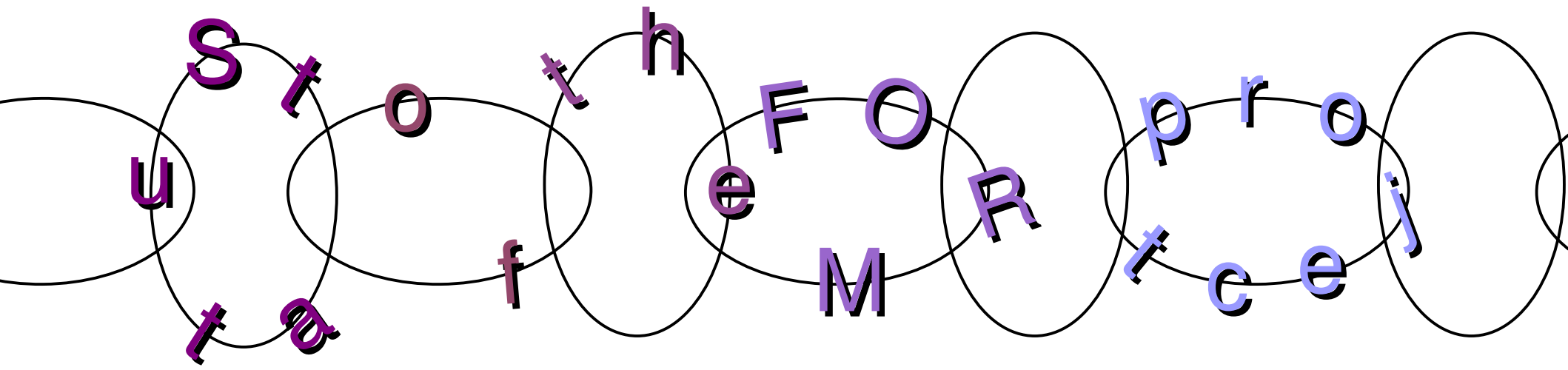
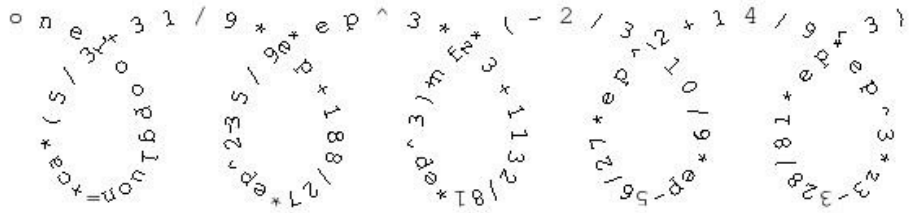



Irina Pushkina

NIKHEF, Amsterdam, The Netherlands

# Status of the FORM project

- Overview of manpower
- New features
  - FORM
  - ParFORM
  - TFORM
- The FORM version of GRACE – the quest for a code simplification
- Future plans



# Overview of manpower

- Jos Vermaseren, NIKHEF, The Netherlands
  - General development of FORM/TFORM
- Irina Pushkina, NIKHEF, The Netherlands
  - Code simplification to facilitate the FORM version of GRACE
- Jens Vollinga (postdoc), NIKHEF, The Netherlands
  - Making FORM the open source
  - FORM/a failsafe system to set up checkpoints from which one can restart after a computer failure
- Jan Kuipers (postdoc), NIKHEF, The Netherlands
  - FORM/incorporation of the polynomial and rational functions
- Misha Tentyukov, University of Karlsruhe, Germany
  - ParFORM project, support for the big FORM runs at Karlsruhe

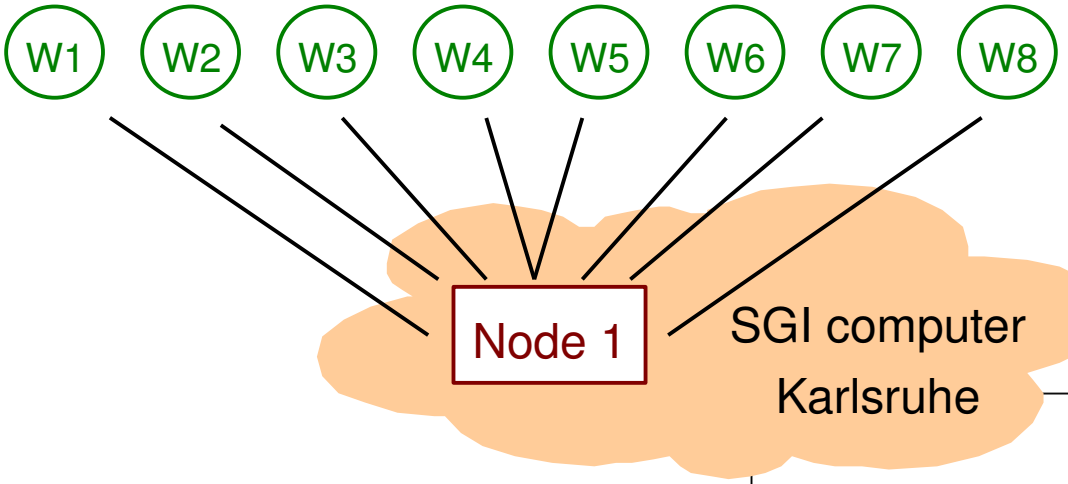
# FORM (version 3.3)

FORM (developed since 1989) as the symbolic manipulation system is heavily optimized for speed and dealing with mathematical expressions of gigantic sizes in an efficient and reliable way.

*ex.* Perturbative Quantum Field Theory, a computation of Feynman diagrams

- The size of the expressions FORM is dealing with has entered the *terabyte* scale.
- The memory management is excellent.
- Implementation of the rules for reduction or transformation of the target expression is flexible and powerful.
- The distribution of the system is in public domain.
  - LINUX/32-bits & LINUX/Opteron
  - Apple/PPC & Apple/Intel
  - Windows (32-bits architecture)

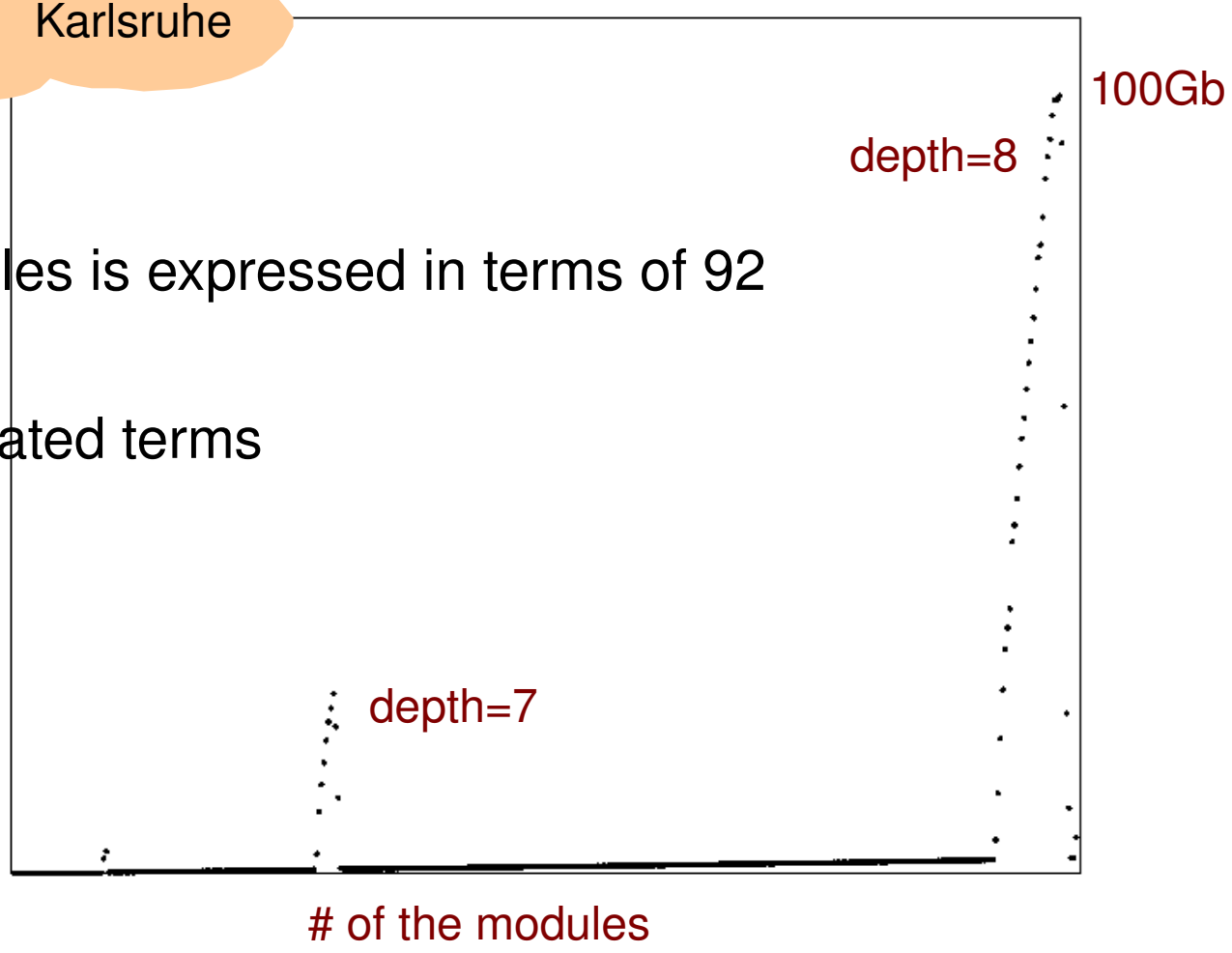
# FORM: example




Example from the field theory

☞ a basis of the weight 28 multiple zeta values

- A system of 971.712 variables is expressed in terms of 92
- 69 days
- 28.700.000.000.000 generated terms



# FORM: new features

- Architecture independent store files (32bit vs 64bit)
- The checkpoints – to protect long-running jobs against system failures
  - ✓ Completely implemented for sequential FORM and TFORM
  - ✓ The documentation is available
- FORM is (almost!) open source ☞ target date is summer 2010
- Documentation
  - ✓ Developers reference manual – the inner workings of FORM
  - ✓ Source code documentation (*doxygen*)
- Repository/access (CVS); webCVS in the future
- Mailing lists for releases and developer coordination
- A web forum

under consideration

# Computers used for the development of FORM

- Laptops using SUSE 10.3, Ubuntu 9.19
- Desktop with QUAD Xeon running Scientific Linux 5.2
- A computer with 4 Opterons running Scientific Linux
- A computer at Karlsruhe:

  SGI 24 nodes each with 8 Xeon cores running SUSE 10
- DELL with 8 hyper-threads at DESY using Scientific Linux 5.4
- Apple computer with 2 Cores

ParFORM is the multi processor version of FORM.

- Operates on clusters (benefits for more than two processors)
- Can handle (almost!) all existing FORM programs
- Requires rather nontrivial installation
- Not all networks are usable
  - ✓ Good results with InfiniBand
- Uses the checkpoint mechanism to protect long-running jobs against system failures

All questions regarding Par(T)FORM are to the next speaker –

Mikhail Tentyukov (Karlsruhe University)

*“Parallel versions of the symbolic manipulation system FORM”*

TFORM as the multi-threaded version of FORM.

- Operates on multi core systems with shared memory (benefits for more than two processors)
- Can handle all existing FORM programs
- The development work involves tuning to more and more cores

J.Vermaseren, M.Tentyukov, hep-ph/0702279

*"The multithreaded version of FORM"*

# ParFORM & TFORM: example

Examples from perturbative QCD:

☞ Massless four-loop integrals and the total cross-section in $e^-$ $e^+$ annihilation into hadrons

☞ Calculation of the order $\alpha_s^4$ contributions to the Bjorken sum rule for polarized electron-nucleon scattering and to the Adler function

☞ Confirmation of the generalized Crewther relation at the order $\alpha_s^4$

- 30.000 5-loop diagrams
- 100 days, 2 clusters (192+120 nodes)
- 10.000.000.000 generated terms on 1TB disk space

P.A.Baikov, K.G.Chetyrkin, J.H.Kuhn, arXiv:1001.3606

# The FORM version of GRACE

Standard models of elementary particles imply that

☞ we have definite Lagrangians and thus we can, in principle, predict any process using perturbation theory.

The interaction Lagrangian is very complex ☞ a technical difficulty

In the perturbative approach, the final four-body processes:

- in the tree level ☞ several tens of diagrams
- in the 1-loop ☞ a few thousand diagrams

GRACE/1-loop is a fully automated system in the standard model (SM) at the 1-loop level.

Full Standard Model

$e^+ e^- \to e^+ e^- \gamma$

Number of diagrams:

Tree level ☞ 32

1-loop ☞ 3456

G.Belanger *et al.*, Phys.Rep., Vol.430 (2006) 117

"*Automatic calculations in high energy physics and Grace at one-loop*"

# GRACE/FORM/1-loop system

- Generation of the 1-loop and the tree diagrams
- Construction of the product graphs

GRACE
the automatic computation program of 1-loop Feynman diagrams

- Procession of the generated graphs through a series of symbolic operations – algebraic calculation of the product of a loop diagram with a tree diagram

  (the shift of loop momentum, the trace calculation of $\gamma$-matrices, the extraction of numerical coefficients for the loop integral and so forth)

GRCFORM

- Writing the resulting formulas as a Fortran routine
  (each diagram is the Fortran file)

FORM
the symbolic manipulation system

# Future plans

Typical diagrams run in a few seconds

☞ the improvement of the quality of the Fortran code is in need!

Example:

| | | |
|---|---|---|
| e e -> e e $\gamma$ | : | 1.4Gbyte of Fortran (at the moment) |
| e e -> $\mu^{+}\bar{\nu}_{\mu}\tau$ d | : | O(50) Gbytes of Fortran code |

Last remark regarding the license condition of FORM:

☞ If you use FORM for a scientific publication, PLEASE refer to it!