The Rice Performance Tools

Rob Fowler

John Mellor-Crummey,

Monika Mevenkamp, Nathan Tallent, Gabriel Marin,
Xuesong Yuan, Jason Eckhard, Scott Warren, Lynn Wang

Center for High Performance Software
Rice University

Outline

- Background: Why we're building performance tools.
- Goals
- Approach
- Success Stories, Status, and Future.
- Examples.

Background

What we (Rice Parallel Compilers Group) do –

- Code optimization
  - Aggressive (mostly source-to-source) compilation.
  - Hand application of optimizing transformations.
    - Try out transformations by hand first.
    - Work on real codes with algorithm, library, and application developers.

We spend a lot of time (too much?) analyzing executions. Why?

1. Deeply pipelined, out of order, superscalar processors with non-blocking caches and deep memory hierarchies.
2. Aggressive (-O4) and/or idiosyncratic vendor compilers.

What we did –-

- Built tools to meet our needs w.r.t. the run/analyze/tune cycle.
- They were so useful that we are now distributing (one-on-one) them
  - In use at DoD, DoE, and NSF supercomputer facilities

Why I'm here --

- To present the tools
- To encourage collaborative work.

Problems with Existing Performance Tools

Most tools are hard to use on big, complex, modular applications. Why?

Despite 25 years of experience, profilers, etc are dramatically underused.

Tools (feel like they) are designed to evaluate architectural features, or operating systems, rather than for application tuning.

Tools have insufficient analytic power.

- Any one performance metric produces myopic view.
  - » Some metrics are causes. Some are effects. Comparisons required!

    Ex. – misses, FLOPS, loads, mispredicts, cycles, stall cycles.
  - » Instruction balances (loads vs FLOPS vs integer ops vs branches).
  - » Fusion of compiler analysis, simulation, and counter profiles.
- Labor intensive analytic methodology is difficult and/or tedious.
  - » (Re-)inserting explicit instrumentation and recompiling.
  - » Manual correlation of data from multiple sources with source code.
  - » Aggregating per line data into larger scopes.
  - » Computing synthetic measures, e.g. loads/flop.
- Tune/reanalyze cycle slowed or prevented by manual intervention and analysis.

More Problems

Language- or compiler-based tools restrict domain of applicability

- Multi-language, multi-module (library.so) applications?
- Cross-platform, cross-compiler comparisons?

User Interface Issues

- GUI problems
  - » Vendor specific, both for target and analysis machines.
  - » Non-portable, non-collaborative visualization.
  - » Single-metric displays don't capture underlying problems
  - » Need for block, loop-level, and user-defined scopes.
- Failure to make compelling cases.

  Hard to convince application developers and/or management with a fat stack of printouts and hand-generated notes.

Goal: Build Tools that We Will Use

- Intuitive, top-down user interface for performance analysis.
- Provide information needed for analysis and tuning.
- Platform and language (compiler) independence
- Eliminate manual labor from the analyze, tune, run cycle.

Approach

- Intuitive, top-down user interface for performance analysis.
  - Use familiar hypertext browsing models.
- Provide information needed for analysis and tuning.
  - Automatically compute derived performance metrics.
  - Assimilate and combine data from multiple sources.
    - » Examples: FLOPS/cycle, miss ratios, (actual cycles – ideal), static analyses.
    - » Filters and XMLwriter convert any "profile" into a standard format.
- Platform and language (compiler) independence
  - Multiple data sources → Cross Platform Comparisons
  - Extract hierarchical program structure from (-g3) binaries.
    - » Handle multiple languages: F77, F9x, C, C++, …
    - » Use compiler-generated symbol table and object → source maps.
    - » Permits analysis of libraries, including dynamically loaded.
- Eliminate manual labor from the analyze, tune, run cycle.
  - Computation of derived metrics.
  - Drive the process with scripts and configuration files.

The components of the HPCView Toolbox

hpcview → creates performance database from sources, profiles, structure information.

Netscape, Internet Explorer → initial user interface (static data)

hpcviewer → Java-based viewer

bloop → extracts structure from binaries

» Based on EEL, distribution restricted.

open_analysis → extracts structure from binaries

» Uses Open64 infrastructure, Rice analysis → GPLable

ptran, ProfileWriter → convert/write data in standard format

hpcprof → cprof for Linux that generates XML files directly

» Cprof output not intended for down stream analysis

xprof → prof extended to handle code replication

» Converts DCPI/ProfileMe output

» Better attribution for templates, includes, compiler replicated code

msgprof → Compatible profiling of library calls that can block, e.g. MPI.

Scripts and makefiles → glue necessary for automation

Status

hpcview → v2.01 “in production”, v2.1 in alpha test

Netscape, Internet Explorer → available everywhere

hpcviewer →v1.0 in alpha test, Java portability issues surprise us!

bloop → distributed to Gov. Labs.

open_analysis → first use this week

ptran, ProfileWriter → production: ssrun, uprofile, static analyses

hpcprof → alpha test on P2 and P3 boxes. Depends on PAPI, etc.

xprof → works for DCPI/ProfileMe, but consumer tools not started

msgprof → being used in GrADS project. Distributable some day.

Scripts and makefiles → an expanding set

Near Future

hpcview → v2.01 "in production", v2.1 in alpha test (release in weeks)

Netscape, Internet Explorer → available everywhere

hpcviewer →v1.0 in alpha test, Java portability issues surprise us!

(release in weeks to friendly users for use on Windows)

bloop → distributed to Gov. Labs. (also to EEL licence holders)

open_analysis → first use this week (release in May)

ptran, ProfileWriter → production: ssrun, uprofile, static analyses

hpcprof → alpha test on P2 and P3 boxes. Depends on PAPI, etc.

(will release with v2.1 of hpcview)

xprof → works for DCPI/ProfileMe, but consumer tools not started

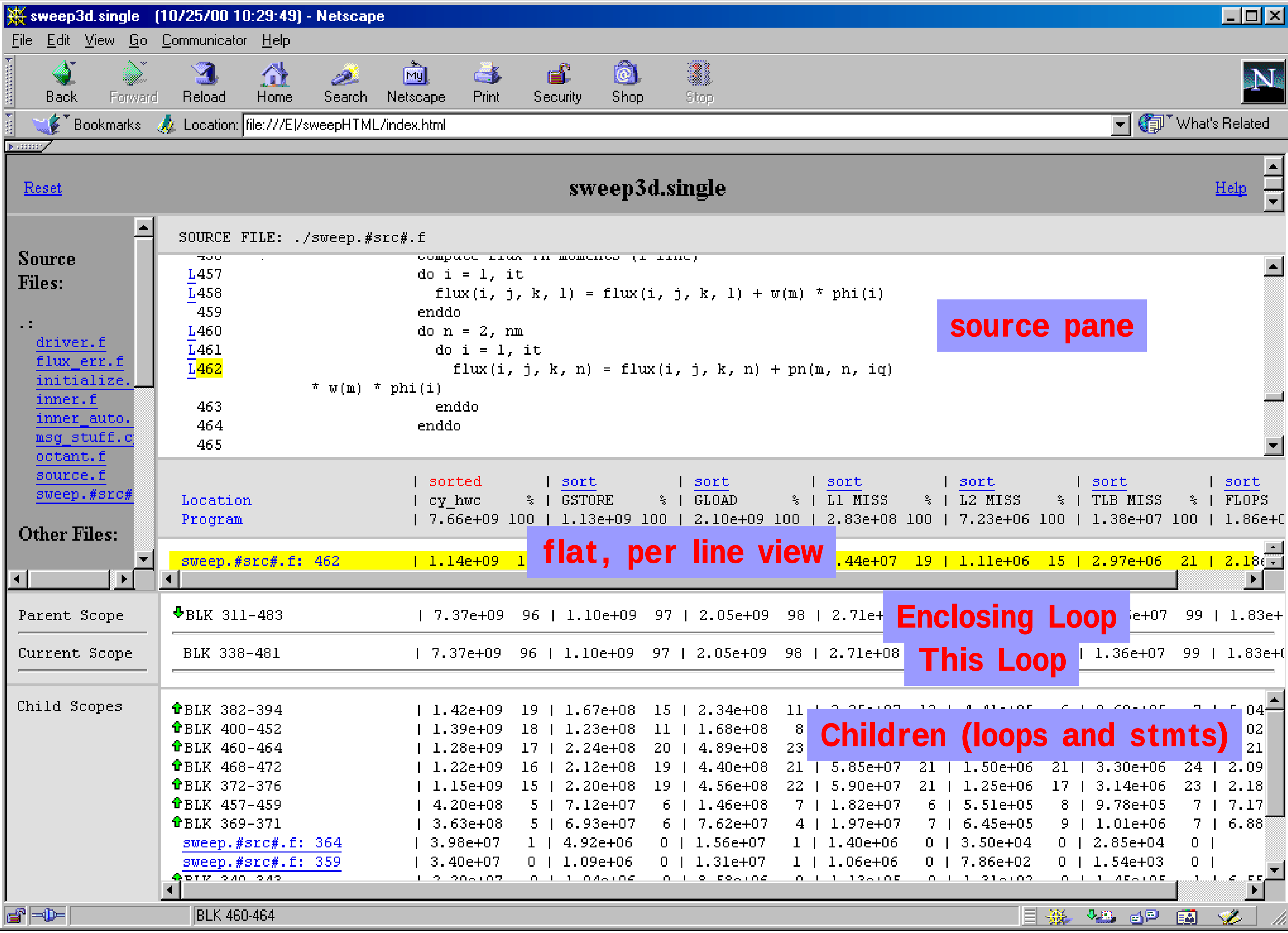
(looking for a student)

Scripts and makefiles → an expanding set

Successes and ???

- The tool suite is in daily use at Rice.
  - NCOMMAS project, among others.
- Successful ongoing use at Sandia and LANL.

  These have been the results of “hands-on deployment”.
  - ASCI codes and “compact applications”
  - Proprietary codes.
  - Science applications: Ocean Modelling, Quantum Chemistry.
- Version(s) have been made available at NCSA, but haven’t caught on.


sweep3d.single (10/25/00 10:29:49) - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: file:///E|/sweepHTML/index.html
What's Related
Reset
sweep3d.single
Help
Source Files:
.:
driver.f
flux_err.f
initialize.
inner.f
inner_auto.
msg_stuff.c
octant.f
source.f
sweep.#src#
Other Files:
SOURCE FILE: ./sweep.#src#.f
L457 do i = 1, it
L458 flux(i, j, k, l) = flux(i, j, k, l) + w(m) * phi(i)
459 enddo
L460 do n = 2, nm
L461 do i = 1, it
L462 flux(i, j, k, n) = flux(i, j, k, n) + pn(m, n, iq)
* w(m) * phi(i)
463 enddo
464 enddo
465
source pane
Location | sorted cy_hwc % | sort GSTORE % | sort GLOAD % | sort L1 MISS % | sort L2 MISS % | sort TLB MISS % | sort FLOPS
Program | 7.66e+09 100 | 1.13e+09 100 | 2.10e+09 100 | 2.83e+08 100 | 7.23e+06 100 | 1.38e+07 100
sweep.#src#.f: 462 | 1.14e+09
flat, per line view
Parent Scope
BLK 311-483 | 7.37e+09 96 | 1.10e+09 97 | 2.05e+09 98
Enclosing Loop
Current Scope
BLK 338-481 | 7.37e+09 96 | 1.10e+09 97 | 2.05e+09 98 | 2.71e+08 | 1.36e+07 99
This Loop
Child Scopes
BLK 382-394 | 1.42e+09 19 | 1.67e+08 15 | 2.34e+08 11
BLK 400-452 | 1.39e+09 18 | 1.23e+08 11 | 1.68e+08 8
BLK 460-464 | 1.28e+09 17 | 2.24e+08 20 | 4.89e+08 23
BLK 468-472 | 1.22e+09 16 | 2.12e+08 19 | 4.40e+08 21 | 5.85e+07 21 | 1.50e+06 21 | 3.30e+06 24
BLK 372-376 | 1.15e+09 15 | 2.20e+08 19 | 4.56e+08 22 | 5.90e+07 21 | 1.25e+06 17 | 3.14e+06 23
BLK 457-459 | 4.20e+08 5 | 7.12e+07 6 | 1.46e+08 7 | 1.82e+07 6 | 5.51e+05 8 | 9.78e+05 7
BLK 369-371 | 3.63e+08 5 | 6.93e+07 6 | 7.62e+07 4 | 1.97e+07 7 | 6.45e+05 9 | 1.01e+06 7
sweep.#src#.f: 364 | 3.98e+07 1 | 4.92e+06 0 | 1.56e+07 1 | 1.40e+06 0 | 3.50e+04 0 | 2.85e+04 0
sweep.#src#.f: 359 | 3.40e+07 0 | 1.09e+06 0 | 1.31e+07 1 | 1.06e+06 0 | 7.86e+02 0 | 1.54e+03 0
Children (loops and stmts)
BLK 460-464

POP 4-way shmem, model_size=medium (11/05/00 18:56:06) - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: file:///E|/popHPCView/index.html What's Related

Reset

POP 4-way shmem, model_size=medium

Help

Source Files:

./compile:
- POP.f
- advection.f
- baroclinic.f
- barotropic.f
- boundary.f
- communicate.f
- constants.f
- convection.f
- diagnostics.f
- expansion.f
- forcing.f
- forcing_ap.f
- forcing_coupled.f
- forcing_pt_interior.f
- forcing_s_interior.f
- forcing_sfwf.f
- forcing_shf.f
- forcing_tools.f
- forcing_ws.f
- global_reductions.f

SOURCE FILE: ./compile/stencils.f

```
 1272    !
 1273    !-----------------------------------------------------------------------
 1274
L1275          do j=2,jmt-1
L1276            do i=2,imt-1
L1277              XOUT(i,j)    = CC(i,j)*X(i,j) +
 1278         &                   CN(i,j)*X(i   ,j+1) + CS(i,j)*X(i   ,j-1) +
 1279         &                   CE(i,j)*X(i+1,j   ) + CW(i,j)*X(i-1,j   )
L1280            end do
L1281          end do
 1282
 1283    !-----------------------------------------------------------------------
 1284    !
 1285    !     boundary updates if required
```

Location	sorted Cycles	%	sort L1 miss	%	sort L2 miss	%	sort FP insts	%	sort cy per F	%
Program	1.88e+11	100	9.44e+09	100	6.84e+08	100	2.65e+10	100	7.09e+00	100
stencils.f: 1277	1.01e+10	5	9.47e+08	10	4.10e+07	6	2.00e+09	8	5.07e+00	71
vertical_mix.f: 605	4.83e+09	3	1.84e+07	0	6.15e+06	1	6.22e+08	2	7.77e+00	110
vertical_mix.f: 633	4.70e+09	2	5.76e+07	1	8.48e+06	1	1.12e+08	0	4.20e+01	592
state_mod.f: 181	4.64e+09	2	2.11e+08	2	1.46e+07	2	2.17e+09	8	2.14e+00	30

Parent Scope

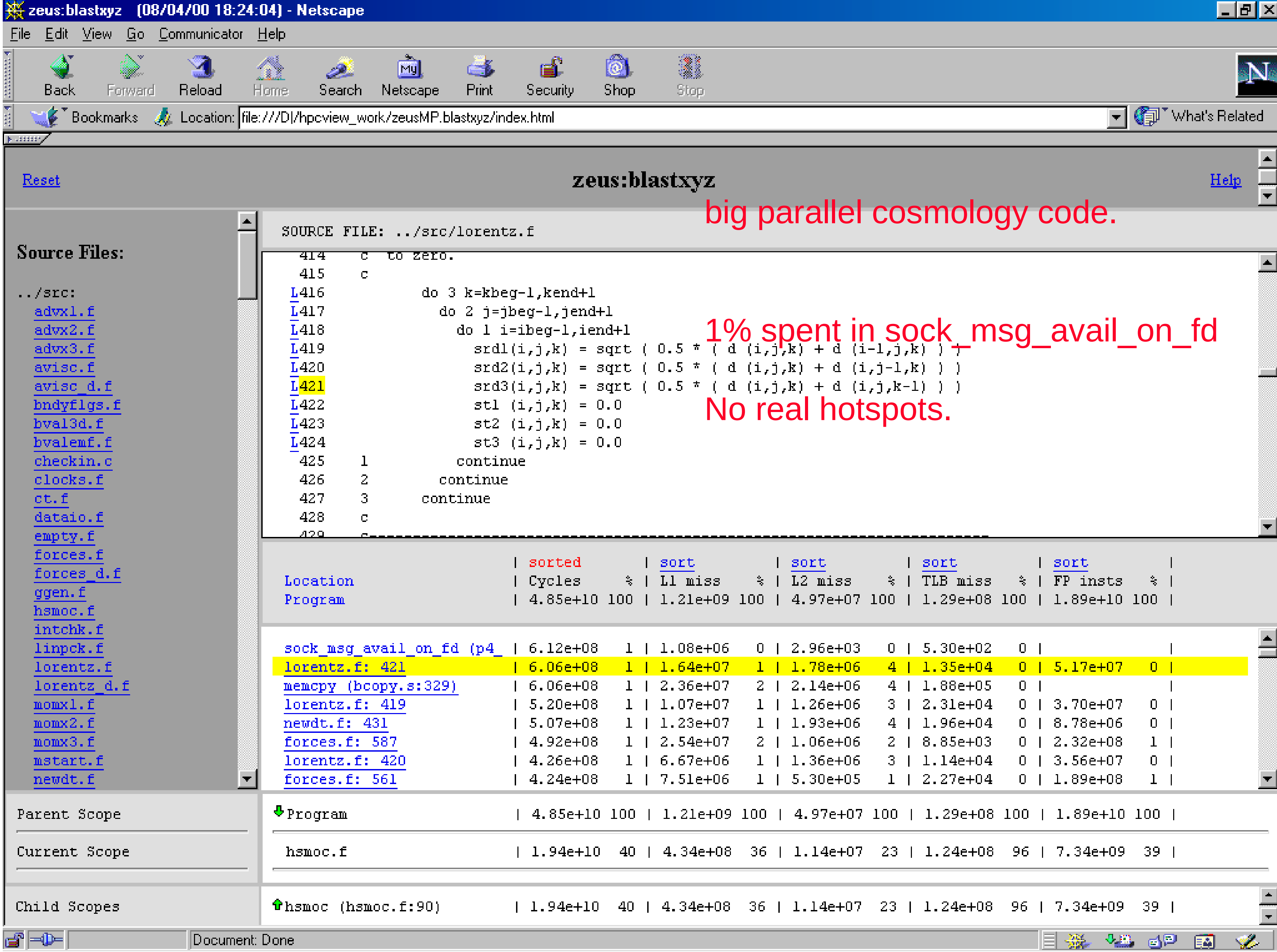
Current Scope

Program	1.88e+11	100	9.44e+09	100	6.84e+08	100	2.65e+10	100	7.09e+00	100

Child Scopes

stencils.f	3.85e+10	20	2.74e+09	29	1.51e+08	22	7.85e+09	30	4.91e+00	69
vertical_mix.f	3.50e+10	19	8.38e+08	9	9.13e+07	13	3.55e+09	13	9.87e+00	139
baroclinic.f	3.05e+10	16	1.58e+09	17	1.23e+08	18	3.42e+09	13	8.92e+00	126

Document: Done



big parallel cosmology code.

1% spent in sock_msg_avail_on_fd

No real hotspots.

bt_danich (12/15/00 13:49:54) - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: file:///D|/hpcview_work/NAS-BT-dHPF/index.html What's Related

Reset **bt_danich** Help

Comparison of the number of graduated loads on a MIPS R10K vs a R12K on the same input and binary.

Source Files:

.:

- add.f
- error.f
- exact_rhs.f
- exact_solution.f
- extra.f
- foo.f
- initialize.f
- lhsx.f
- lhsy.f
- lhsz.f
- rhs.f
- x_solve.f
- y_solve.f
- z_solve.f

Other Files:

- BufferHashTable.C
- Buffers.C
- HPFrtd.C
- HashTable.C
- LogicalAssertion.f
- LongMap.C
- LongMap.h
- LongMapTmpl.h

SOURCE FILE: ./x_solve.f

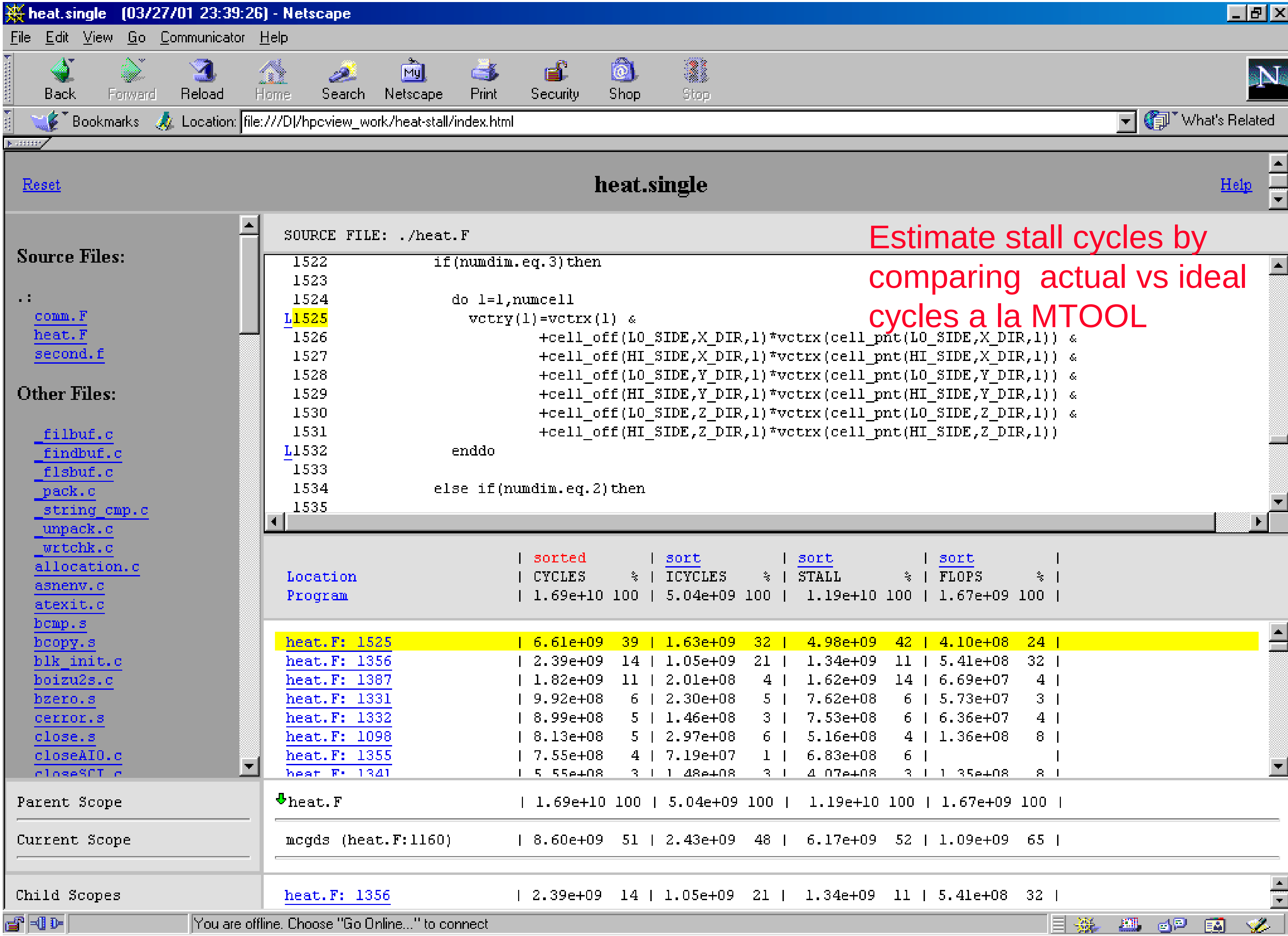
```
 2125         *yidl, j - gridx_myid2, k - gridx_myid3, tile))
 2126    CHPF$               endon
 2127                        ifwd = 1
L2128                        do l = 1, 5
 2129                          do m = 1, 5
L2130                            tmp_lhs(m, l) = lhs(m, l, cc, i - gridx_myidl -
 2131         *l, j - gridx_myid2, k - gridx_myid3, tile)
 2132                          enddo
 2133                        enddo
 2134                        ifwd = 1
 2135    CHPF$               on home(lhs(1, 1, aa, i, j, k)) begin
L2136                        call matmul_sub(lhs(1, 1, aa, i - gridx_myidl, j - g
 2137         *ridx_myid2, k - gridx_myid3, tile), tmp_lhs(1, 1), lhs(1, 1, bb, i
 2138         * - gridx_myidl, j - gridx_myid2, k - gridx_myid3, tile))
L2139                        call binvcrhs(lhs(1, 1, bb, i - gridx_myidl, j - gri
```

	sorted		sort		sort	
Location	glmapy	%	glnirv	%	gldiff	%
Program	9.94e+09	100	9.63e+09	100	-3.03e+08	100
extra.f: 72	5.29e+08	5	5.08e+08	5	-2.11e+07	7
extra.f: 21	4.84e+08	5	5.92e+08	6	1.08e+08	-35
extra.f: 62	3.14e+08	3	2.84e+08	3	-3.03e+07	10
extra.f: 57	2.88e+08	3	1.94e+08	2	-9.33e+07	31
extra.f: 67	2.58e+08	3	2.90e+08	3	3.17e+07	-9
x_solve.f: 2130	2.36e+08	2	2.16e+08	2	-1.97e+07	7
exact_solution.f: 19	2.35e+08	2	2.39e+08	2	4.03e+06	0
x_solve.f: 1053	2.28e+08	2	2.28e+08	2	-1.40e+04	0

Scope						
Parent Scope						
Current Scope: Program	9.94e+09	100	9.63e+09	100	-3.03e+08	100
Child Scopes: extra.f	4.14e+09	42	4.06e+09	42	-8.39e+07	28

extra.f: 67

Estimate stall cycles by comparing actual vs ideal cycles a la MTOOL



sweep3d alpha cyclesEV6 vs EV67ls [04/14/01 18:31:07] - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: file:///D|/hpcview_work/Comp6_67/index.html What's Related

Reset

sweep3d alpha cyclesEV6 vs EV67ls

Help

Source Files:

/home/rjf/HARD_CODES/sweep3d

- flux_err.f
- initialize.f
- inner.f
- source.f
- sweep.f

SOURCE FILE: /home/rjf/HARD_CODES/sweep3d_alphas/vl/sweep.f

```
L427            phi(i+10) = phi_i10
L428            phi(i+11) = phi_i11
 429         end do
L430      do i = 1 + ((it)/12)*12 , it
L431            phi_i = src(i,1,j,k)
L432            do n = 2, nm
L433               phi_i = phi_i + pn(n,m,iq)*src(i,n,i,k)
 434            end do
L435            phi(i) = phi_i
 436         end do
L437         if ( .not. do_fixup) then
 438
 439    c I-line recursion: without flux fixup
```

How much better is an EV67?

This question is ill formed for program units that are too small. Compilation issues, hardware cost attribution, etc.

Location	sort Ev6Cycle	%	sort Ev67Cycl	%	sorted cy6/cy67	%	sort Ev67BMis	%
Program	1.63e+10	100	1.09e+10	100	1.49e+00	100	1.78e+07	100
sweep.f: 431	6.40e+07	0	8.39e+06	0	7.62e+00	510	0.00e+00	0
source.f: 43	7.34e+06	0	1.05e+06	0	7.00e+00	469	0.00e+00	0
sweep.f: 394	2.52e+07	0	4.19e+06	0	6.00e+00	402	0.00e+00	0
sweep.f: 422	3.15e+07	0	5.24e+06	0	6.00e+00	402	0.00e+00	0
sweep.f: 423	3.15e+07	0	5.24e+06	0	6.00e+00	402	0.00e+00	0
sweep.f: 421	3.15e+07	0	5.24e+06	0	6.00e+00	402	0.00e+00	0

Parent Scope

LP 358-547:sweep.f	1.57e+10	97	1.05e+10	96	1.50e+00	100	1.57e+07	88

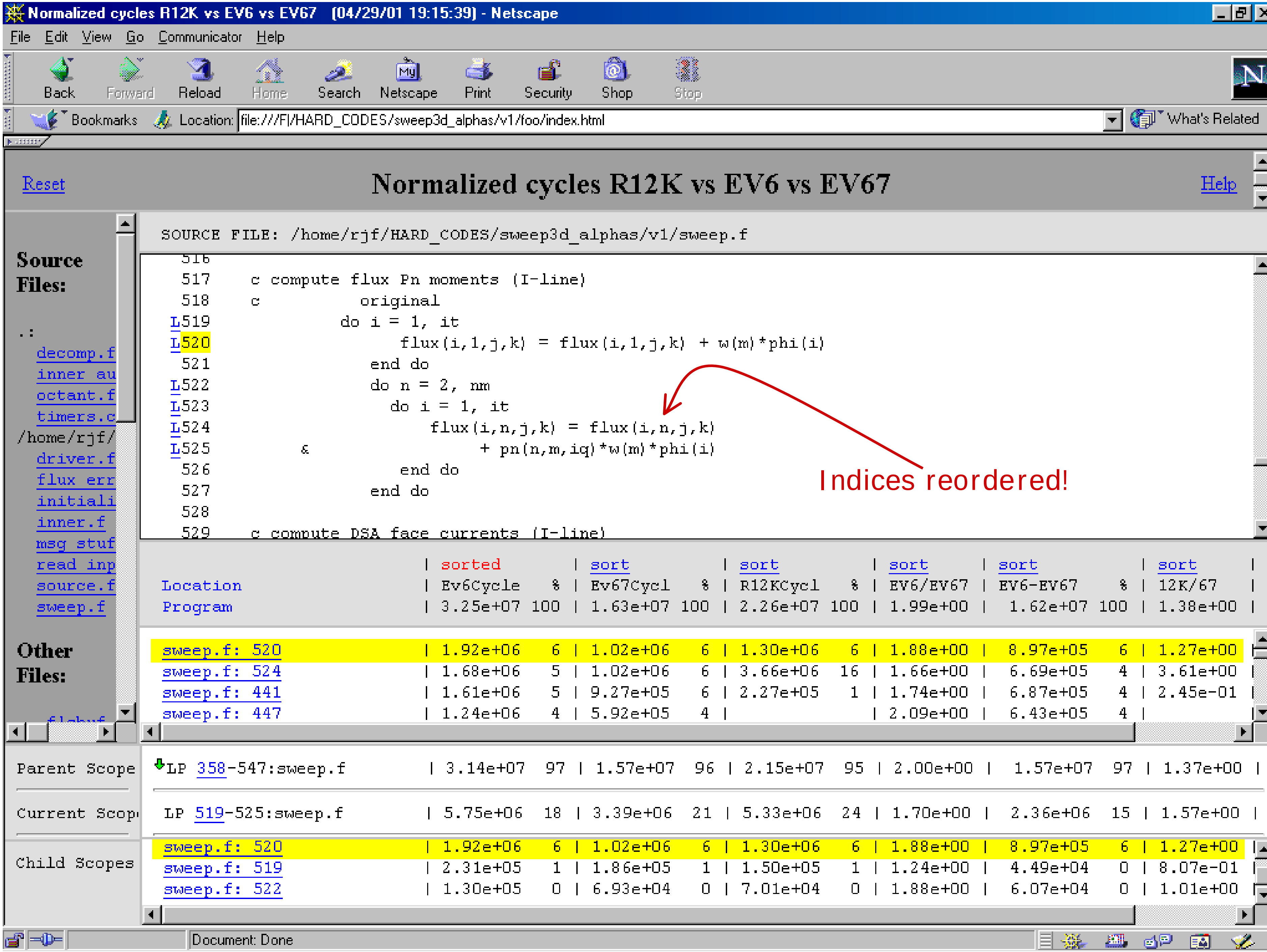
Current Scope

LP 430-435:sweep.f	2.01e+08	1	9.65e+07	1	2.09e+00	140	0.00e+00	0

Child Scopes

sweep.f: 431	6.40e+07	0	8.39e+06	0	7.62e+00	510	0.00e+00	0
LP 432-433:sweep.f	1.15e+08	1	6.40e+07	1	1.80e+00	121	0.00e+00	0
sweep.f: 430	1.99e+07	0	2.10e+07	0	9.50e-01	64	0.00e+00	0
sweep.f: 435	2.10e+06	0	3.15e+06	0	6.67e-01	45	0.00e+00	0

Document: Done



Normalized cycles R12K vs EV6 vs EV67 (04/29/01 19:15:39) - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: file:///F|/HARD_CODES/sweep3d_alphas/v1/foo/index.html What's Related

Reset

Normalized cycles R12K vs EV6 vs EV67

Help

Source Files:

.:
decomp.f
inner au
octant.f
timers.c
/home/rjf/
driver.f
flux err
initiali
inner.f
msg stuf
read inp
source.f
sweep.f

Other Files:

SOURCE FILE: /home/rjf/HARD_CODES/sweep3d_alphas/v1/sweep.f

```
  516
  517     c compute flux Pn moments (I-line)
  518     c          original
 L519              do i = 1, it
 L520                 flux(i,1,j,k) = flux(i,1,j,k) + w(m)*phi(i)
  521              end do
 L522              do n = 2, nm
 L523               do i = 1, it
 L524                   flux(i,n,j,k) = flux(i,n,j,k)
 L525          &                + pn(n,m,iq)*w(m)*phi(i)
  526                 end do
  527              end do
  528
  529     c compute DSA face currents (I-line)
```

Indices reordered!

Location	sorted Ev6Cycle	%	sort Ev67Cycl	%	sort R12KCycl	%	sort EV6/EV67	sort EV6-EV67	%	sort 12K/67
Program	3.25e+07	100	1.63e+07	100	2.26e+07	100	1.99e+00	1.62e+07	100	1.38e+00
sweep.f: 520	1.92e+06	6	1.02e+06	6	1.30e+06	6	1.88e+00	8.97e+05	6	1.27e+00
sweep.f: 524	1.68e+06	5	1.02e+06	6	3.66e+06	16	1.66e+00	6.69e+05	4	3.61e+00
sweep.f: 441	1.61e+06	5	9.27e+05	6	2.27e+05	1	1.74e+00	6.87e+05	4	2.45e-01
sweep.f: 447	1.24e+06	4	5.92e+05	4			2.09e+00	6.43e+05	4	
Parent Scope: LP 358-547:sweep.f	3.14e+07	97	1.57e+07	96	2.15e+07	95	2.00e+00	1.57e+07	97	1.37e+00
Current Scope: LP 519-525:sweep.f	5.75e+06	18	3.39e+06	21	5.33e+06	24	1.70e+00	2.36e+06	15	1.57e+00
Child Scopes: sweep.f: 520	1.92e+06	6	1.02e+06	6	1.30e+06	6	1.88e+00	8.97e+05	6	1.27e+00
sweep.f: 519	2.31e+05	1	1.86e+05	1	1.50e+05	1	1.24e+00	4.49e+04	0	8.07e-01
sweep.f: 522	1.30e+05	0	6.93e+04	0	7.01e+04	0	1.88e+00	6.07e+04	0	1.01e+00

Document: Done

Thought for the Day

The Hitchiker's Guide to the Galaxy, in a moment of reasoned lucidity which is almost unique among its current tally of five million, nine hundred and seventy-three thousand, five hundred and nine pages, says of the Sirius Cybernetics Corporation products that “It is very easy to be blinded to the essential uselessness of them by the sense of achievement you get from getting them to work at all. In other words - and this is the rock-solid principle on which the whole of the Corporation's galaxywide success is founded -- their fundamental design flaws are completely hidden by their superficial design flaws.”

(Douglas Adams, "So Long, and Thanks for all the Fish")

Contacts

NCSA activities: http://www.cs.rice.edu/~dsystem/ncsa/index.htm

HPCView tools: http://www.cs.rice.edu/~dsystem/hpcview/

Group: hpc@cs.rice.edu

Rob: rjf@rice.edu

John: johnmc@rice.edu

David: whalley@cs.fsu.edu