

Lightning Fast Startup

Michael Meeks
& Jan Holesovsky

September 18, 2006



*“Stand at the crossroads and look; ask for the ancient paths, ask where the good way is, and walk in it, and you will find rest for your souls...” -
Jeremiah 6:16*

Novell®

Overview

- Performance problems
- Some approaches
 - genuine speedups / round-up from last year.
 - fast splash / 2nd starter
 - systray app
- Memory savings / profiling.
- Conclusions / Questions

The problem archetypes

- scraping data off the spinning oxide
 - Intel doing some work here
- algorithmic sillies
 - linking
 - poor I/O management
 - bad code
 - > legacy optimisations, 'clever' hacks etc.
- memory burn

Startup profile / linking



Cycle Estimation

Search: ELF Object

- Self ELF Object
- 1 844 025 145 ld-2.3.90.so
 - 668 407 868 configmgr2.uno.so
 - 574 582 449 libc-2.3.90.so
 - 408 847 459 libuno_sal.so.3
 - 133 243 472 libfontconfig.so.1.0.4
 - 103 815 389 libuno_cppu.so.3
 - 101 947 336 libpsp680li.so

Before:

Incl.	Self	Called	Function
4 591 111 075	694	(0)	0x00000810
1 733 183 131	11 414 325	63 059	_dl_lookup_syml
1 680 244 936	1 265 152 958	88 591	do_lookup_x
1 503 104 191	42 597 208	149	_dl_relocate_obj
1 418 894 633	452 028	161	dl_open_worker
440 413 722	440 413 722	3 564 951	strcmp
276 872 663	6 564 190	12 640	_dl_fixup
276 781 111	180 310	12 630	_dl_runtime_reso
101 072 485	5 043	1	_dl_start
101 066 985	3 596	1	_dl_sysdep_start
101 062 498	21 351	1	dl_main
53 893 795	10 687 075	7 087	_dl_map_object
46 379 171	15 578 777	268 954	_dl_name_match
43 423 517	374 297	6 925	openaux
42 199 815	502 904	6 790	_dl_catch_error2
40 515 338	40 515 338	63 059	_dl_elf_hash
10 128 140	78 162	1 143	call_init <cycle 1
5 212 988	147 653	939	_dl_catch_error <
3 337 715	2 231 567	58	_dl_map_object
2 789 716	382 949	149	_dl_check_map
2 461 693	1 395 338	147	_dl_map_object

0x00000810

Types Callers All Callers Source Callee Map



Incl.	Self	Distance	Calling	Callee
4 487 810 890	508	1	1	0x08064120 (soffice.bin)
4 487 793 178	752	2	1	__libc_start_main (libc-2.3...
4 487 738 564	483	3	1	main (soffice.bin)
4 486 908 455	1 096	4	1	sal_main (soffice.bin)

Caller Map Paris Call Graph Callees All Callees Assembler



Cycle Estimation

Search: ELF Object

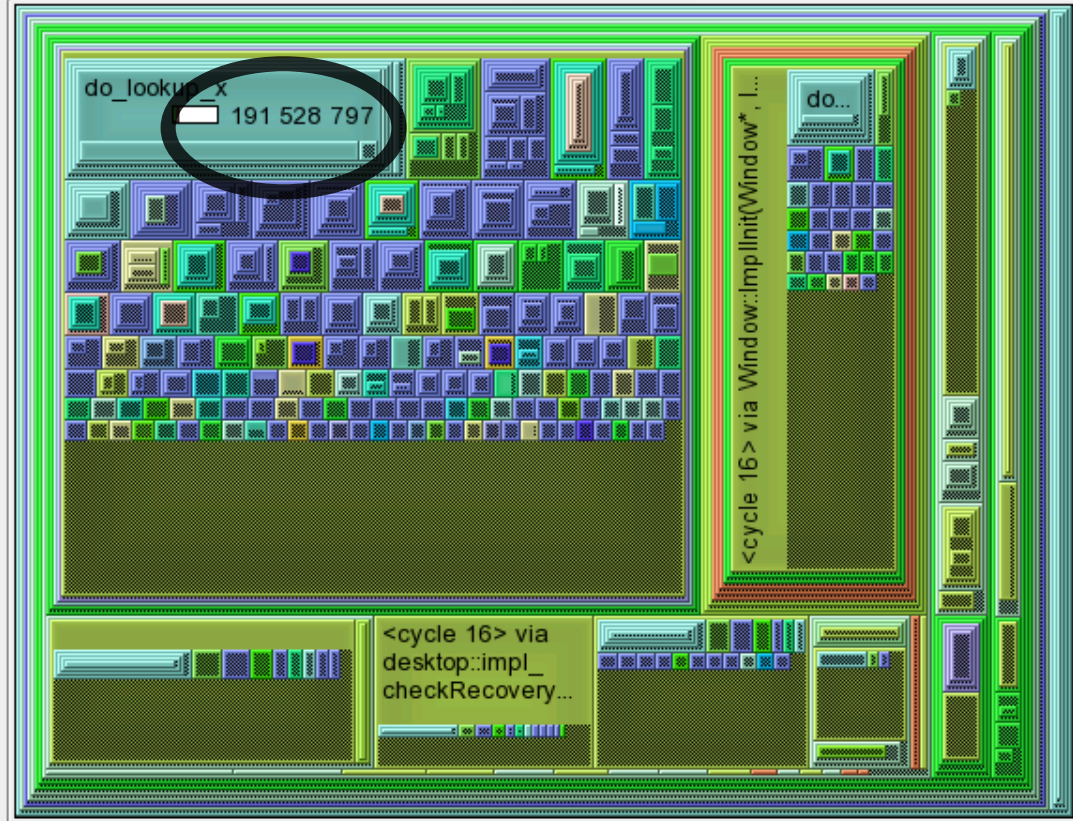
Self	ELF Object
740 398 622	configmgr2.uno.so
663 665 044	ld-2.3.90.so
548 032 624	libc-2.3.90.so
455 196 515	libuno_sal.so.3
132 285 679	libfontconfig.so.1.0.4
108 870 174	libuno_cppu.so.3
104 087 068	libuno_cppuhelpergcc3.so.3

After

Incl.	Self	Called	Function
90 245 371	1 676 902	3 648	configmgr::configuration::...
80 675 656	2 819 420	47 990	configmgr::configuration::...
74 083 116	10 709 892	785 896	rtl::OUString::~OUString(
69 342 341	1 011 270	31 578	configmgr::view::getMem
69 338 569	4 505 176	625 928	configmgr::memory::Acce
65 166 182	13 668 516	634 802	configmgr::memory::Acce
60 392 804	2 617 734	88 176	configmgr::view::ViewTre
59 815 769	3 668 668	108 633	configmgr::memory::Acce
59 150 778	1 373 563	85 841	configmgr::configuration::...
56 320 390	347 511	12 099	configmgr::configuration::...
55 954 229	590 730	12 100	configmgr::configuration::...
54 627 773	1 433 685	48 147	configmgr::data::GroupN
53 097 794	5 408 986	48 147	configmgr::data::GroupN
53 089 973	428 172	12 124	configmgr::configuration::...
53 033 170	3 026 734	88 717	configmgr::view::Tree::Tre
51 958 124	227 646	6 494	configmgr::configuration::...
51 344 391	531 644	21 136	configmgr::view::ViewTre
51 140 415	1 499 130	12 124	configmgr::configuration::...
50 770 475	789 678	21 136	configmgr::view::ViewStr
49 829 002	2 612 868	88 717	configmgr::configuration::...
49 079 248	1 412 524	12 604	configmgr::configuration::...

0x00000810

Types Callers All Callers Source Callee Map



Incl.	Self	Distance	Calling	Callee
3 329 213 237	388	1	1	0x08064110 (soffice.bin)
3 329 199 849	1 084	2	1	__libc_start_main (libc-2.3.9...
3 322 308 405	513	3	1	main (soffice.bin)
3 321 648 741	1 184	4	1	sal_main (soffice.bin)

Caller Map Paris Call Graph Callees All Callees Assembler

Linking fixes

- **-zdynsort**
 - re-written by RedHat - --hash-style – in new glibc / binutils - ~50% linking win
- **-Bdirect**
 - direct linking variant – still in wilderness, **Ulrich Drepper** refuses to discuss it, or even read the patch carefully.
 - 75% linking speedup
- **reduced linking scope**
 - a hack to deal with vague linkage
 - substantial win.

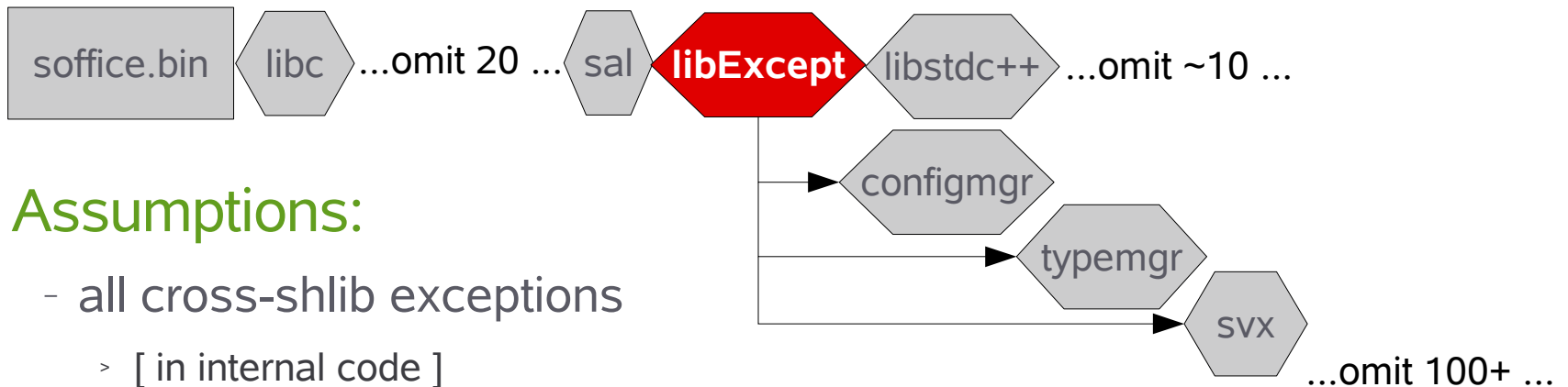
Reduced linking scope

- Special case Internal components to shrink lookups
 - > an evil hacked to defeat the ODR, and lookup only in a small scope.

Before:



After:



- Assumptions:

- all cross-shlib exceptions
 - > [in internal code]
- declared using UNO & hence compiled into 'libExcept'

- Substantial speedup.

Slight of hand speedups

How other people cheat

- Firefox
 - statically links
 - still slow to read
 - factory process
- KDE – kdeinit
 - kills link issues
- MS Office
 - greater fragmentation into components
 - > runs up against link times $O(\text{num-libraries})$
 - I/O re-ordering
 - > not present on Linux, cf. Intel's talk.

Ways we cheat already

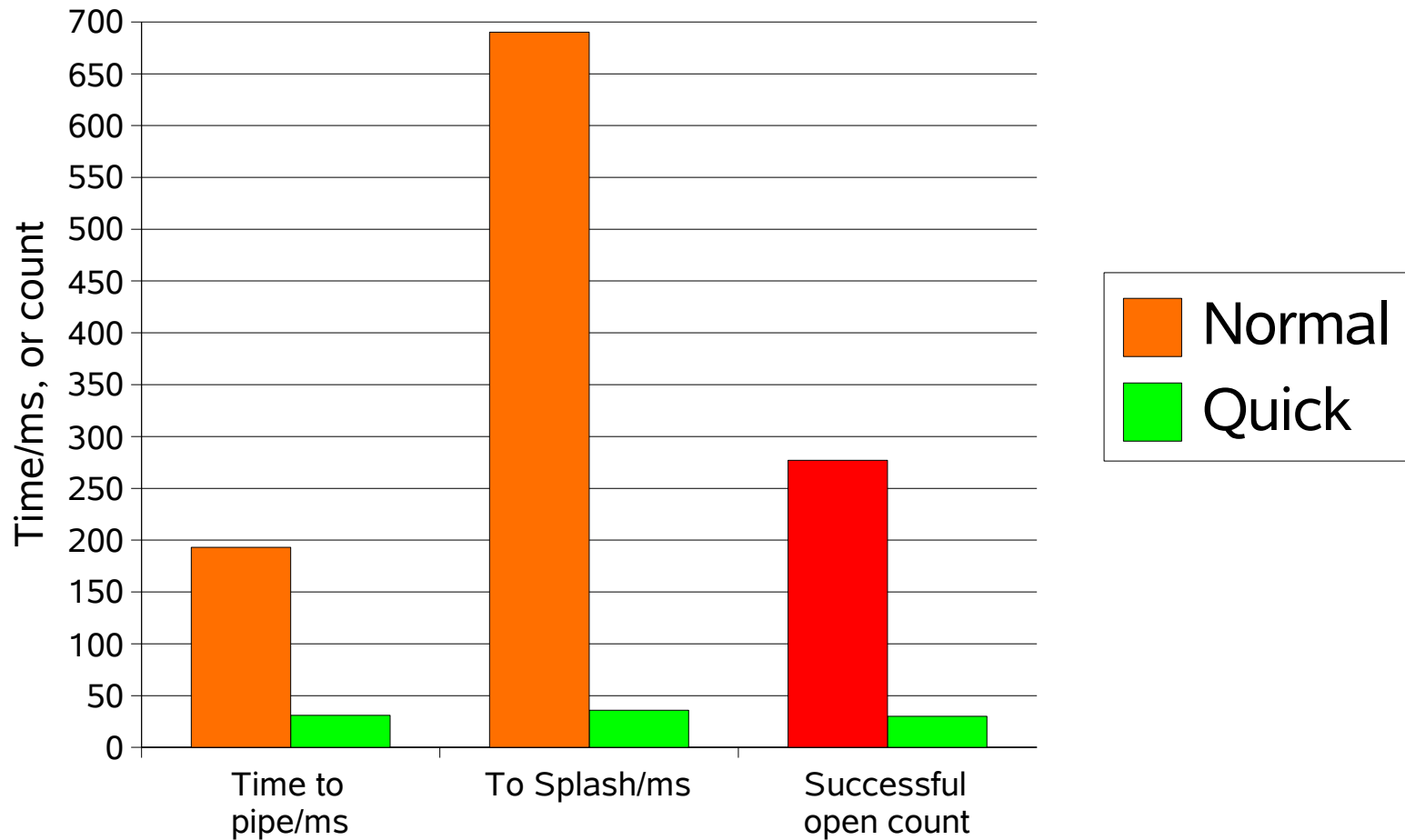
- **Factory process**
 - OO.o sits in the background
 - the 2nd time we run it – we start quicker
- **Caches**
 - configmgr stores big caches in ~/.ooo-2.0
 - imagemgr stores big cache in ~/.ooo-2.0

Apparently faster startup

- Quick 2nd starter:
 - detects factory process
 - if not – starts splash ~immediately
 - else pipes arguments to main process
- Perception rather than reality: Marketing.

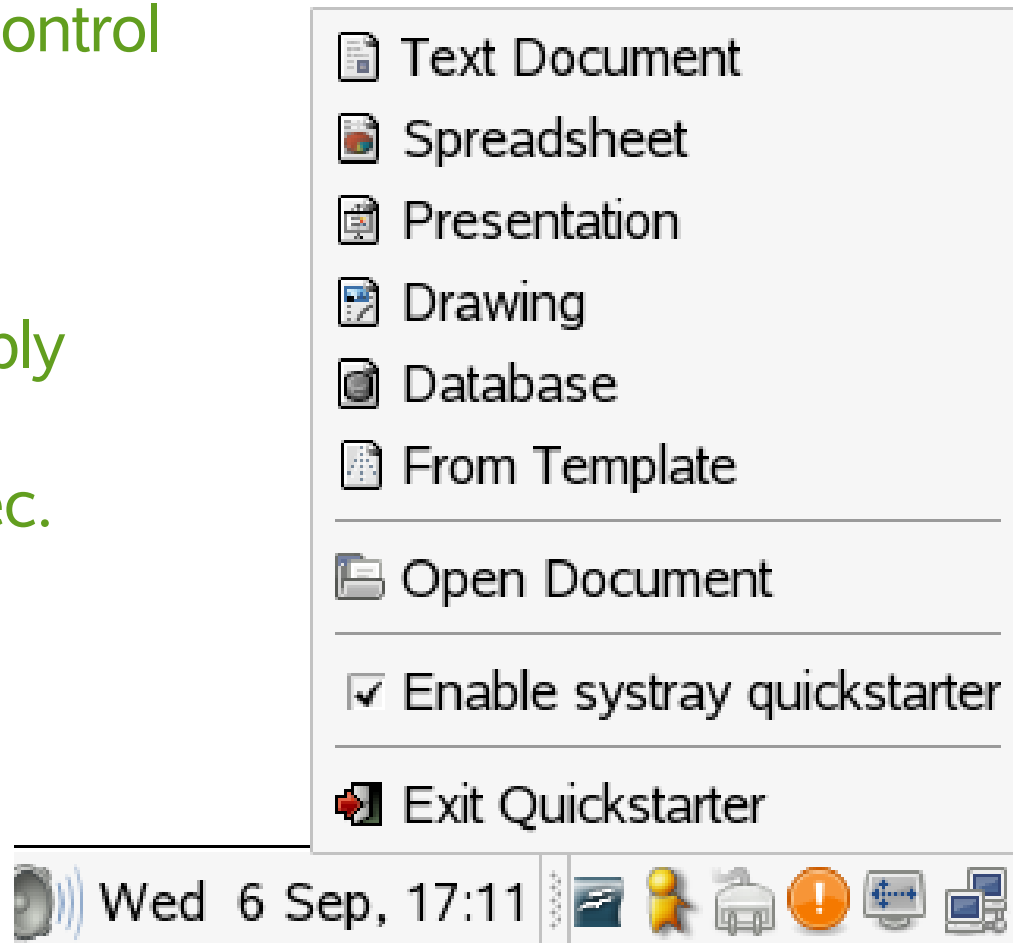


Timings / data: (warm)



Making it usable: the systray ...

- UI for lifecycle control
- cheesy buttons
- semi-instant apply
- freedesktop spec.
- 1st cut in 2.1 (?)



Memory issues ...

Memory tools ...

- Massif

- space **time** not a useful measure
 - > blighted by full-run-then-analyse design.

- speedprof / memprof

- ideal, accurate, full-stack information
- no re-compile necessary, interactive profile request
- <http://wiki.services.openoffice.org/wiki/Speedprof>

Memory tools ...

MemProf - ./soffice.bin (5744) - Running

File Process Settings Help

Run Kill Profile Leaks Save Record

0k 16384k

of Allocations:120660 Bytes / Allocation:78.55 Total Bytes:9478103

Profile Leaks

Name	Self ▲	Total
/data/OODInstall/progra...	7848748	7997490
/usr/lib/libfontconfig.so.1...	543229	543229
/usr/lib/libfreetype.so.6.3.8	240536	240536
_dl_new_object	78079	78079
XOpenDisplay	58500	59266
g_object_constructor	51928	165767
__dcigettext	41985	64955
__new_exitfn	34840	34840
/usr/lib/libcrypto.so.0.9.8	27896	41608
g_mem_chunk_alloc0	27392	27392
_XEnq	25064	25064

Callers

Name	Self	Total ▲
_STL::__node_alloc<true, 0>::_S_chunk_alloc(...	0	1721368
/data/OODInstall/program/libuno_cppuhelpergc...	0	212364
salhelper::SimpleReferenceObject::operator ne...	0	188424

Descendants

Name	Self	Cumulative ▲
▼ operator new(unsigned int)	0	3486594
▶ allocate(unsigned int, (anonymous n...	0	3486594

Image strips / lists

- 3 problems:
 - images.zip file read every start anyway
 - large ~2.4Mb of image cache in ~/.ooo-2.0
 - image strips in cache – horizontal
 - > extremely poor working set effects:
- Re-worked ImageList - (cws ka009)
 - saved 3 Mb overall, and ~Mb per window
 - > lost multi-Mb in ~/.ooo-2.0
 - > each images.zip entry – appears just once
 - scripts to re-sort images.zip – to avoid seek problems
 - startup performance neutral, but some 1st time latency

Strings ...

- Lots of strings are allocated repeatedly
 - `makeAny(rtl::OUString::createFromAscii("Foo"))`; etc.

Total ustring size

739,436

Total wastage

666,164

Unique strings

73,272

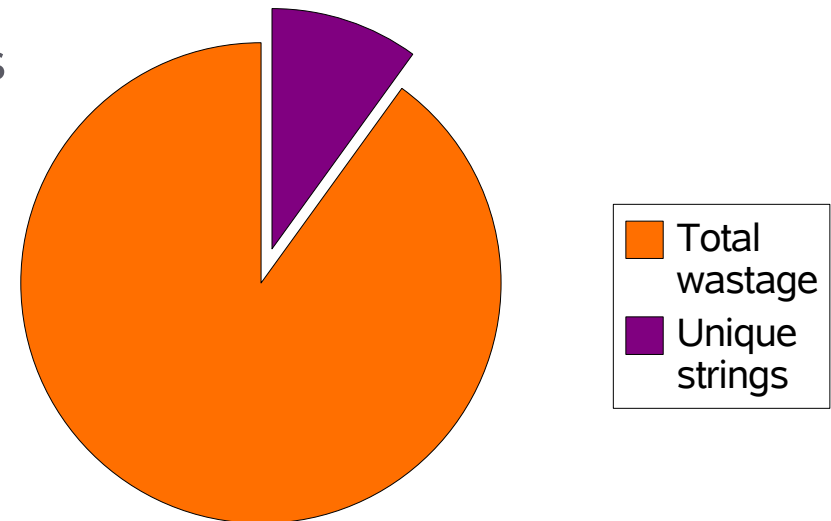
- >90% of strings are duplicates

- a number of scattered patches
- configmgr yet to go
 - > lots of fat there ...

- malloc overhead

- real savings: larger

String duplication



Future work ...

- Code/shared size: still most significant: ~100Mb.
- Configmgr
 - remove custom structures and shared memory arch.
 - new chunky API
- ScCell
 - huge vs. Excel
 - patch to remove ScPostIt * from base cell
 - each cell has a row-index in column*.cxx
 - > better sparse array needed.
 - string duplication needs fixing
 - Data Pilot representation not efficient

Conclusion

Startup:

linking getting slowly fixed
configmgr a large issue

Memory

code-size still critical
configmgr needs love

Strings:

getting fixed.
check your code for offenders

<http://go-oo.org/~michael/ustrings.ods>

Oh, that my words were recorded, that they were written on a scroll, that they were inscribed with an iron tool on lead, or engraved in rock for ever! I know that my Redeemer lives, and that in the end he will stand upon the earth. And though this body has been destroyed yet in my flesh I will see God, I myself will see him, with my own eyes - I and not another. How my heart yearns within me. - Job 19: 23-27

Novell®