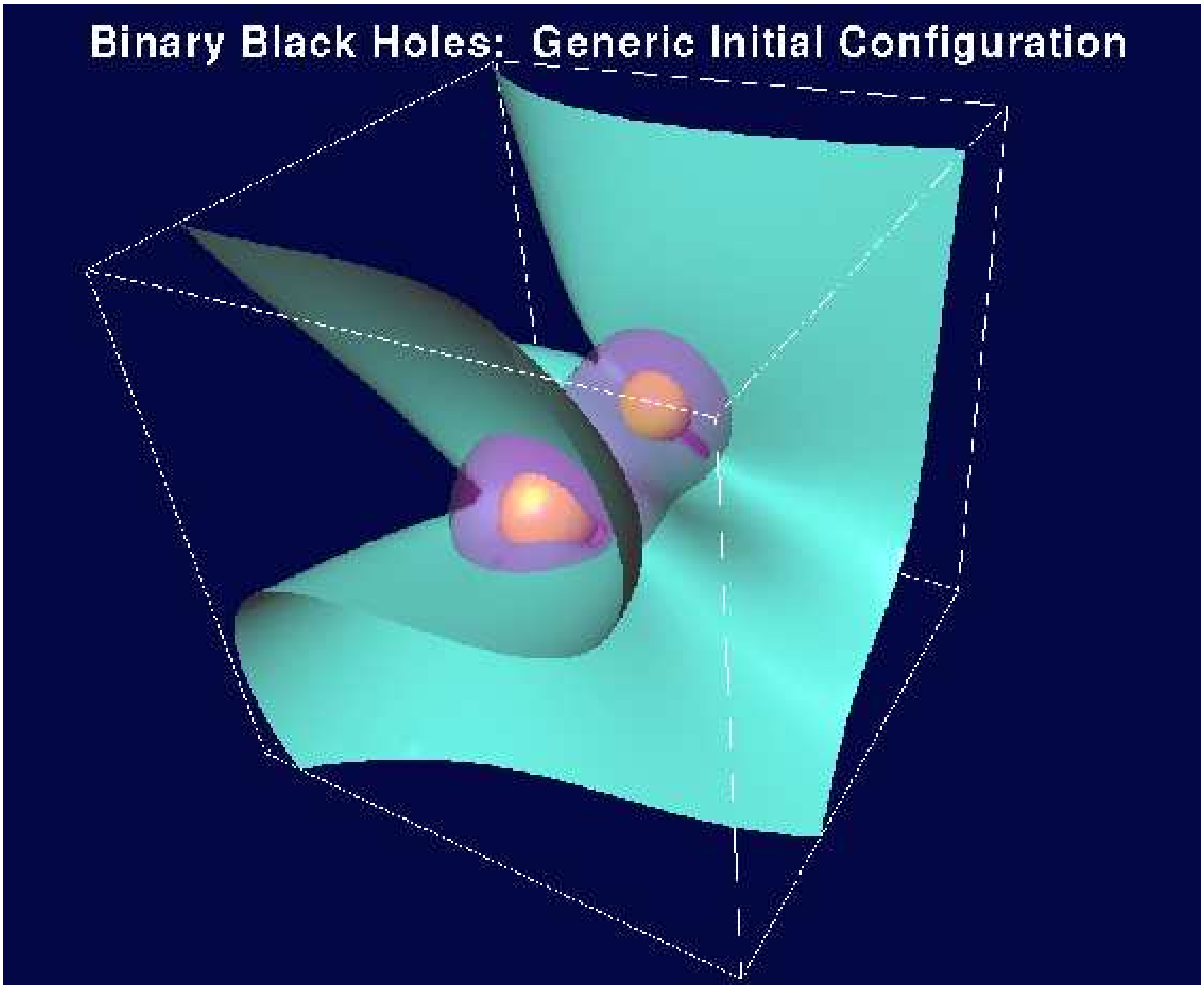
- Reduced spatial dimensionality (spherical, $1+1$, axisymmetric, $2+1$)
- "Simple" matter: typically scalar field instead of perfect fluid
- Key non-linear features retained (e.g. black hole formation)

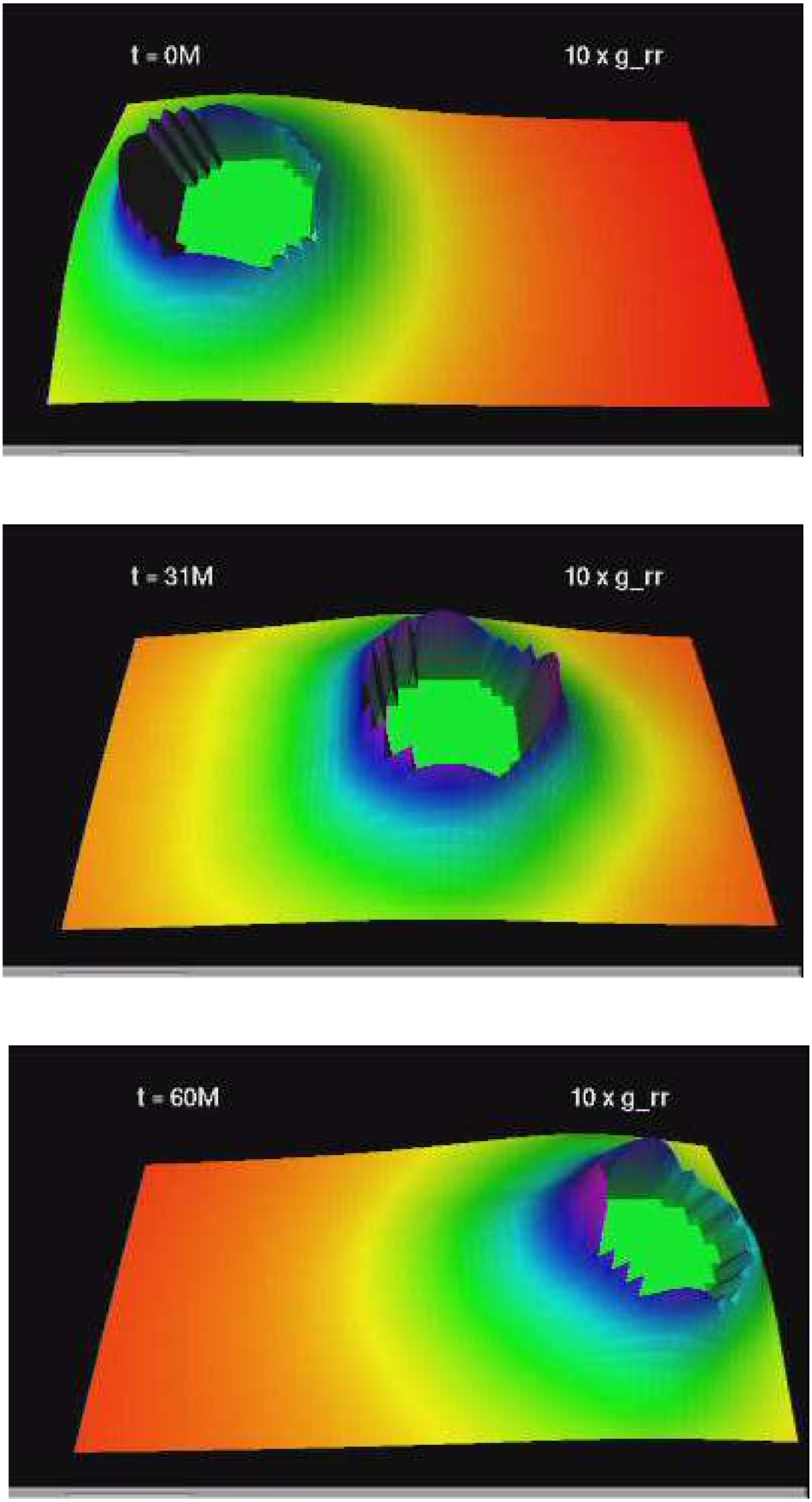
Numerical Relativity
Challenges

- Large computational requirements
  - Back-of-the-envelope estimate for single 2 BH collision: 1 CPU week on 1 Tflop/s system
- Physical interpretation of results (incl. visualization)
  - Large number of dynamical variables
  - Dynamical vbls tend to be tensor components, so so often have no intrinsic physical interpretation *per se*
  - No "lab" for intuition
- Coordinate Freedom
  - Prescription for coordinatization of space-time must be given, can not assume to be known *a prioir*, as in non-general-relativistic dynamics.
  - Bad prescription of coordinates can (and often does!) lead to encounters with physical or coordinate singularities.
- Singularity Avoidance
  - BH space-times generically contain physical singularities; must be avoided or dealt with in a special fashion
- STABILITY (Convergence)

Visualization of Initial Data for 2 Black Holes

(Cook *et al*, *Phys. Rev. D*, 1993)




t = 0M
10 x g_rr
t = 31M
10 x g_rr
t = 60M
10 x g_rr

Infrastructure for (Adaptive) Parallel Computations
Motivation & Goals

- Observation: Numerical relativity codes have tended to be remarkably homogeneous from a "high-level" point of view: Almost all have employed low order (second-order) finite difference techniques on a single mesh, and have had the following structure:

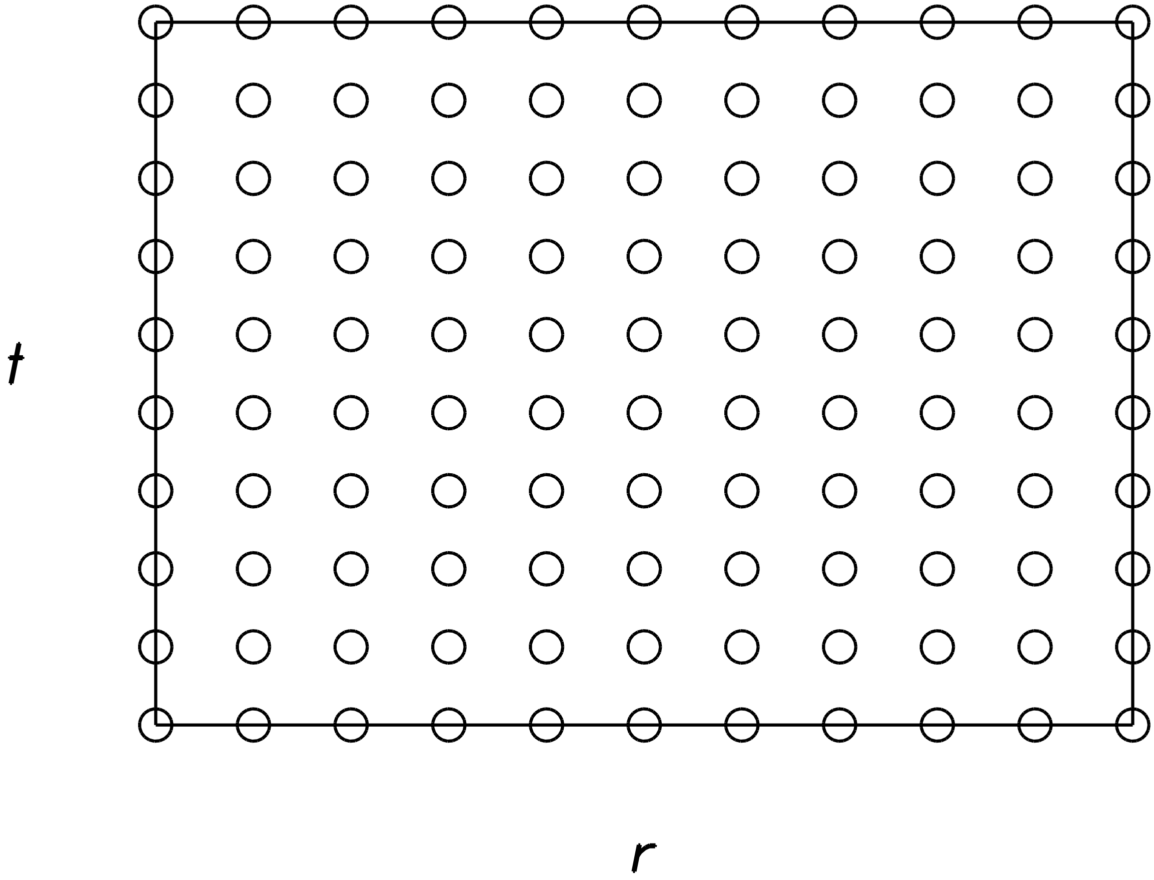
```
Read (initial) state
for NUM_STEPS
  for NUM_UPDATES & maybe until convergence
    U (Grid Function(s)) -> Grid Function(s)
  end for
end for
Write (final) state
```

- Most of the hard work in developing a new code involves the construction of stable, accurate updates, `U`

- Also clear that significant dynamic range in black-hole problems such as binary coalescence means that adaptive-mesh-refinement (AMR) algorithms essential for efficient computation

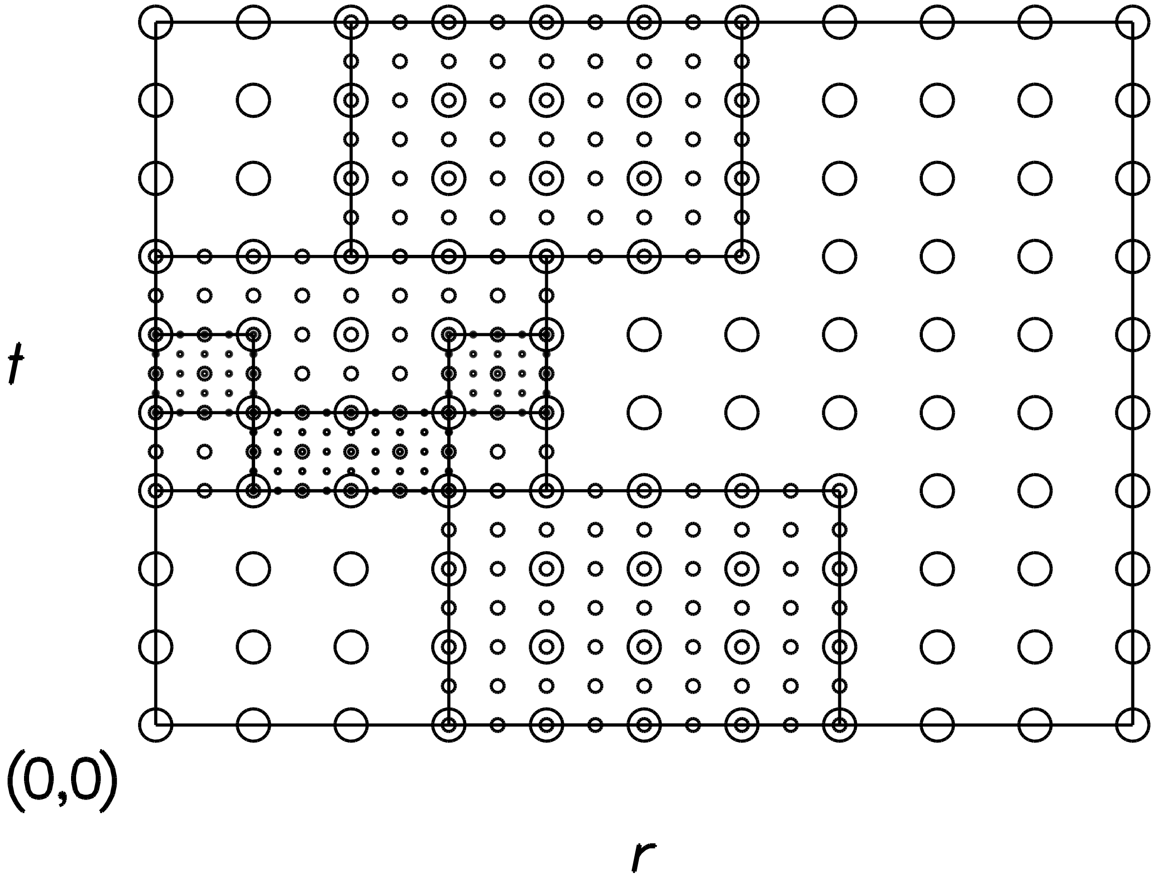
- Ultimate goal: allow relativist to concentrate on developing stable, uni-grid code on serial architecture: parallelism and adaptivity to be "automatically" provided by the infrastructure

Uniform Finite Difference Mesh



Schematic Adaptive-Mesh Structure

2 : 1 Refinement in Space and Time



Infrastructure for Adaptive Parallel Computations
DAGH / GrACE

Manish Parashar (Rutgers) & J.C. Browne (UT Austin)
http://www.caip.rutgers.edu/ parashar/TASSL/

- Two main components
  - A set of programming abstractions in which computations on dynamic hierarchical grid structures are directly implementable.
  - A set of distributed dynamic data-structures that support the implementation of the of the abstractions in parallel execution environments and preserve efficient execution while providing transparent distribution of the grid hierarchy across processing elements.
- Key Features
  - Transparent access to scalable distributed dynamic Arrays, Grids, Grid-Hierarchies
  - Shadow grid-hierarchy for efficient error estimation (regridding criterion)
  - Automatic dynamic partitioning and load distribution
  - Locality in face of mutli-level data (space-filling curves).
  - Some special support for multi-grid

Infrastructure for Adaptive Parallel Computations

DAGH / GrACE 2-D Wave Example (Schematic)

```
#include "GrACE.h"
#include "GrACEIO.h"

bb[0]=xmin; bb[1]=xmax; bb[2]=ymin; bb[3]=ymax;
shape[0]=Nx; shape[1]=Ny;

GridHierarchy GH(2,NON_CELL_CENTERED,1);
GH.ACE_SetBaseGrid(bb, shape);
GH.ACE_ComposeHierarchy();
GH.ACE_IOType(ACEIO_HDF_RNPL);

BEGIN_COMPUTE

GridFunction(2)<double> phi("phi",1,1,GH,ACEComm,ACENoShadow);

for( step++; step <= nsteps; step++ ){
  forall(phi,tc,lev,c)
    update( ··· )
  end_forall
  phi.GF_Sync(tc+idt,lev,ACE_Main);
}
```

Infrastructure for Adaptive Parallel Computations

CACTUS / PUGH

Paul Walker et al (MPI Potsdam)

http://www.cactuscode.org/

- Includes PUGH package, which implements DAGH-style memory distribution/parallelization, but in a more compact C library, and only for uni-grid applications.

- Provides users of CACTUS with automatic access to parallelism.

- Code runs on essentially anything, and routinely is near or at the record for highest-sustained Gigafloppage on "realistic" problem: From http://www.ncsa.uiuc.edu/access.html

  "In June [99], the team virtually owned NCSA's 256-processor Origin2000 for a capability computing run of more than two weeks. By the time Suen and Seidel had finished their simulations, they had output nearly a terabyte of data and logged an astonishing 140,000 CPU-hours on the Origin2000."

- Significant level of support from MPI Potsdam and NSCA