

Shared Memory Management for VisualWorks



**Shared Memory Management
for VisualWorks**

Holger Guhl

Senior Smalltalker at Georg Heeg eK

www.heeg.de



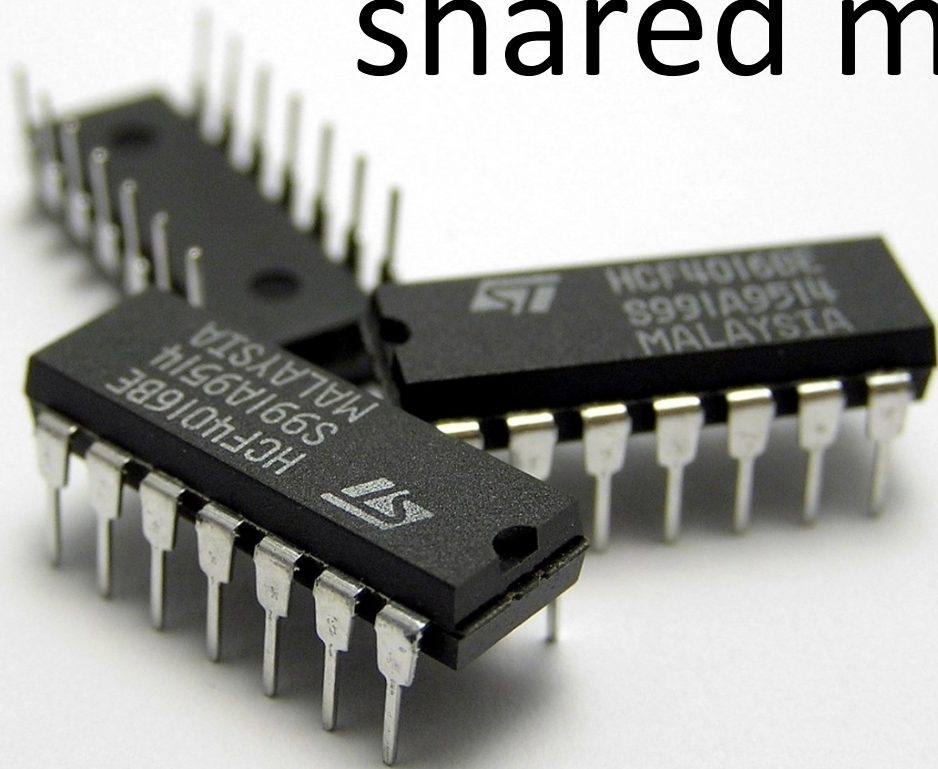
mailto: holger@heeg.de

Shared Memory Management for VisualWorks

What?

shared memory

Why?



Personal Experience:
software assessment project

huge customer project took 4 hours!



Polycephaly

= use all CPUs

Polycephaly Example

| drones result |

drones := VirtualMachines new: 4.

result := drones

do: '[:a :b | a + b]'

with: #(1 2 3 4 5)

with: #(5 4 3 2 1).

result = #(6 6 6 6 6)

Project “Software Assessment”



Perfect options for task distribution
(per package, class)



Needs to report much results

☹ Disappointing results:

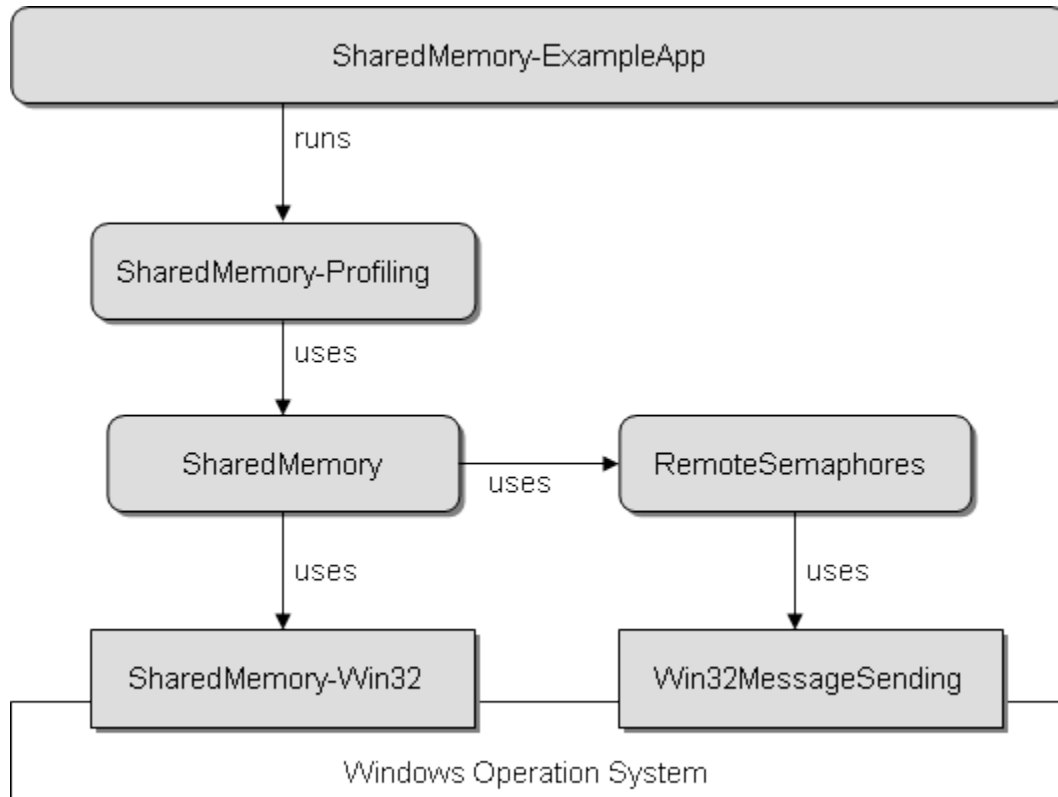
Communication overhead consumed all time gains



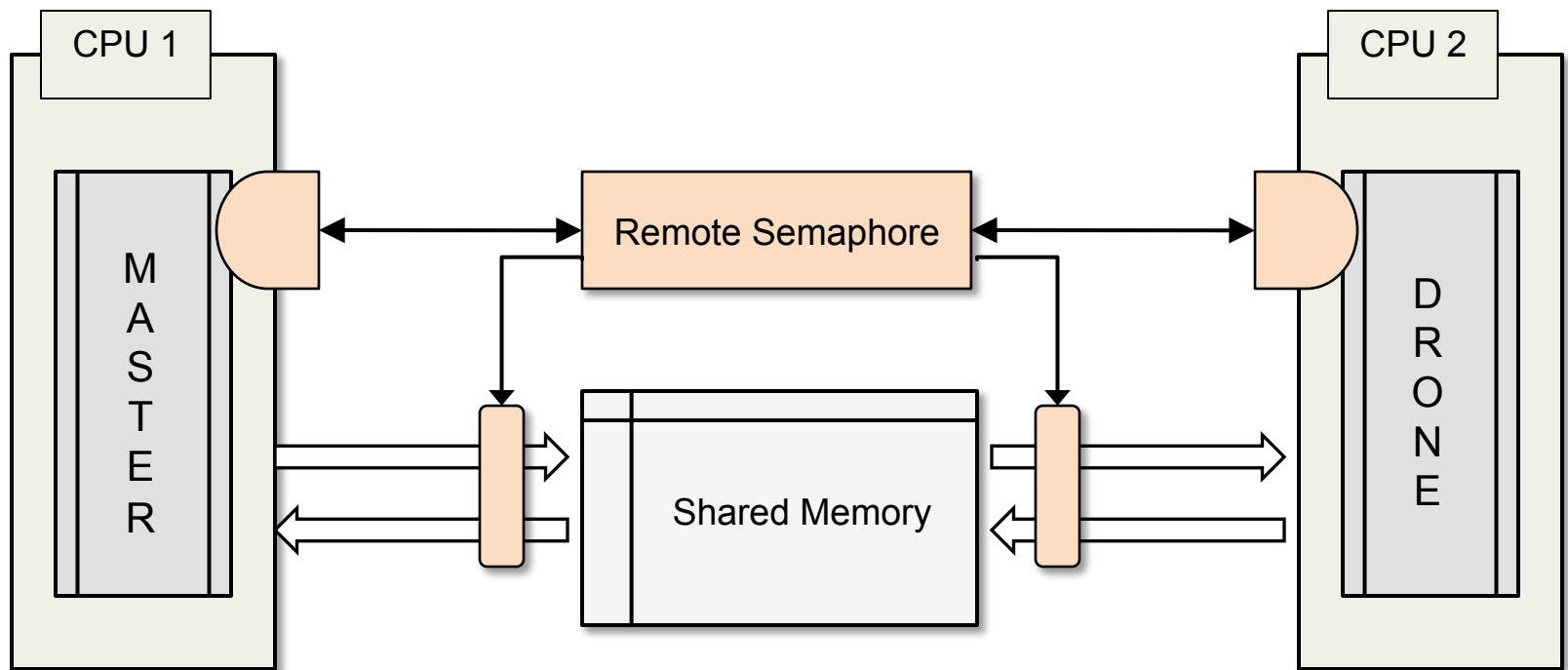
Shared Memory

= more speed?

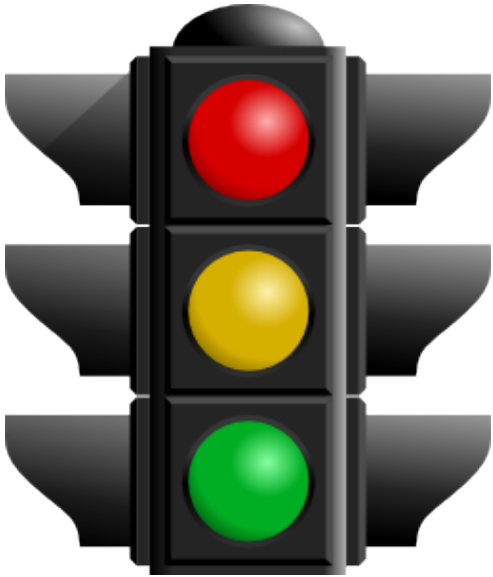
Tools Architecture



Application Scheme



RemoteSemaphore

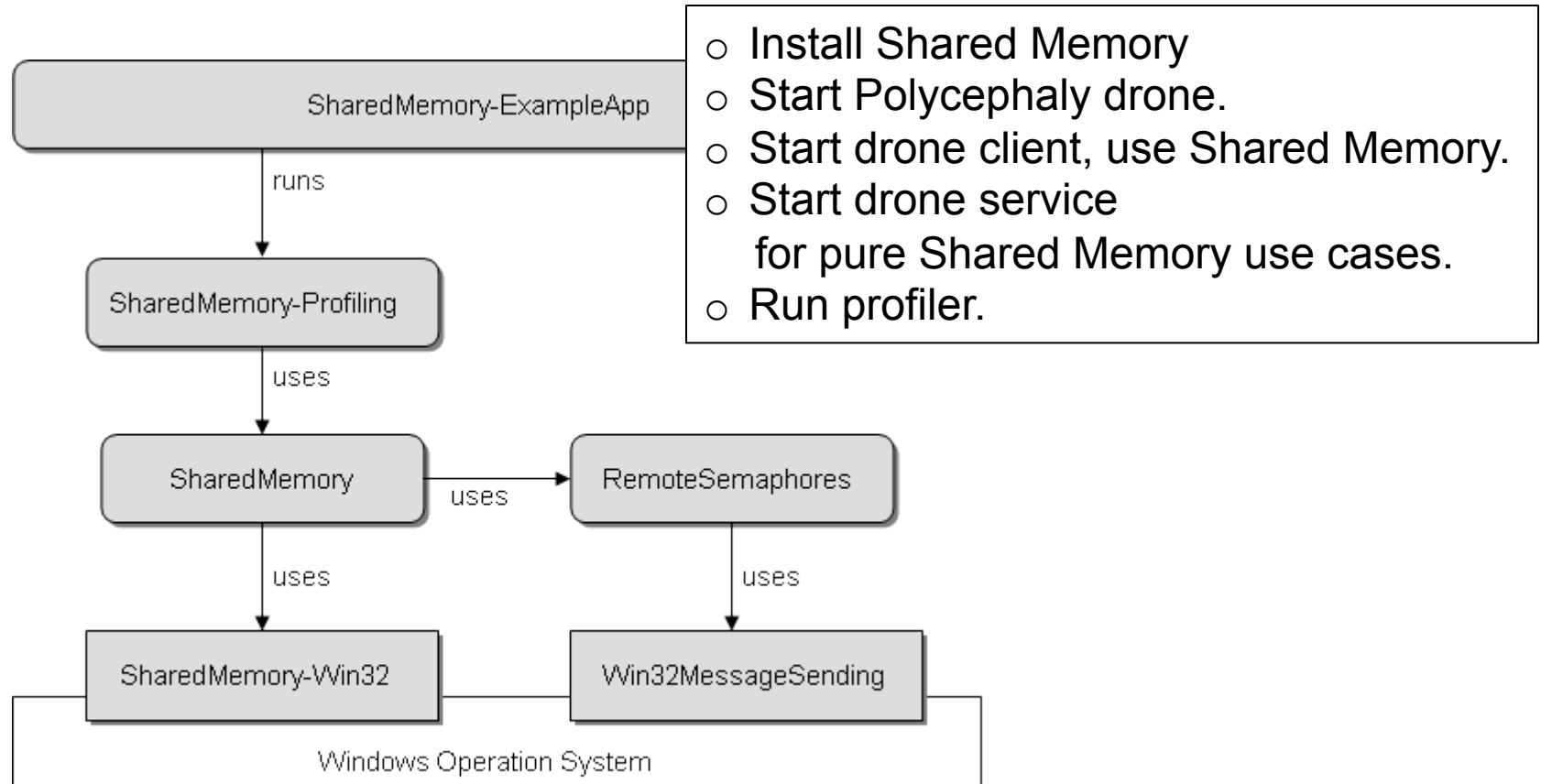


- Mutually exclusive Shared Memory access
- Created by master image
- Twin in drone image
- Synchronized via OS event messages

SemaphoreEventRelay

- Organize connection to other images
- Instances in connected images
- Communication center for remote semaphore messages
 - #signalRemote
 - #waitRemote

SharedMemory-ExampleApp



Time Profiles



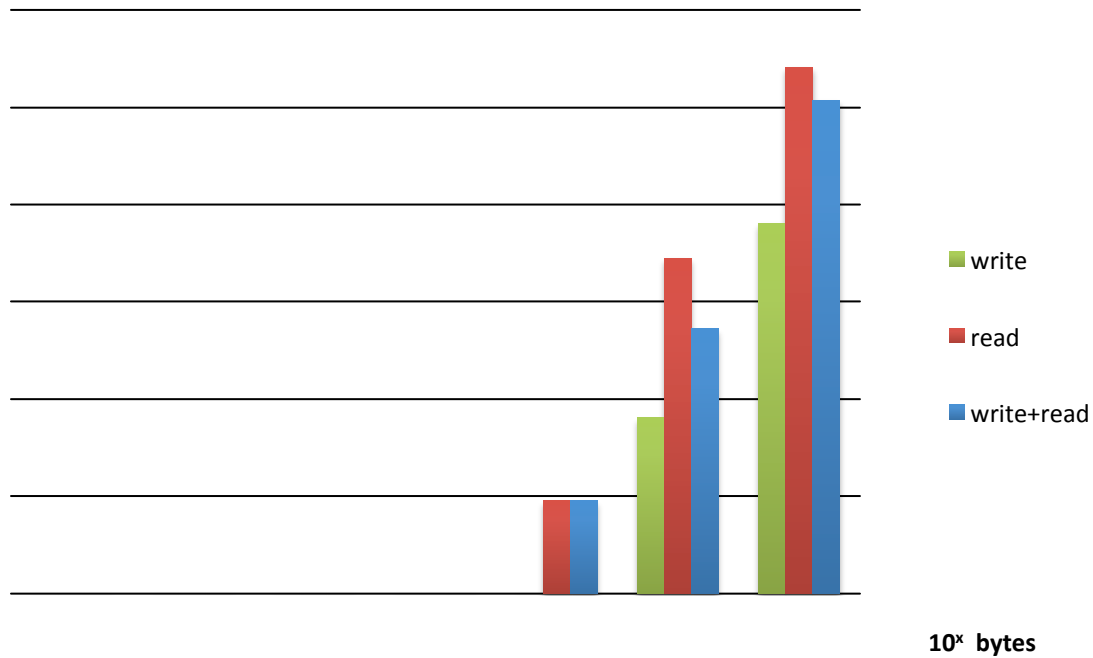
Shared memory raw read/write

Read bytes	Time (μs)	Write bytes	Time (μs)	Read+write	Time (μs)
<= 1,000	<= 1	1,000	1	1,000	1
10,000	3	10,000	1	10,000	3
100,000	53	100,000	8	100,000	23
1,000,000	509	1,000,000	80	1,000,000	343

Shared memory raw read/write

Shared mem write+read 10^x bytes

log μ s



Shared Memory Overhead

- SemaphoreEventRelay mutex roundtrip
 - with OS event messages = 117 μ s
 - with UDP socket messages = 399 μ s

Polycephaly Use Cases

Overhead: Request to evaluate the empty block = 12,519 μs

request bytes	Time (μs)	Request Array of size	Time (μs)
10	9,387	10	8,185
100	8,499	100	9,387
1,000	8,327	1,000	12,001
10,000	10,478	10,000	25,702
100,000	21,463	100,000	166,141
1,000,000	123,933	1,000,000	1,518,115

Polycephaly with Shared Memory

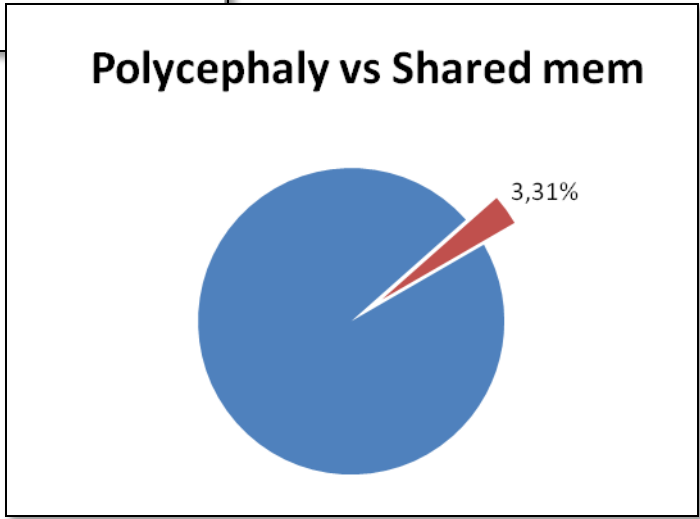
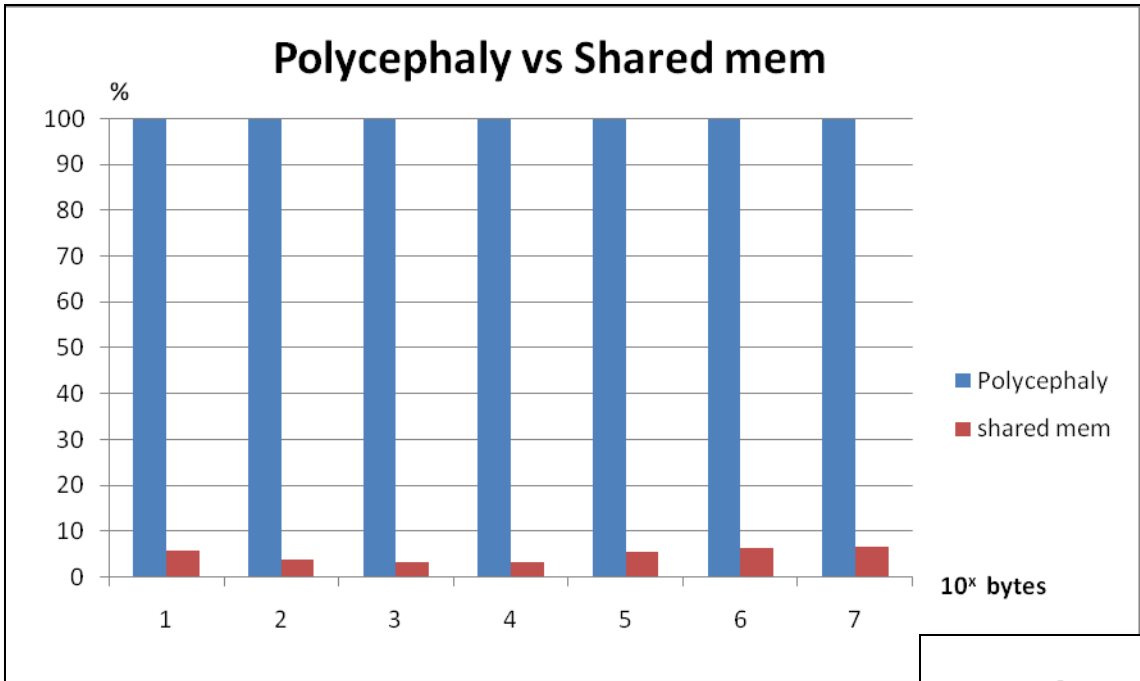
- Polycephaly requests, result via protected shared memory

request bytes	Time (μ s)	factor	request Array of size	Time (μ s)	factor
1	16,434	1.8	1	17,133	2.4
10	15,425	1.6	10	18,281	2.2
100	15,346	1.8	100	17,277	1.8
1,000	15,344	1.8	1,000	18,327	1.5
10,000	15,324	1.5	10,000	18,010	0.7
100,000	15,341	0.7	100,000	23,810	0.1
1,000,000	17,883	0.1	1,000,000	104,050	0.1

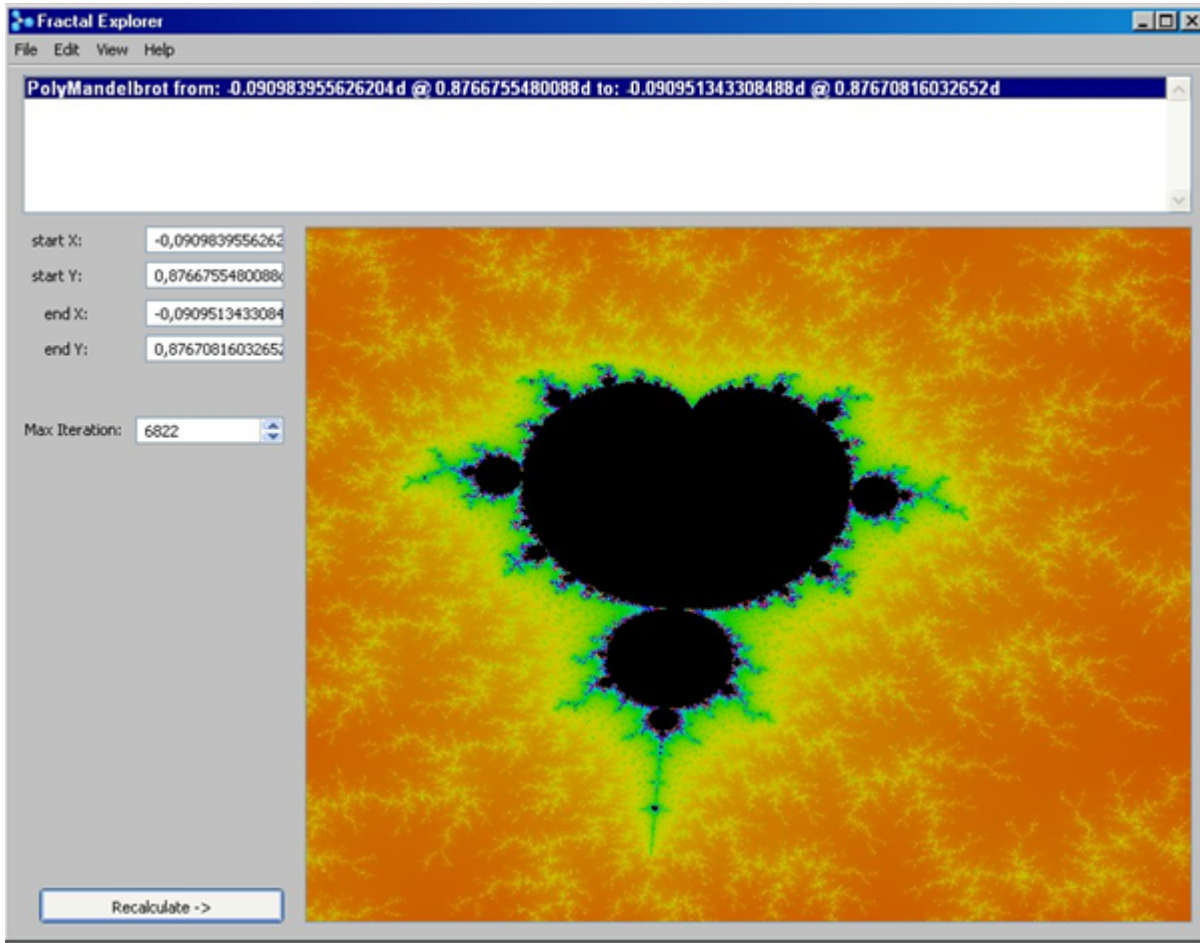
Shared Memory Command And Data Exchange

- Shared memory command, protected write, unprotected read

request Array of size	Time (μ s)	Polycephaly time (μ s)	factor
10	307	8,185	0.04
100	312	9,387	0.03
1,000	397	12,001	0.03
10,000	1,372	25,702	0.05
100,000	10,604	166,141	0.06
1,000,000	99,093	1,518,115	0.07



Fractal Explorer



Shared Memory
data exchange
saves 5 seconds

from a total of 109
seconds...

Conclusions

- Shared Memory can speed up transfer significantly
 - massive data exchange
 - simple data
- Impressive reduction potential of **97%**
- Perspectives for applications that suffer from high load for data exchange.