

Coign: Automated, Distributed Partitioning of Component Applications

Galen C. Hunt

Microsoft Research
Redmond, WA USA

Michael Scott

Dept. of Computer Science
University of Rochester
Rochester, NY USA

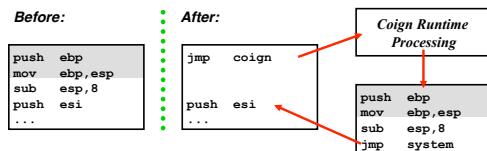
Why Partition Automatically?

- ◆ One-time solutions are expensive.
- ◆ Changes between networks can render a static partition obsolete.

Network	Raw Bandwidth	Available Bandwidth	Null RPC Round Trip	32KB Out RPC (Round Trip)	Ratio of 32KB to Null RPC (Round Trip).
ATM	155 Mbps	71.5 Mbps	519 μs	3.62 ms	7 : 1
100BaseT	100 Mbps	44.1 Mbps	383 μs	5.67 ms	15 : 1
10BaseT	10 Mbps	7.8 Mbps	636 μs	32.0 ms	50 : 1
ADSL Central	1.384 Mbps	1.1 Mbps	28.2 ms	220 ms	8 : 1
ADSL Remote	64 kbps	54.8 kbps	31.7 ms	4.56 s	144 : 1
ISDN (1 Channel)	64 kbps	53.5 kbps	59.4 ms	4.67 s	79 : 1

Binary Compatibility, Runtime Redirection

- ◆ All relevant COM library functions are patched to re-route control through the Coign runtime.
 - ✓ Low overhead.
 - ✓ Works for dynamic function pointers.



Problem:

- ◆ Distributed systems are need to:
 - ❖ for Remote access of distributed data,
 - ❖ for Use of remote processors and/or memory,
 - ❖ for Collaboration, etc.
- ◆ Distributing an application is non-trivial.
- ◆ **Insight:**
 - ❖ Use good software-engineering practices to create a sound non-distributed application,
 - ❖ Let system software decide how to partition and distribute the application.

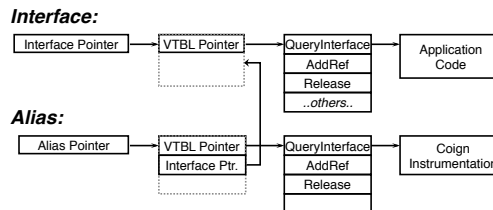
In-Process Communication Measurement

- ◆ Why not place each component in a separate process to measure communication?
- ◆ Out-of-process measurement fails for opaque interfaces.
 - ❖ Opaque interfaces cannot be remotod.
 - ❖ No a priori list of exported interfaces.

```
interface IVoid : IUnknown {
    HRESULT Opaque([in] void *pvData);
};
```

Binary Compatibility, Interface Aliases:

- ◆ Execution enters a new object through an alias.
- ◆ Any non-aliased interface passed out of a object belongs to that object.

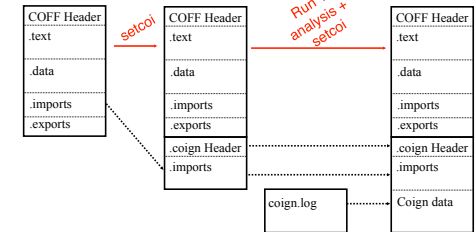


Solution:

- ◆ Good software engineering:
 - ❖ Applications should be divided into small reusable, functionally contained units.
 - ❖ **Component:**
A Reusable piece of software in binary form.
- ◆ Coign partitions and distributes components based on runtime communication patterns.
- ◆ Coign predicts distributed communication by measuring the data sent across component interfaces if the interfaces were distributed.

Binary Compatibility, Static Re-linker:

- ◆ Static re-linker adds DLL import for "coignrte.dll" and a section for partitioning profile data.



Per Interface Statistics:

Oxygen Text Props	[c12]	calls	BytesIn	Unc	BytesOut	Unc
Callers:						
Oxygen Composite Seq	[c08]					
IConnectionPointCon	10	1	4	0	24	0
IPersistStream	8	1	8	0	8	0
Oxygen Notifier	[c09]					
IConnectionPoint	24	2	8	0	12	0
		4	20	0	44	0
Callers:						
Oxygen Style Manager	[c11]					
IConnectionPointCon	10	8	192	0	32	0
IPersistStream	20	1	8	0	0	0
IUnknown	1	2	32	0	8	0
Oxygen Compound Props	[c13]					
ITextProps	e00	177	2832	0	2800	0
IUnknown	1	2	32	0	8	0
		190	3096	0	2848	0

[c12] Created by [c11] using coCreateInstanceEx.
[c12] Total execution time: 6ms.