

如何定位性能瓶颈

51Testing

第二十八期软件测试沙龙
(深圳站)

性能瓶颈定位和性能测试



围绕三个目开展的性能测试：

- 通过性能测试，评估系统的性能指标。例如：为了写产品说明书
- 通过性能测试，评估系统的各种性能参数是否满足需求规定
- 已经知道系统存在性能问题，通过性能测试定位问题

- 性能瓶颈：制约系统性能的最主要因素
- 性能瓶颈定位：指的是为了找出制约（系统、路径等）性能的最主要的因素而展开的分析、设计、测试、比较、调优等工作。

性能瓶颈分类



- 根据性能瓶颈影响范围的不同和触发时间的不同，我们可以将其分为三类
 - 系统类
 - 事件类
 - 路径类

性能瓶颈表现形式



- 两大表现形式
 - 响应时间
 - 资源占用

性能瓶颈定位目的



- 两个目的
 - 改善用户体验
 - 用户体验是非常专业的一个研究方向，产品设计人员必修课。和性能相关的UE指标是响应时间（从用户发出请求，到接收到反馈）
 - 降低成本
 - 通过解决软件中的性能问题,减少处理相同业务对硬件资源的消耗从而降低成本。

影响性能问题的主要因素



— 影响性能问题的主要因素

- 硬件
- 带宽
- 程序
 - Web server
 - App server
 - 客户端
 - 中间件
 - 三方控件
- DBMS
-

影响性能的因素很多，但是归根结底还是能从程序中找到答案

性能瓶颈定位



- 系统瓶颈
 - 分析
 - 设计用例
 - 执行用例
 - 分析性能数据
 - 调优
 - 比较

性能瓶颈定位



- 路径瓶颈
 - 分析
 - 设计用例
 - 执行用例
 - 分析性能数据
 - 调优
 - 比较

性能瓶颈定位细节



- 应用程序必需单独安装在（输出至）一个空的分区，如果再次测试改程序不同的版本，则需要格式化该分区。每次测试程序的分区必需保持不变，包括分区容量、文件系统和分配单元大小。这是为了保持磁盘I/O影响的恒定。如果随意放置到不同的分区，或者和其它程序混装在一个分区，可能会导致和I/O相关的误差在20%以上。
- 测试机器保持清洁，不要安装过多程序，尤其是定期活动或引起网络连接的程序。
- 需要重启操作系统，重启后等待一段时间。知道操作系统达到稳定状态。CPU和内存占用恒定。网络占用和I/O活动都为0。

性能瓶颈用例设计



- 性能瓶颈用例设计
 - 性能瓶颈用例设计的目的和功能用例设计目的不一样
 - 功能是为了发现缺陷
 - 性能是为了达到可靠的覆盖率
 - 通过用例的执行，达到覆盖可能出现瓶颈的代码的目的

性能瓶颈定位细节



- 测试电脑的电源管理功能，有的CPU有自动调整运行频率的功能。为此需要在测试时设定为固定频率运行。独立显卡也有类似的问题
- 每次比较测试都遵循这样的原则，那么得到的数据就是稳定可靠的了

性能瓶颈定位



- 事件瓶颈
 - 分析
 - 设计用例
 - 执行用例
 - 分析性能数据
 - 调优
 - 比较

- 定位不同的性能瓶颈时采取的流程基本类似，在具体实施时略有差异
- 性能瓶颈定位特点：
 - 参与人员多，需要多工种配合、客服人员、测试人员、开发人员等
 - 采用方法多，结合使用测试分析法、静态测试、动态测试等方法
 - 借助工具多，要结合使用白盒测试工具，测试工具，测试分析工具等

性能数据分析



File Edit Options Window Help

All (Modules: 16 Methods: 895)

COMPUTER - 560 (Deadlock Demo)

Source (8.6042%)

Deadlock Demo.exe (8.6042%)

- DeadlockPhilosophersView.cpp (
- DeadlockPhilosophersDoc.cpp (1
- MainFrm.cpp (1.4378%)
- vector (.1879%)
- Philosopher.cpp (.1639%)
- MemDC.h (.1093%)
- Chopstick.cpp (.0956%)
- PhilDlg.cpp (.0459%)
- DeadlockPhilosophers.cpp (.033
- PhilNumberDlg.cpp (.0274%)
- PhilChooser.cpp (.0262%)
- xmemory (.0134%)
- new (.0019%)
- xutility (.0012%)

System (91.3958%)

- Top 20 Source Methods
- Top 20 Methods
- Top 20 Called Source Methods
- Top 20 Called Methods

Method Name	% in Method	% with Children	Called	Image	% in Image	Average ...	First (m
#4749 (mfc42d.dll)	.0000	.0000	1	MFC42D.DLL	.0000	.0000	.0000
#4408 (mfc42d.dll)	.0000	87.3228	1	MFC42D.DLL	.0000	.0000	.0000
#3373 (mfc42d.dll)	.0000	.0000	1	MFC42D.DLL	.0000	.0000	.0000
#1797 (mfc42d.dll)	.0000	.0000	1	MFC42D.DLL	.0000	.0000	.0000
#3938 (mfc42d.dll)	.0000	.0000	1	MFC42D.DLL	.0000	.0000	.0000
CrtSetReportHook	.0000	.0000	1	MSVCRTD.DLL	.0019	.0006	.0000
GetCurrentProcess	.0000	.0000	1	kernel32...	.0002	.0006	.0000
GetCurrentThread	.0000	.0000	1	kernel32...	.0002	.0006	.0000
CrtSetDumpClient	.0000	.0000	1	MSVCRTD.DLL	.0020	.0006	.0000
GetThreadLocale	.0000	.0000	1	kernel32...	.0002	.0006	.0000
#3244 (mfc42d.dll)	.0000	.0000	1	MFC42D.DLL	.0001	.0006	.0000
GetCommandLineA	.0000	.0000	1	kernel32...	.0002	.0007	.0000
_p_commode	.0000	.0000	1	MSVCRTD.DLL	.0022	.0007	.0000
_p_argc	.0000	.0000	1	MSVCRTD.DLL	.0022	.0007	.0000
GetSystemTimeAsFileTime	.0000	.0000	1	kernel32...	.0002	.0007	.0000
(unknown)	.0000	.0000	1	MFC42D.DLL	.0001	.0007	.0000
_p_fmode	.0000	.0000	1	MSVCRTD.DLL	.0025	.0008	.0000
#308 (mfc42d.dll)	.0000	.0000	1	MFC42D.DLL	.0001	.0008	.0000
CrtSetReportMode	.0000	.0000	1	MSVCRTD.DLL	.0027	.0008	.0000
IsChild	.0000	.0000	1	user32.dll	.0001	.0009	.0000
_set_app_type	.0000	.0000	1	MSVCRTD.DLL	.0028	.0009	.0000
wcscpy	.0000	.0000	1	msvcrt.dll	.1738	.0009	.0000
CheckMenuItem	.0000	.0000	1	user32.dll	.0001	.0009	.0000
GetSystemTime	.0000	.0000	1	kernel32...	.0003	.0009	.0000
#4176 (mfc42d.dll)	.0001	.0001	3	MFC42D.DLL	.0002	.0004	.0000
strtol	.0001	.0001	1	MSVCRTD.DLL	.0040	.0012	.0000
GetCommandLineW	.0001	.0001	2	kernel32...	.0004	.0006	.0000
FreeEnvironmentStringsA	.0001	.0001	1	kernel32...	.0004	.0013	.0000
GetProcessVersion	.0001	.0001	1	kernel32...	.0004	.0013	.0000
#1748 (mfc42d.dll)	.0001	.0001	2	MFC42D.DLL	.0002	.0007	.0000
free	.0001	.0001	1	msvcrt.dll	.2697	.0014	.0000
GdiValidateHandle	.0001	.0001	2	gdi32.dll	.0003	.0007	.0000
FreeSid	.0001	.0001	2	advapi32...	.1735	.0007	.0000
#4017 (mfc42d.dll)	.0001	.0001	3	MFC42D.DLL	.0003	.0005	.0000
ImmResizeIMCC	.0001	.0004	1	imm32.dll	.3341	.0015	.0000
lstrcpynA	.0001	.0001	2	kernel32...	.0004	.0008	.0000
#734 (mfc42d.dll)	.0001	.0001	1	MFC42D.DLL	.0003	.0015	.0000
#2103 (mfc42d.dll)	.0001	.0001	2	MFC42D.DLL	.0003	.0008	.0000
#4143 (mfc42d.dll)	.0001	.0001	1	MFC42D.DLL	.0003	.0016	.0000
#3836 (mfc42d.dll)	.0001	.0005	1	MFC42D.DLL	.0003	.0016	.0000
#1100 (mfc42d.dll)	.0001	.0001	1	MFC42D.DLL	.0003	.0016	.0000
RtlReAllocateHeap	.0001	.0001	1	ntdll.dll	.0065	.0016	.0000

常用指标

- % of Method 列
 - 执行当前语句及其调用的method耗时占执行整个method的百分比
- % in Method 列
 - 执行该Method耗时占整个session的百分比
- % in Image列
 - 执行该Method耗时占image中所有Method执行的百分比

常用指标

- %with Children 列
 - 执行该Method以及子函数耗时占整个session的百分比
- Average 列
 - 在整个会话期间，执行该method的平均时间
- Average with Children列
 - 在整个会话期间，执行该method及其子method的平均时间

常用指标

- **Called** 列
 - 方法被调用的次数
- **Child Methods**
 - 被该Method调用的Method
- **Count** 列
 - 该语句被执行的次数
- **Cycle**
 - CPU完成一个时钟周期的时间，对于150MHZ的处理器来说，需要消耗

常用指标

- **First列**
 - 该方法收集首次执行时间
- **Method**
 - 你的程序中用到的函数和方法
- **Method 列**
 - 该行调用的Method的个数
- **Image列**
 - 包含该方法的Image

常用指标

- **Maximum**
 - Method执行最大耗时
- **Minimum**
 - Method执行最少耗时
- **Real 列**
 - 函数执行的真实时间，包括函数等待关联函数返回值的时间
- **Time列**
 - 该语句以及它所调用的方法的平均执行时间

个人常用指标



- FCC (Function Called Count)
 - 函数被调用次数

- FAT(Function Average Time)
 - 函数执行平均时间

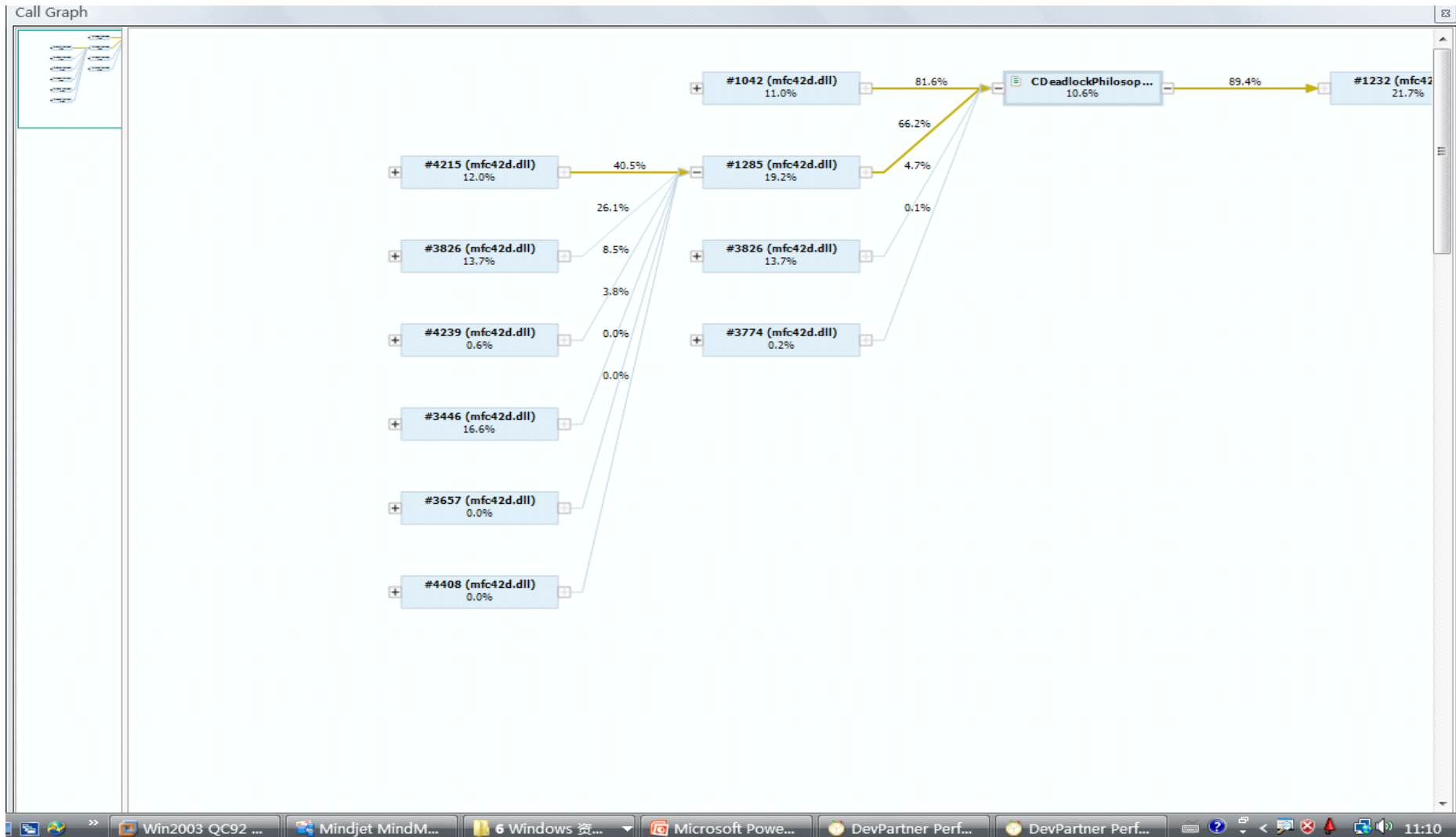
- FTT (Function Total Time)
 - 函数总耗时间
 - PEC
 - 页面出错次数

精确定位—代码行

The screenshot shows the Visual Studio performance profiler interface. The left pane displays a tree view of the application's modules and source files. The main pane shows a table of method execution statistics for the selected source file, 'DeadlockPhilosophersView.cpp'. The table columns are 'Count', '% with Children', 'Time (ms)', and 'Source'. The method 'void CDeadlockPhilosophersView::OnPhilosophersView' is highlighted in grey, indicating it is the most time-consuming method.

Count	% with Children	Time (ms)	Source
1	.0000	.0006	void CDeadlockPhilosophersView::OnPhilosophersView {
1	.0000	.0006	if(!m_bThreadsRunning) {
1	.0008	.0186	CPhilNumberDlg theDlg; theDlg.SetNumber(m_nDiners);
1	.0003	.0056	theDlg.DoModal();
1	4.5794	100.8705	m_nDiners = theDlg.GetNumber(); SetTable();
1	.0001	.0027	}
1	.2121	4.6723	void CDeadlockPhilosophersView::OnUpdatePhil {
1	.0004	.0080	pCmdUI->Enable(!m_bThreadsRunning);
1	.0000	.0006	}
5	.0001	.0028	void CDeadlockPhilosophersView::OnPhilosophersView {
5	.0019	.0423	// Update the status bar CString str; CMainFrame* pFrame = (CMainFrame*) AfxGetApp()->GetMainFrm(); CStatusBar* pStatus = &pFrame->m_wndStatusBar; if (pStatus) { str.Format(AFX_IDS_RUNMESSAGE); pStatus->SetPaneText(0, str); }
5	.0001	.0030	}
1	.0000	.0005	void CDeadlockPhilosophersView::OnPhilosophersView {
1	.0001	.0026	str.Format(AFX_IDS_RUNMESSAGE);
1	.0001	.0032	pStatus->SetPaneText(0, str);
1	.0000	.0005	}
1	.0000	.0005	if (pStatus) {
1	.0144	.3173	str.Format(AFX_IDS_RUNMESSAGE);
1	.0215	.4736	pStatus->SetPaneText(0, str);
1	.0005	.0107	}

路径分析



覆盖率



File Edit Options Window Help

155 of 2538 lines executed (6.1072%)

57 of 898 methods called (6.3474%)

Method List Source[PhilChooser.cpp] Session Summary

Method Name	% Covered	Called	# Lines Not Exec...	# Lines Executed	Total # Lines	Image
std::vector<bool, class...	.0000	0	4	0	4	Deadlock...
std::vector<bool, class...	.0000	0	4	0	4	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	8	0	8	Deadlock...
std::vector<bool, class...	.0000	0	10	0	10	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	3	0	3	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	3	0	3	Deadlock...
std::vector<bool, class...	.0000	0	3	0	3	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	7	0	7	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	3	0	3	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	3	0	3	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	4	0	4	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	2	0	2	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	1	0	1	Deadlock...
std::vector<bool, class...	.0000	0	4	0	4	Deadlock...

Methods Not Covered

Methods Less Than 20% Covered

Over 30 Lines, Less Than 10% Covered