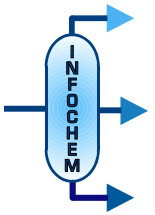
Who Knows Where the Time Goes? Computational overheads of using the thermodynamics interfaces

Richard Szczepanski, Infochem
Jasper van Baten, AmsterCHEM
Tom Williams, PSE

6th CAPE-OPEN European Conference
Munich
2-3 April 2009

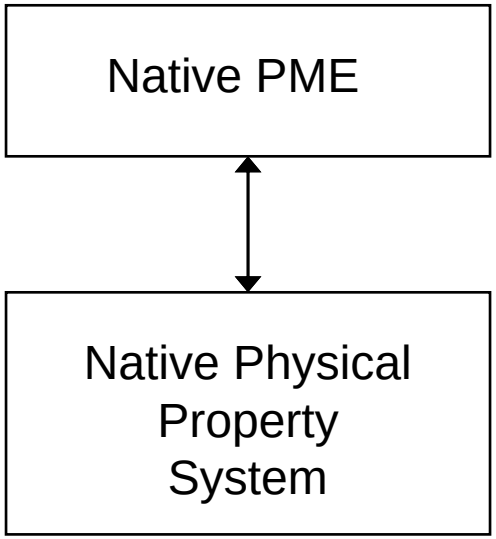
Objectives

- Determine computational overhead of using CO thermo interface compared with a native interface
- Identify how overhead is divided between different software components
- Recommendations for
  - PP developers
  - PME/application developers
  - Future CO specs
- Discussion

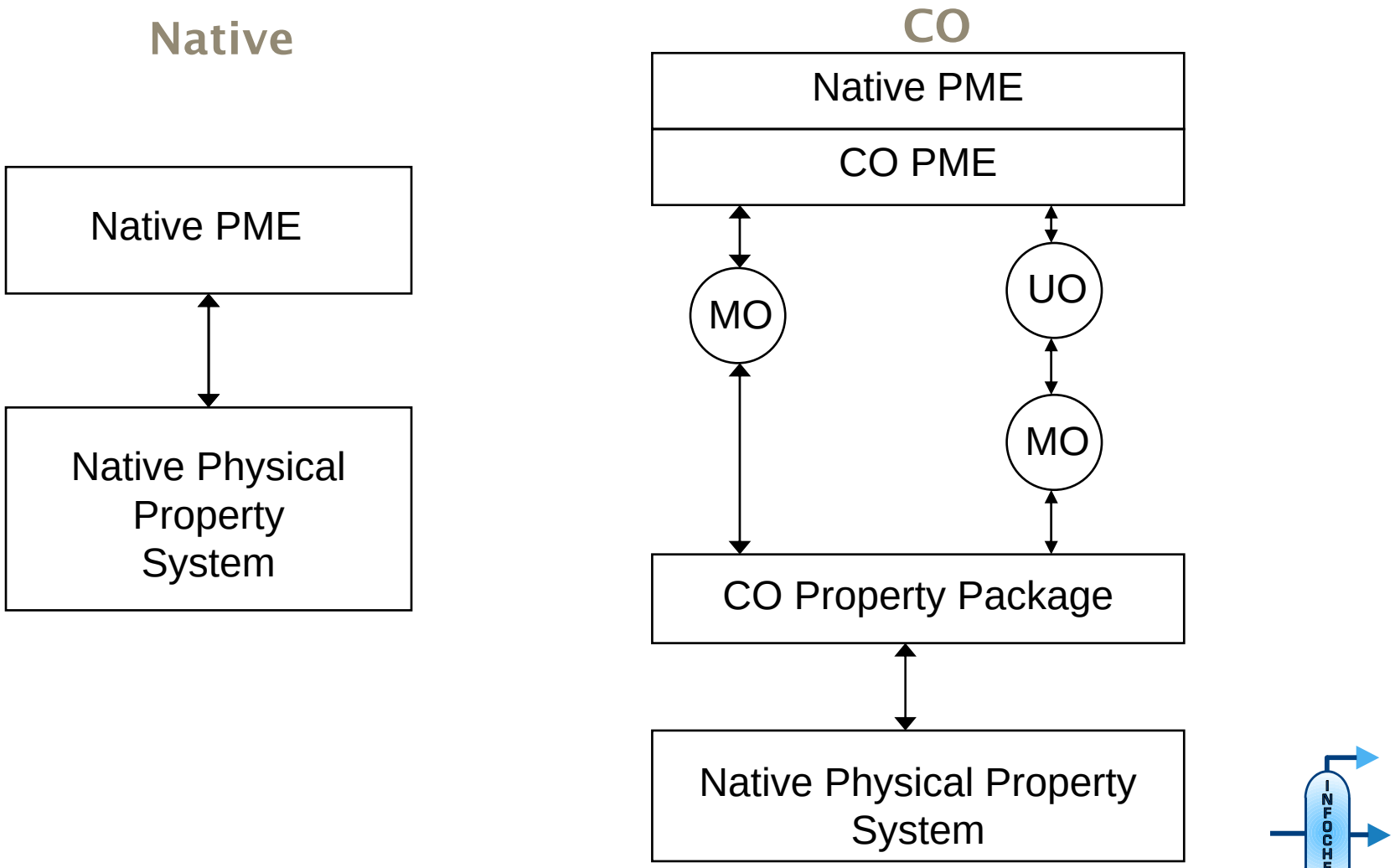


Software components

Native

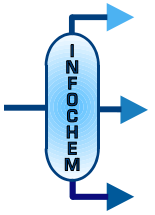


Software components



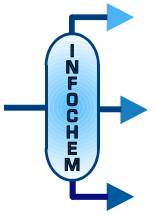
Tools Used

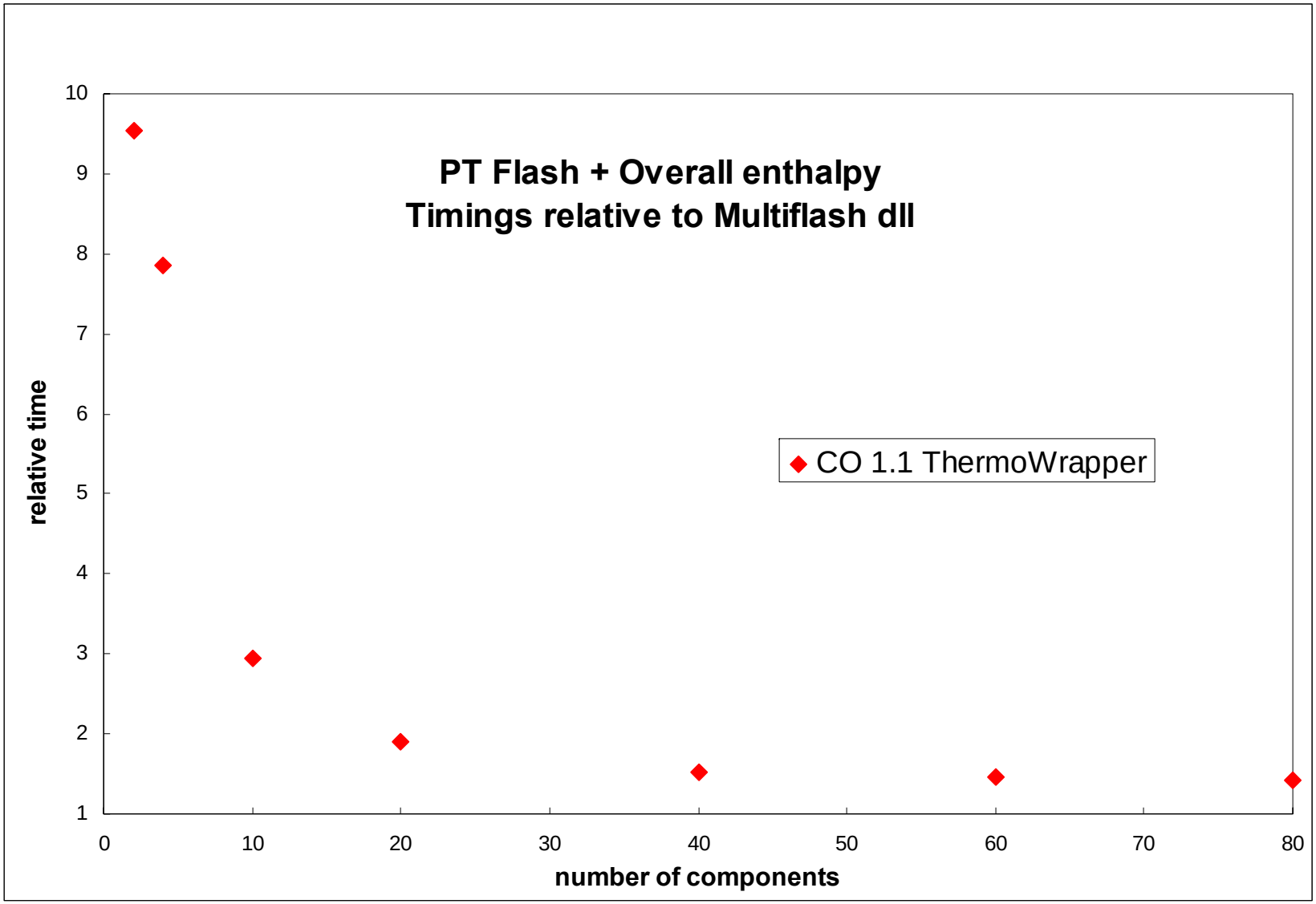
- Native physical property system
  - Multiflash 3.8 dll
  - RKS equation of state
  - Equimolar mixtures of 2 to 80 compounds (hydrocarbons)
  - Calculations over grid of P, T points with large number of repetitions
  - CP time reproducibility: 5 – 10%
- Multiflash CO Property Package
  - Implemented in C++
  - Supports CO thermo 1.0 and 1.1
- ThermoWrapper for CO 1.1
  - Library of Fortran-callable routines for using CO interfaces
  - Provides a Material Object implementation
- CO 1.0 Test Application
  - Custom application using CO 1.0 interface and MO
- Matlab CO Thermo Import (AmsterCHEM)
  - Allows a CO 1.1 PP to be imported into Matlab and used to perform physical property calculations

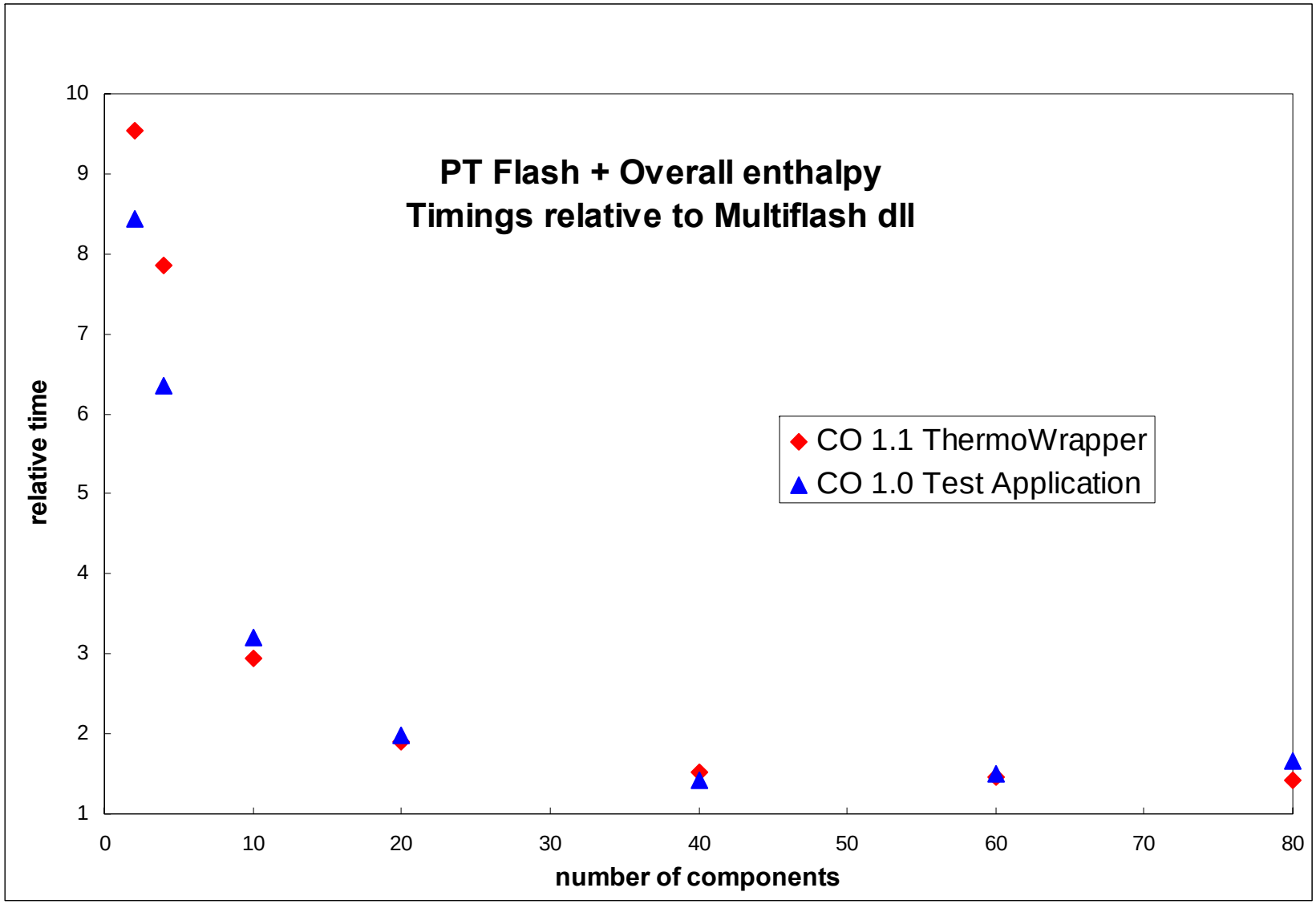


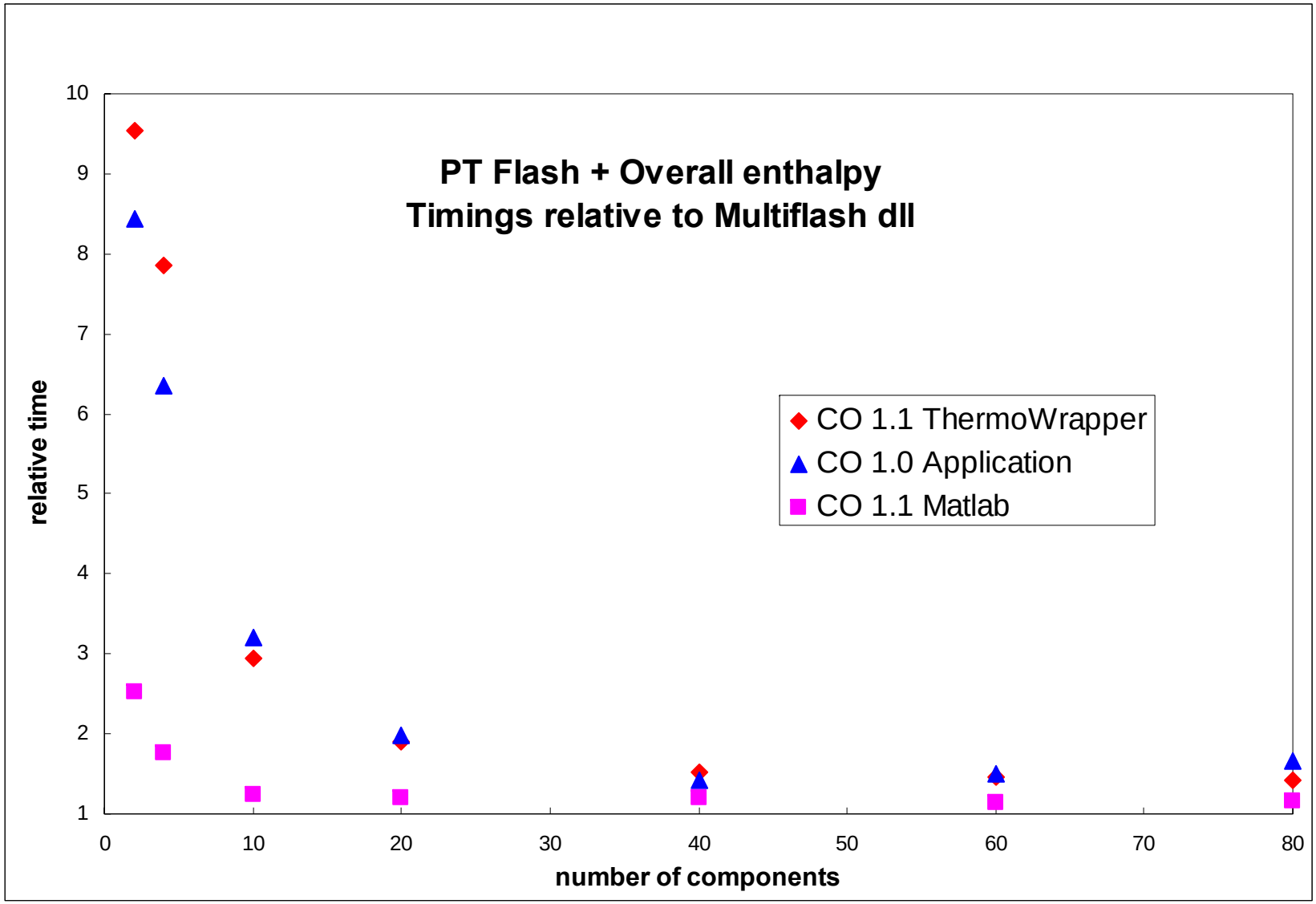
PT Flash & Overall Enthalpy (CO 1.1)

- Application
  - Specify MO to be used: **PP_SetMaterial**
  - Specify list of phases to be considered: **MO_SetPresentPhases**
  - Set overall composition and 2 constraints to define calculation (P and T): **MO_SetOverallProp x 3**
  - Call Property Package: **PP_CalcEquilibrium**
- PP_CalcEquilibrium
  - Get calculation conditions: **MO_GetOverallTPFraction**
  - Get list of *possible* phases for calculation: **MO_GetPresentPhases**
  - **Do (P,T) flash calculation: call Multiflash dll**
  - Set list of phases *actually present* at equilibrium: **MO_SetPresentPhases**
  - Set phase compositions,phase fractions, T, P: **MO_SetSinglePhaseProp x 4NP**
- Application
  - Get list of phases at equilibrium: **MO_GetPresentPhases**
  - Get phase fraction and composition **MO_GetSinglePhaseProp x 2NP**
  - Calculate phase enthalpy **PP_CalcSinglePhaseProp x NP**
  - Get phase enthalpy **MO_GetSinglePhaseProp x NP**



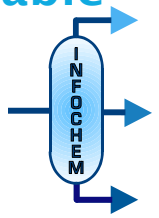

PT Flash + Overall enthalpy
Timings relative to Multiflash dll
relative time
number of components
CO 1.1 ThermoWrapper
10
9
8
7
6
5
4
3
2
1
0
10
20
30
40
50
60
70
80


PT Flash + Overall enthalpy
Timings relative to Multiflash dll
relative time
number of components
CO 1.1 ThermoWrapper
CO 1.0 Test Application
10
9
8
7
6
5
4
3
2
1
0
10
20
30
40
50
60
70
80


PT Flash + Overall enthalpy
Timings relative to Multiflash dll
relative time
number of components
CO 1.1 ThermoWrapper
CO 1.0 Application
CO 1.1 Matlab
10
9
8
7
6
5
4
3
2
1
0
10
20
30
40
50
60
70
80

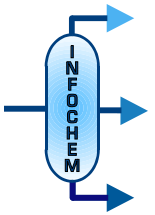
Comments

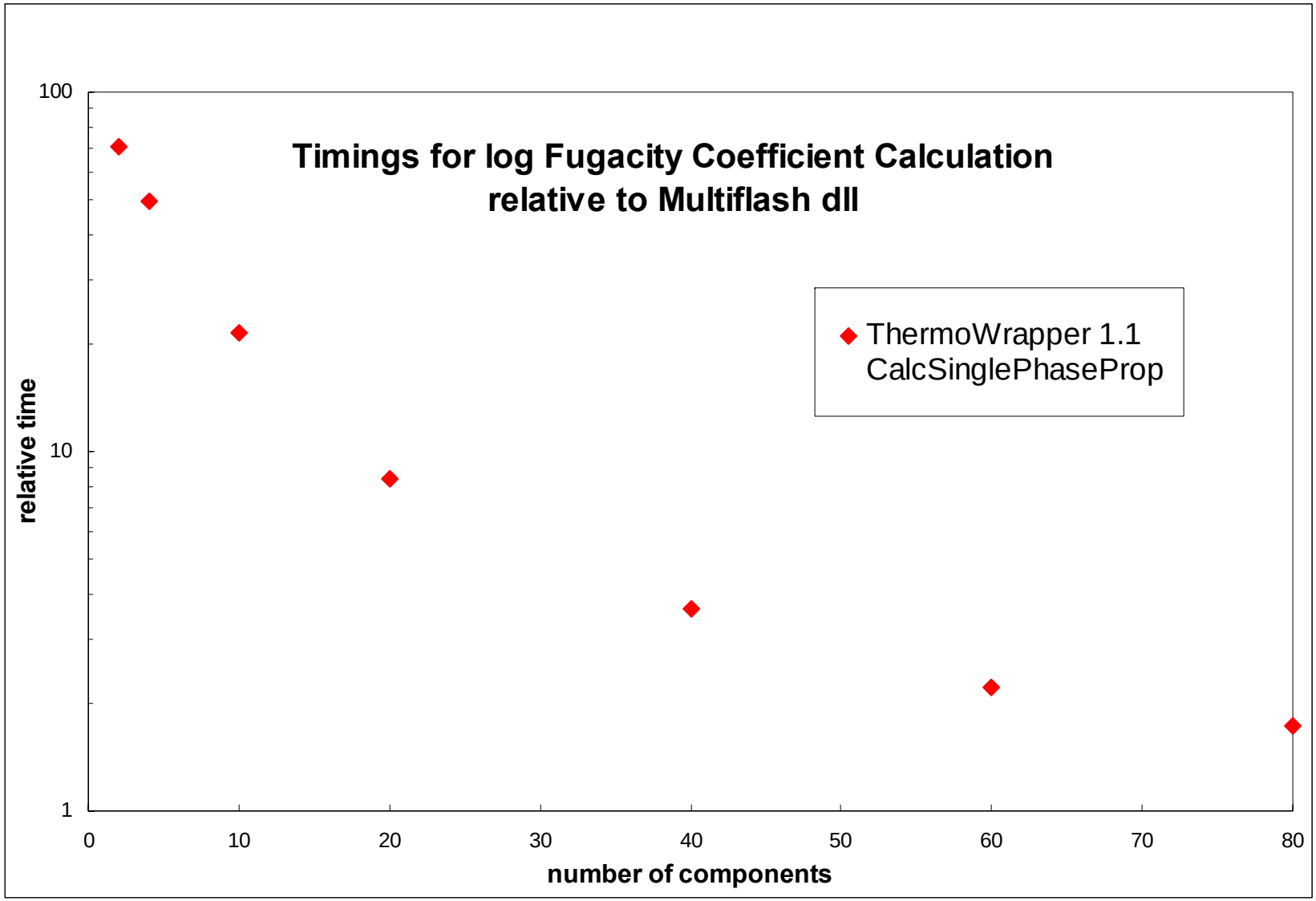
- Applications
  - ThermoWrapper: CO 1.1, Fortran, versatile MO
  - Test application: CO 1.0, C++, versatile MO
  - Matlab: CO 1.1, C++, simple MO
- Differences between 1.0 and 1.1
  - Analysis of compounds in MO
    - For 1.1 is only done when SetMaterial called
    - For 1.0 must be done on every call for a calculation
  - Getting calculation conditions
    - 1.1 has GetTPFraction and GetOverallTPFraction
  - Fewer arguments in 1.1
    - Set/Get: no compound list
    - Calculate: no calcType or MO
- Performance
  - No significant penalty for large number of compounds (>40) whatever the implementation
  - For more complex models overhead will be smaller
  - By appropriate design of MO it is possible to have a reasonable overhead even for small number of compounds



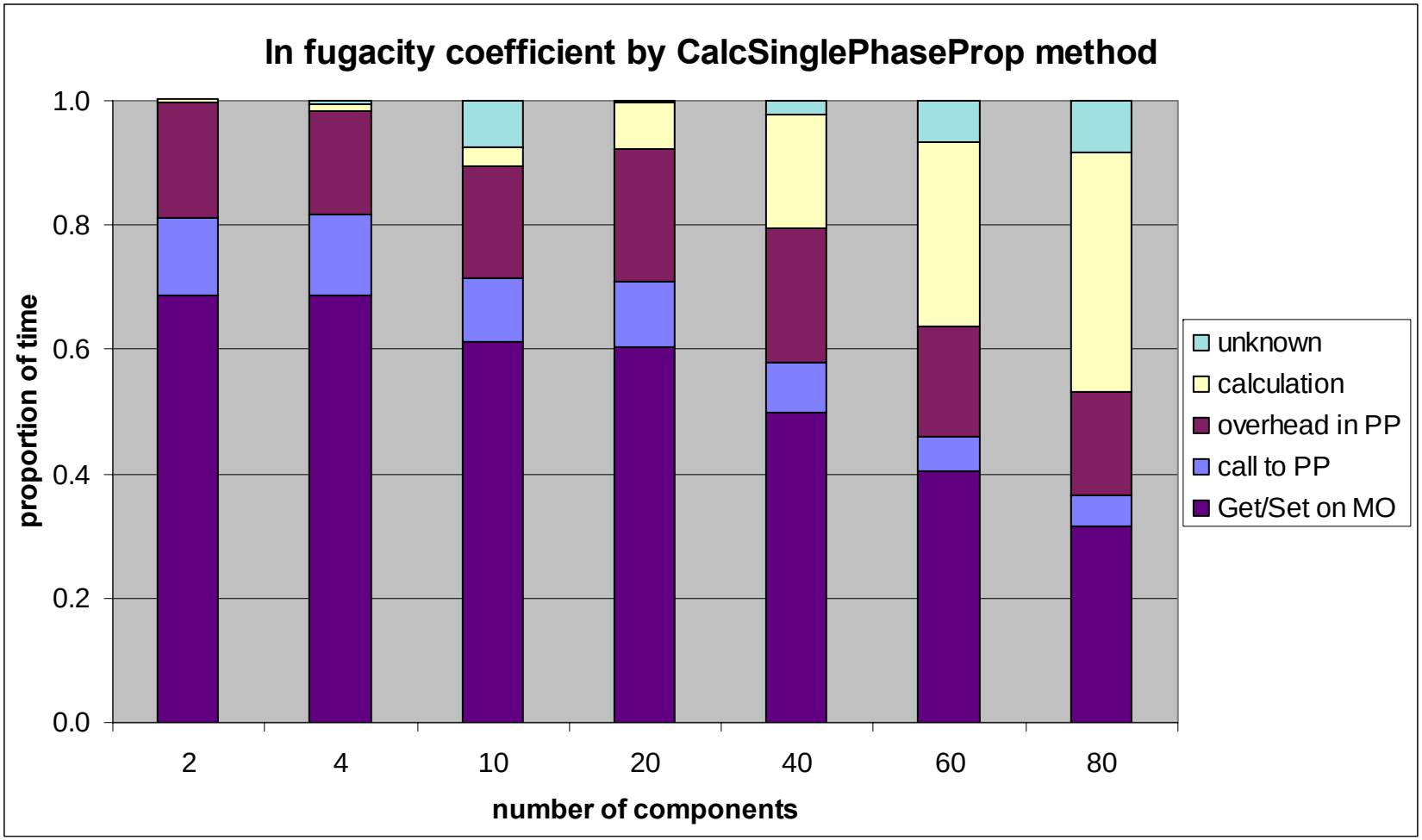
Property Calculation (CalcSinglePhaseProp)

- Application
  - Set P, T and composition of a phase: MO_SetSinglePhaseProp x 3
  - Call to Property package: PP_CalcSinglePhaseProp
- PP_CalcSinglePhaseProp
  - Get P, T and composition of phase: MO_GetTPFraction
  - Calculate property: call Multiflash dll
  - Set property value(s): MO_SetSinglePhaseProp
- Application
  - Get property value(s): MO_GetSinglePhaseProp



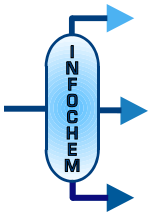

Timings for log Fugacity Coefficient Calculation
relative to Multiflash dll
ThermoWrapper 1.1
CalcSinglePhaseProp
relative time
100
10
1
0
10
20
30
40
50
60
70
80
number of components

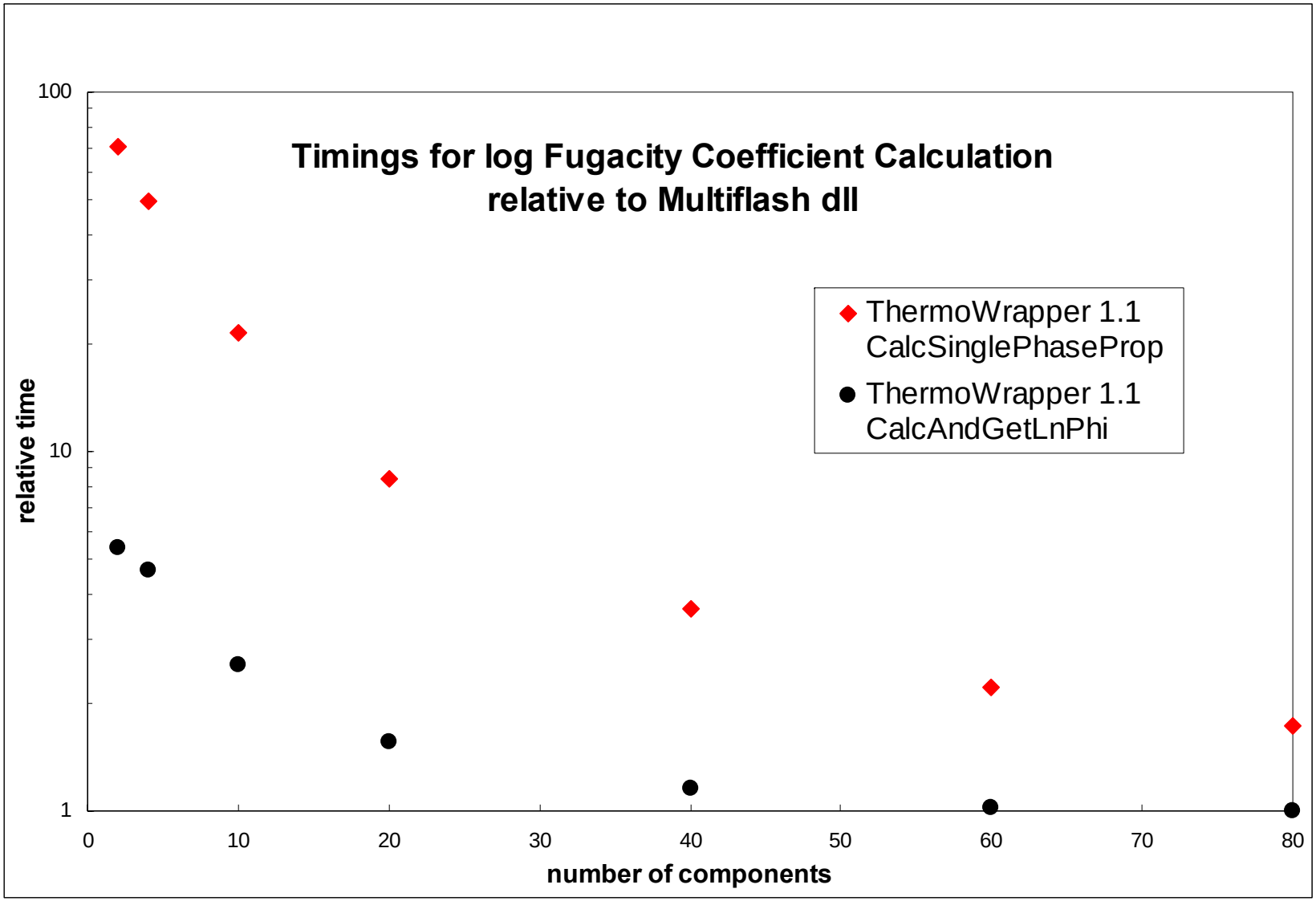
Timings for CalcSinglePhaseProp (VB PP)



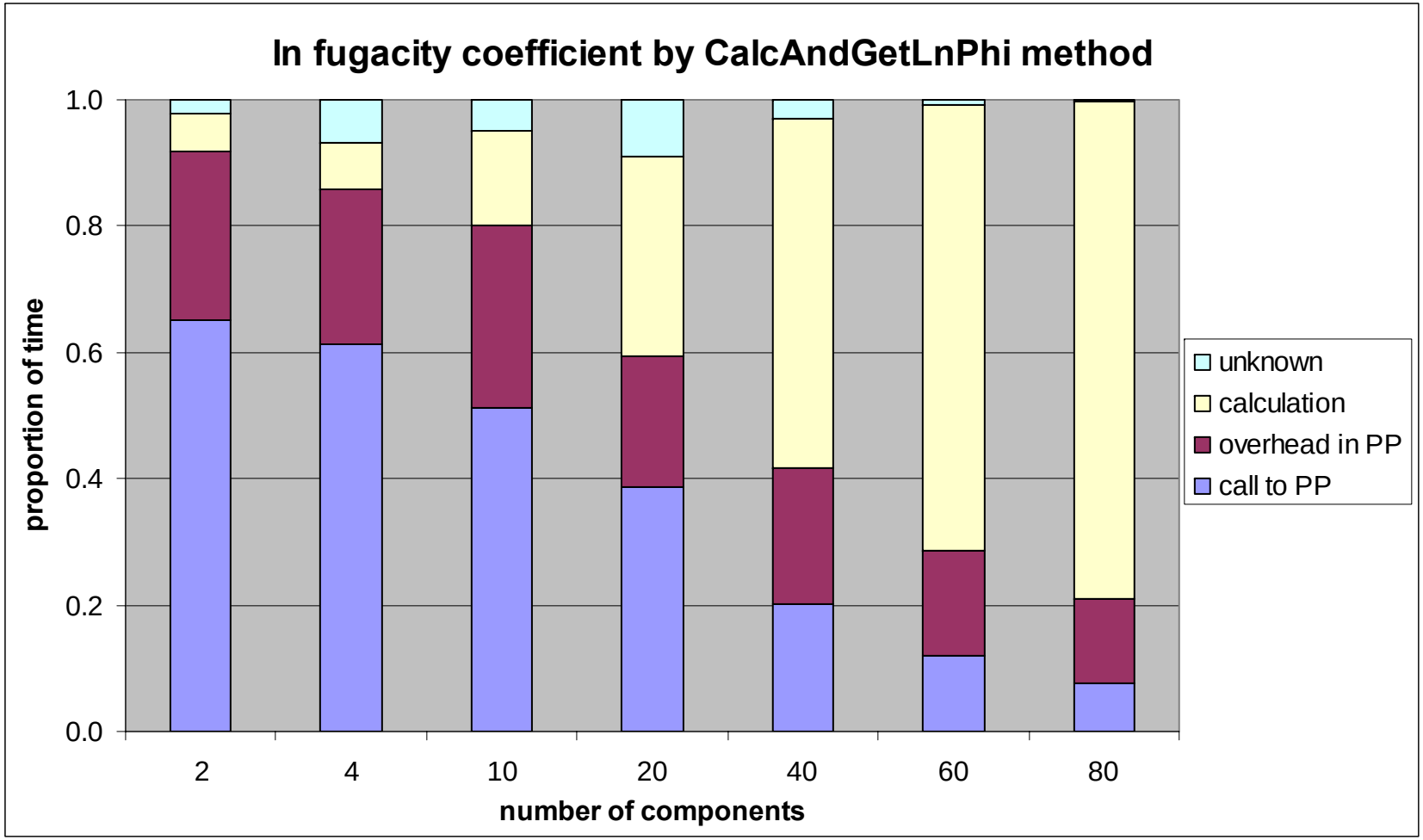
Property Calculation (CalcAndGetLnPhi)

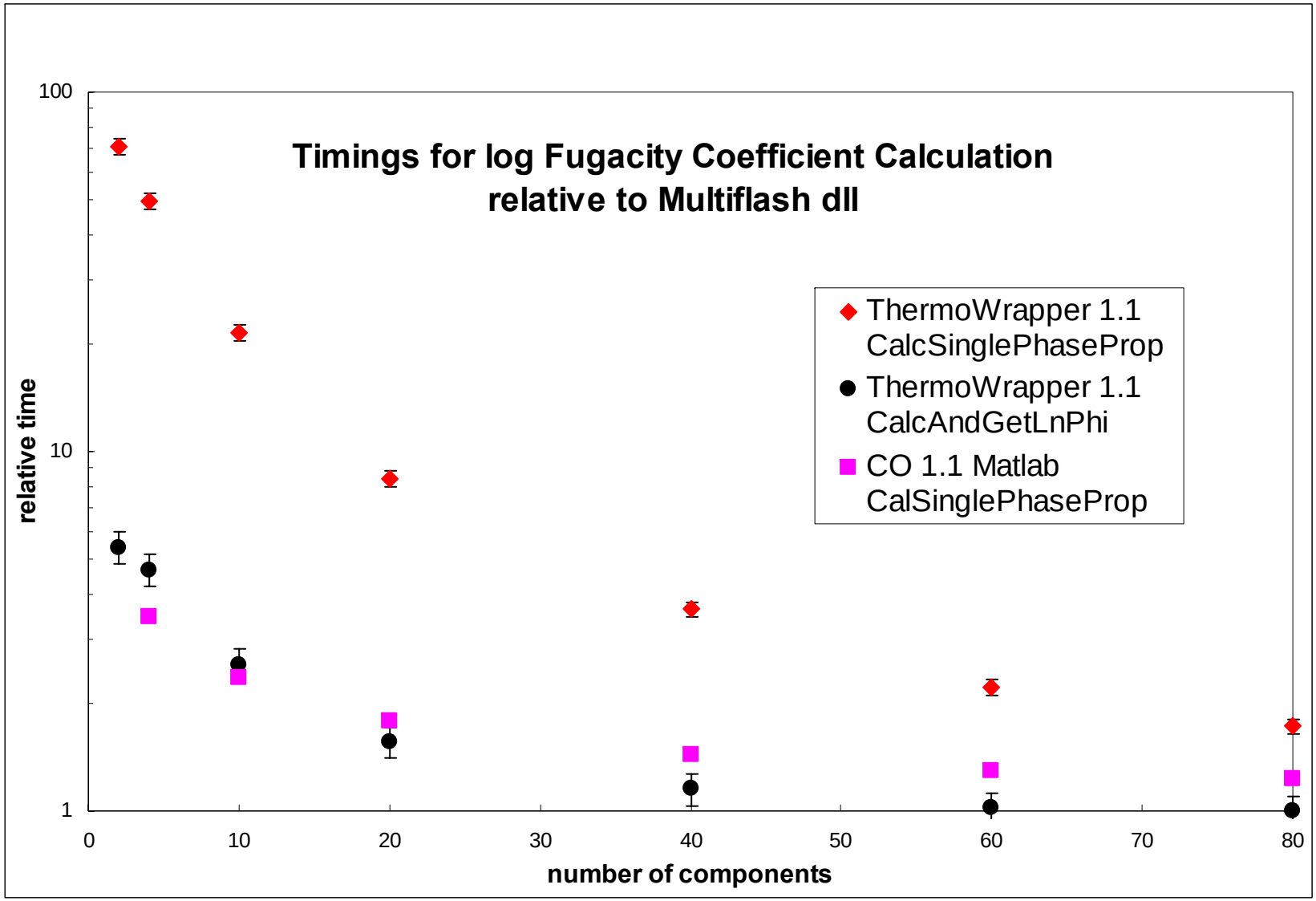
- Method does not use Material Object for communication
- PME
  - Call Property Package:
    PP_CalcAndGetLnPhi(T,P,x,lnφ)
- PP_CalcAndGetLnPhi
  - Type conversions COM to double
  - Calculate lnφ: call Multiflash dll
  - Type conversions double to COM




Timings for log Fugacity Coefficient Calculation
relative to Multiflash dll
ThermoWrapper 1.1
CalcSinglePhaseProp
ThermoWrapper 1.1
CalcAndGetLnPhi
relative time
100
10
1
0
10
20
30
40
50
60
70
80
number of components

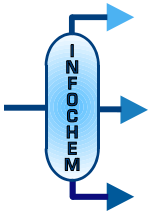
Timings for CalcAndGetLnPhi (VB PP)




Timings for log Fugacity Coefficient Calculation
relative to Multiflash dll
relative time
number of components
100
10
1
0
10
20
30
40
50
60
70
80
ThermoWrapper 1.1
CalcSinglePhaseProp
ThermoWrapper 1.1
CalcAndGetLnPhi
CO 1.1 Matlab
CalSinglePhaseProp

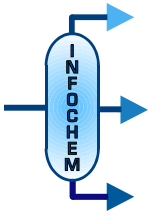
Conclusions and Recommendations 1

- The overhead of using a CO property package can be made quite small: factor of between 1 and 2
- Much of the overhead seems to be associated with the design and operation of the Material Object
  - Competing objectives of efficiency and generality
    - error checking and diagnostics
    - type conversions
    - support of both thermo 1.0 and 1.1 in the same MO
    - PME interaction with MO
  - Attend the short course on implementing MOs
- Thermo 1.1 offers the possibility of more efficient operation
  - SetMaterial
  - GetTPFraction and GetOverallTPFraction
  - Fewer arguments



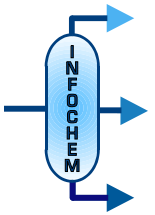
Conclusions and Recommendations 2

- PP Design is also important
  - Re-writing the Multiflash PP in C++ instead of VB reduces CalcSinglePhaseProp time by 25% for small no. of compounds, no difference for large no.
  - Essential to analyze the compound list efficiently and only when SetMaterial is called
- PME design
  - The PME should use SetMaterial only when the MO changes its compound list or compound order (or phase list for flashes)
  - PME owns the MO so can avoid all CO Set/Get calls
- Comparisons of CO and native applications for complete flowsheets would be more realistic for estimating overheads
  - However > 80% of simulation time is typically spent in phys props calculations



Conclusions and Recommendations 3

- Improvements to thermo interfaces
  - methods to identify compounds, phases and properties by integers (handles) rather than strings
  - Direct methods (similar to CalcAndGetLnPhi) for evaluating properties in order to eliminate use of MO as much as possible
  - SetTPFraction and SetOverallTPFraction methods to eliminate multiple references to MO



Who Knows Where the Time Goes? Computational overheads of using the thermodynamics interfaces

Richard Szczepanski, Infochem
Jasper van Baten, AmsterCHEM
Tom Williams, PSE

6th CAPE-OPEN European Conference
Munich
2-3 April 2009