Computing Challenges in Information Systems

Panelists:

Glenn Luecke - Iowa State University

Panos Nasiopoulos - University of British Columbia

Thomas M. Prinz - Friedrich Schiller University Jena

Philippe Marchildon - École des Sciences de la Gestion (Montréal)

Glenn Luecke

- **Current Positions**
  - Professor of Mathematics (Iowa State University)
  - Adjunct Professor of Computer Engineering
  - Senior Member of the ACM
  - Director of High Performance Computing Education

- **Topic:** The Challenges of Tools for Exascale Computing

- **Short Bio**
  - Professor Luecke's High Performance Computing (HPC) Group is involved in research in the areas of parallel algorithms, parallel tools, and the evaluation of high performance computing systems. He has had over 60 graduate students, visiting scholars, and post doctoral students. Professor Luecke has been awarded over $6 million in contracts and grants performing a wide variety of research activities for the following organizations: The United States Department of Defense, Cray Inc., Sun Microsystems, Silicon Graphics Inc., NASA via Computer Science Corporation, IBM, Visual Numerics and Hitachi Data System

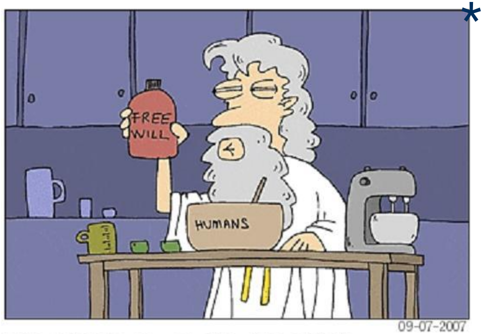
Thomas M. Prinz

- **Current Positions**
  - Chair of Methodology and Evaluation Research at Friedrich Schiller University Jena (Germany)

- **Topic:** Intelligent versus intelligently built services.

- **Short Bio**
  - Thomas M. Prinz, Phd, is a research assistant and software architect in the Course Evaluation Service at the Friedrich-Schiller-University Jena, Germany. Since 2015, he is responsible for the evaluation software used on the entire university and for the software architecture of the tool Coast. His research interests include the theory of compiler construction, the user-centered verification of business processes, human machine interaction, multi variate analysis methods, and the theoretical background of empirical studies.

Intelligent versus intelligently built services.



EVER WONDER IF GOD HAD ANY SECOND THOUGHTS?

Dr. Thomas M. Prinz

Course Evaluation Service
Friedrich Schiller University Jena, Germany

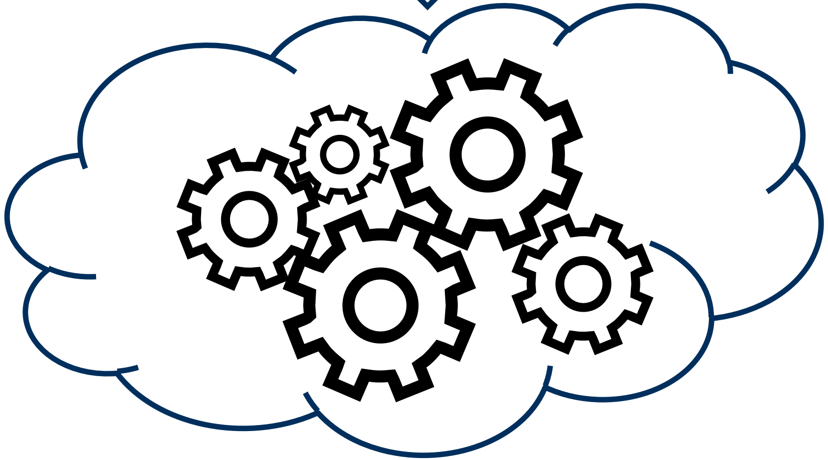
*http://blog.eternalvigilance.me/2013/11/no-free-will-no-moral-responsibility-william-lane-craig/

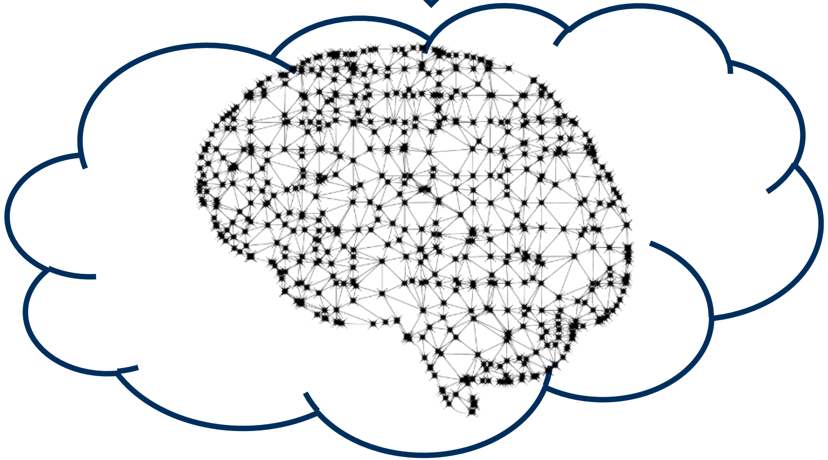
Intelligent versus intelligently built services.

Input

x intelligently built service

x intelligent service





Output

y

?

Predestination: Determinism

Free will: Undeterminism

Intelligent versus intelligently built services.

Are we able to guarantee the success of software project?

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Standish Group 2015 Chaos Report, https://www.infoq.com/articles/standish-chaos-2015

Intelligent versus intelligently built services.

Are we able to guarantee proper functionality for deterministic services?

Projects	Version	Lines of Code	No. Files	Bug Count	Duration
Ant	1.8.4	254,431	1,167	2,567	5.18 years
Maven	3.05	51,651	346	2,945	10.66 years
CMake	2.8.10	406,715	1,104	5,192	9.72 years
QMake	5.0.1	33,583	53	844	6.96 years

Projects	# Bug Per kLOC	# Bug Per File	# Bug Per Year
Ant	10.09 bugs/kLOC	2.20 bugs/file	495.56 bugs/year
Maven	57.02 bugs/kLOC	8.51 bugs/file	276.27 bugs/year
CMake	12.77 bugs/kLOC	4.70 bugs/file	534.16 bugs/year
QMake	25.13 bugs/kLOC	15.92 bugs/file	121.26 bugs/year

Xia, Xin; Zhou, Xiaozhen; Lo, David; Zhao, Xiaogiong:
An Empirical Study of Bugs in Software Build Systems.
13th International Conference on Quality Software (QSIC), Najing, 2013, 29-30 July 2013, pp. 200 - 203

Intelligent versus intelligently built services.

How do we know that the system does its work
when we are not even able to decide it,
when the system is deterministic?

FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Philippe Marchildon

- **Current Positions**
  - Assistant professor in the Département de management et technology at the École des Sciences de la Gestion
- **Topic:** Moving from one information technology (IT) to the next
- **Short Bio**
  - Philippe Marchildon is an Assistant professor in the Department of Management and Technology at École des Sciences de la Gestion de l'Université du Québec à Montréal (ESG-UQAM). Over the last few years, he had the opportunity to teach several classes at both the undergraduate and graduate level and to publish his research results in important journals and conferences in the field of information systems. When he teaches, he always put student first and uses several complementary pedagogical approaches (i.e., lectures, case study, laboratories). His main research interests include: IT adoption and post-adoption, creation and co-creation of IT value, enterprise architecture, theory building and barriers to change (i.e., path dependence, structural inertia, imprinting).

Moving from one IT to the next!

- We are in a context of post-adoption
  - Everyone has now embraced information technology

- As such, we face a new challenge
  - Before, it was about convincing people that the digital space was better
  - Now, it's more about getting people to move from one IT to next

- Information technology is a special kind of resource
  - Decreasing versus increasing returns

Vicious or virtuous IT cycles

- Coordination effects
  - Coordination effects are based on the self-reinforcing logic of coordination, which rests on the benefits of rule-guided behavior. According to this logic, the more actors adopt and apply a8common standard, rule, or routine, the more efficient the interaction among these actors will be.
- Complementarity effects
  - Complementarity effects are based on the self-reinforcing logic of complementarity, which rests on the synergy resulting from the interaction of two or more discrete but interrelated resources, rules, and/or practices (Schreyögg and Sydow, 2011; Sydow et al., 2009). When this logic prevails, the benefits provided by a set of elements (i.e., resources, rules, and/or practices) are greater than the sum of the benefits provided by each of the elements in the set alone (Schreyögg and Sydow, 2011; Sydow et al., 2009)
- Learning effects
  - Learning effects are based on the self-reinforcing logic of learning, which rests on the fact that the more often an operation is performed, the more efficiency will be gained in subsequent iterations (Sydow et al., 2009).
- Adaptive expectation effects
  - Adaptive expectation effects are based on the self-reinforcing logic of adaptive expectation, which rests on the interactive building of preferences across actors (Schreyögg and Sydow, 2011; Sydow et al., 2009).

How to turn things around

Take advantage of existing cycles

or

Deploy resources to develop new ones

Panos Nasiopoulos

- **Current Positions**
  - Professor with the Department of Electrical and Computer Engineering at the University of British Columbia
- **Topic:** Merging digital video technologies - Light Field and HDR
- **Short Bio**
  - Panos Nasiopoulos earned his bachelor's degree in physics from the Aristotle University of Thessaloniki (1980), Greece, and his bachelor's (1985), master's (1988), and Ph.D. (1994) degrees in electrical and computer engineering from the University of British Columbia (UBC), Canada. He is a professor with the Department of Electrical and Computer Engineering and the former Director of ICICS and the Master of Software Systems at UBC. Before joining UBC, he was the President of Daikin Comtec US and Executive Vice President of Sonic Solutions. He is a registered professional engineer in British Columbia, a fellow of the Canadian Academy of Engineering, and has been an active member of the Standards Council of Canada, MPEG, ACM and IEEE.

Thoughts Regarding Tools for Exascale Computing

Glenn Luecke
Professor and Director of HPC Education and Training
Iowa State University, Ames, Iowa, USA
Barcelona, Spain, February 20, 2018

Tool Purpose and Challenges

- Purpose of tools: aid in the development of fast, correctly running applications.
- Today's HPC machines are becoming more and more complex due to many cache levels and the number of cores/node getting larger and larger, vector operations, accelerators attached to NUMA nodes, hybrid program parallelization with MPI, OpenMP, CUDA, OpenACC, Co-array Fortran, UPC
- It is estimated that Exascale HPC machines will have at least 500,000 cores and likely will have a hybrid architecture and procesors speed may depend on temperature.
- Tool design depends on both the underlying hardware and the system software being used.

Performance Analysis, Debugging and Correctness Tool Issues

- scalability
- hybrid architectures
- fault tolerance
- complex memory hierarchies
- multiple languages
- huge data
- multiple hardware
- multiple system software
- power constraints
- easy-to-use (includes good documentation)

Current Tool Design

- Tools present information to the programmer and the programmer decides how to change his application code.
- The information the tool presents is often complex making it difficult for the programmer to know what code changes are needed to fix the problem.
- Often an expert is needed to understand the data presented and know what to do to change his application code.
- Debugging often can be done on small problems, but not always.
- Ongoing support is needed.

Some References

- ASCR Tools Challenges for Exascale Computing, October 2011
- 6th Workshop on Extreme-Scale Programming Tools, June 2017
- First Workshop on Software Challenges to Exascale Computing, Nov 2017
- HPCToolkit
- Intel's Parallel Studio XE
- Allinea's Performance Reports, MAP and DDT
- KOJAK: A Tool Set for Automatic Performance Analysis of Parallel Programs (Juelich, Univ of Tenn) 2003
- Vampir trace tool (U of Dresden) last release Nov 2017.
- The TAU Performance System (U of Oregon)

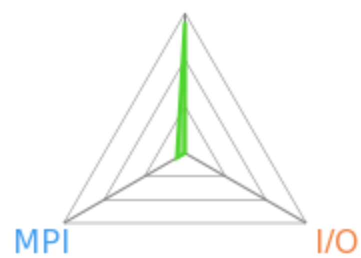
Allinea and Intel Tools

- Allinea
  - Performance Reports
  - MAP
  - DDT (debugging)
  - Remote Desktop Client => huge improvement in GUI perf.
- Intel Parallel Studio XE 16.0
  - Profile Function or Loop Execution Time
  - Trace Analyzer and Collector
  - Vtune Amplifier

Conclusions/Thoughts

- Exciting area of current research!
- Performance & debugging tools need to tell application developers where the problems are and how to fix them.
- AI may be helpful (Allinea's Peformance Report is a step in this direction, see next page
- Easy-to-use and scalable
- Single tool for performance
- Single tool for debugging
- Good documentation is essential
- Remote desktop is needed
- Ongoing support is needed

Command:	mpirun -f /var/spool/torque/aux//145014.share -perhost 1 -np 8 ./episnp-mpi-openmp-perf
Resources:	4 nodes (16 physical, 16 logical cores per node)
Memory:	126 GB per node
Tasks:	8 processes, OMP_NUM_THREADS was 8
Machine:	node196
Start time:	Tue Oct 27 02:50:19 2015
Total time:	27068 seconds (451 minutes)
Full path:	/data013/reecygroup/episnp
Input file:	
Notes:	



Summary: episnp-mpi-openmp-perf is Compute-bound in this configuration

Compute	93.4%	Time spent running application code. High values are usually good. This is **very high**; check the CPU performance section for advice.
MPI	6.6%	Time spent in MPI calls. High values are usually bad. This is **very low**; this code may benefit from a higher process count.
I/O	0.0%	Time spent in filesystem I/O. High values are usually bad. This is **negligible**; there's no need to investigate I/O performance.

This application run was Compute-bound. A breakdown of this time and advice for investigating further is in the CPU section below.

As very little time is spent in MPI calls, this code may also benefit from running at larger scales.