

Curriculum Vitae

Anton Burtsev

Contact

50 S. Central Campus Dr., Rm 3190
Salt Lake City, Utah 84112
Tel: +1 801 581 3012
Email: aburtsev@flux.utah.edu
WWW: <http://www.cs.utah.edu/~aburtsev>

Research interests

Operating systems: new abstractions for structuring operating systems, virtualization, microkernels, distribution models, persistence, threading models, IPC mechanisms, lock-free synchronisation, scalability, multicores.
Distributed systems debugging: replay debugging of large distributed systems, delta debugging, model checking.
Distributed operating systems: networking and datacenter operating systems.
Security: distributed and decentralized access control models, access control in operating systems.
Languages: language abstractions for operating systems, operating system interface, domain specific languages.

Education

2005 – present

Ph.D. student at the University of Utah, supervision: Jay Lepreau, GPA: 4.00.
Dissertation area: *Replay debugging of large distributed systems*.

2002 – 2005

Ph.D. student at the Institute for Applied System Analysis at National Technical University of Ukraine “Kyiv Polytechnic Institute”, supervision: Yuriy Timoshenko, GPA: 4.00.
Dissertation area: *Distributed operating system architecture based on replicated objects*. (Wasn't completed)

2000 - 2002

M.Sc. diploma with honors in Applied Mathematics at the Institute for Applied System Analysis at National Technical University of Ukraine “Kyiv Polytechnic Institute”, supervision: Yuriy Timoshenko, GPA 4.00.
Theses: *Distributed operating system architecture based on distributed shared objects*.

1996 - 2000

B.Sc. diploma with honors in Applied Mathematics at the Institute for Applied System Analysis at National Technical University of Ukraine “Kyiv Polytechnic Institute”, supervision: Alexander Selin, GPA: 3.95.
Theses: *Intelligent software agents*.

Research activity

2005 – present (University of Utah)

Time-travel for large distributed systems

My primary research work is development of a replay debugging environment for large distributed systems. This research is a part of the Emulab Network Testbed – a public testbed where researchers can run networking and distributed system experiments in a realistic and controlled environment. Naturally, any experiment in Emulab requires debugging support. Thus ability to replay the execution of an entire experimental network is a powerful aid in preparing and analysing experiments. We develop a time-travel system, which runs along with an experiment launched in Emulab and captures original execution of the entire network. In case of an error, a researcher is able to rollback execution of the network to a point before the crash and inspect system's state or debug it with a traditional step-by-step debugger. We presented our initial results in the poster session at the NSDI'06 conference Abstract [[PDF](#)], poster [[PDF](#)].

2008 – present (University of Utah)

Aggressive server consolidation through pageable virtual machines

We extend the Xen virtual machine monitor with the ability to host a hundred of virtual machines on a single physical node. Similarly to a demand paging of virtual memory, we page out idle virtual machines making them available on demand. Paging is transparent. An idle virtual machine remains fully operational. It is able to respond to external events with a delay comparable to a delay under a medium load.

To achieve desired degree of consolidation, we identify and leave in memory only a minimal working set of pages

required to maintain the illusion of running VM and respond to external events. To keep the number of active pages small without harming performance dramatically, we build a correspondence between every event and its working set. Reducing a working set further, we implement a copy-on-write page sharing across virtual machines running on the same host. To decrease resources occupied by a virtual machine's file system, we implement a copy-on-write storage and golden imaging. Abstract [[PDF](#)], poster [[PDF](#)].

2007 – present (University of Utah)

Statefull Swapout (a true time-sharing for computer networks)

The project aims to extend the Emulab Network Testbed with a true time-sharing scheduling of network experiments. Emulab includes more than 350 PC nodes. Some 1500 users run experiments of size 1-300 nodes, on a first-come first-served basis. Demand far exceeds capacity, so Emulab provides a weak form of time-sharing. We would like to schedule Emulab experiments similarly to how OS schedules its processes. The Emulab scheduler could preempt low-priority or idle experiments, save the full state of entire network, and replace it with other experiments. The goal is that swapout/swapin must be transparent to the experiment, and except for scheduling delays, transparent to users. In the context of Emulab, this means that we are able to preempt large long running experiments without compromising the correctness of experimental results. We presented our initial results in the poster session at the NSDI'07 conference. Abstract [[PDF](#)], poster [[PDF](#)].

2004 August – November (Sydney, Australia)

Research internship

Implementation of μ ITRON real-time embedded operating system on top of L4 microkernel

Research group of Prof. Gernot Heiser. National ICT Australia, Embedded, Real-Time and Operating Systems Research Group, University of New South Wales.

Supervision: Gernot Heiser, Ihor Kuz

μ ITRON is a specification of a real-time embedded operating system kernel. Being minimalist and powerful it became a de-facto standard on Japanese embedded system market. Implementation of a μ ITRON compatible kernel on top of L4 was a good test of both completeness and flexibility of the L4 API and of its potential applicability to development of embedded systems (source code [[link](#)], informal technical report [[PDF](#)]).

2000 – 2005

Postgraduate research (Kiev, Ukraine)

E1 distributed operating system

E1 distributed operating system project is aimed to provide a reliable software platform for distributed applications. State and functionality of all operating system components is encapsulated by distributed objects. A complete or partial copy of distributed object's state is placed on every network node, where the object is accessed. Consistency of object's state is ensured by replication algorithms (project web page: <http://www.e1os.org>).

1999 – 2000

Undergraduate research (Kiev, Ukraine)

An agent operating system

The project aimed on investigating applicability of agent oriented programming and artificial intelligence to construction of operating systems capable to operate in complex reactive environments. All operating system components were designed as intelligent software agents cooperating to achieve common goals. Agent's behavior was described in terms of intensional logic.

Internships

2004 August – November (Sydney, Australia)

Research internship

Embedded, Real-Time and Operating Systems Research Group, University of New South Wales, Sydney Australia.

Awarded scholarships

March 2008. USENIX Student Grant to attend the NSDI'08 conference.

October 2007. ACM Student Grant to attend the SOSP'07 conference.

April 2007. USENIX Student Grant to attend the HotOS'07 workshop.

Nov 2006. USENIX Student Grant to attend the OSDI'06 conference.

Sep 2004. NICTA International Internship Scholarship.

Relevant educational projects**2008 – present (University of Utah)****Source code comprehension, analysis and compilation through syntax tree transformations.**

We use Parsing Expression Grammars (PEG) to parse source code and generate a syntax tree of a large software system (e.g. Linux kernel). We use syntax tree in two ways. First, we implement an intelligent code browsing, cross-referencing and visualizing tool, which, we hope, simplifies understanding of the source code for a kernel programmer. For example, using syntax tree information, we are able to precisely identify all places in a kernel code where a field of a particular structure is referenced by pointer (e.g. “sk_buff->next”). Second, using the syntax tree we implement a source to source compilation of a kernel code. For example, we hope to automatically identify all places, where guest VM receives non-deterministic events from a VMM and wrap them with a logging code for time-travel.

1998-1999 (Kiev, Ukraine)

Implementation of a C language compiler with educational purposes within the "Computer Systems" class.

1997-1998 (Kiev, Ukraine)

Implementation of a small 3D game engine for a DOOM-like computer game within the "Programming" class.

Teaching and educational experience**2007-2008 (University of Utah)****Computer Systems Lunch Seminar**

Web page: <http://www.cs.utah.edu/classes/csl-sem>

Organization of a seminar, where students and faculty members can keep track of most recent advances in systems research.

2002 – 2004 (Kiev, Ukraine)**Teaching assistant**

National Technical University of Ukraine

Classes: “Introduction to Digital Circuits”, “Computer Systems”.

Work experience**2005 – present****Research assistant****Flux Research Group, School of Computing, University of Utah.**

Research in the areas of distributed systems debugging, transparent checkpointing, and time-sharing of computer networks. Work on implementing replay debugging for Emulab Network Testbed. Hacking Xen virtual machine monitor to extend it with deterministic replay capabilities.

1999 - 2005**Software developer (part-time position, Kiev, Ukraine)****WestGate Software Security Inc. (www.westgate-usa.com)**

Design and development of software projects in a field of hardware-based software protection and security. The list of completed industrial projects can be found at <http://www.cs.utah.edu/~aburtsev/industrial.html>

Areas of expertise

System and kernel programming, operating systems programming.

Web-based distributed applications (server side programming, XML technologies, database programming).

Hardware-based and cryptographic data and software protection.

Public key infrastructure (PKI) cryptography.

Skills**Programming languages**

C/C++, Haskell, Java, SQL, PHP, Python, IA-32 and IA-64 assemblers, OZ, UNIX Shell, Visual Basic, Java Script.

Operating systems

Xen: microkernel, dom0/domU kernel and python control layer hacking. **L4 microkernel:** operating system

programming, microkernel hacking. **FreeBSD and Linux**: application, system and kernel programming, CGI, XML technologies, Java Servlets, database programming (PostgreSQL, Interbase), system administration. **Windows**: application programming, COM/DCOM, Microsoft Media Technologies, Microsoft XML technologies, basic GUI programming, system administration.

Publications

Posters

Anton Burtsev, Mike Hibler, Jay Lepreau. Aggressive Server Consolidation through Pageable Virtual Machines, *Student Research Poster Competition, School of Computing, University of Utah, Utah, March 2008, Awarded 2nd place*. Abstract [[PDF](#)], poster [[PDF](#)].

Anton Burtsev, Prashanth Radhakrishnan, Mike Hibler, Jay Lepreau. True Time-Sharing in Emulab through Preemption and Stateful Swapout, *4rd Symposium on Networked Systems Design and Implementation*, Boston, May 2007. Abstract [[PDF](#)], poster [[PDF](#)].

Anton Burtsev, Prashanth Radhakrishnan, Mike Hibler, Jay Lepreau. Time Travel for Closed Distributed Systems, *3rd Symposium on Networked Systems Design and Implementation*, San Jose, May 2006. Abstract [[PDF](#)], poster [[PDF](#)].

Reports

Anton Burtsev, Prashanth Radhakrishnan. Time-Travel for Closed Distributed Systems. Informal report, Flux Research Group, University of Utah, May 2006 [[PDF](#)].

Anton Burtsev. Implementation of μ ITRON Embedded Operating System Specification on top of L4 Microkernel. Report Paper for *DiSy: Operating Systems, Embedded and Distributed Systems Research Group*, University of New South Wales, Sydney, Australia, December 2004 [[PDF](#)].

Earlier work published in Ukraine (2000-2004)

A.Y. Burtsev, L.B. Ryzhyk, Y.A. Timoshenko. Automatic object serialization mechanism for a distributed component model. *Proc. 4rd International Conference of Students, Postgraduate Students and Young Scientists on System Analysis and Information Technologies*, Kyiv, Ukraine, 2004.

L.B. Ryzhyk, A.Y. Burtsev. Architectural design of E1 distributed operating system, to appear in *System Research and Information Technologies International Scientific and Technical Journal*, October 2004 (This is a shortened version of a technical report available on-line [[PDF](#)]).

A.Y. Burtsev, L.B. Ryzhyk, Y.A. Timoshenko. Operating system for parallel distributed computing. *Proc. 1st International Radio-Electronics Forum "Applied Radio-Electronics. State and Future Trends"*, Kharkiv, Ukraine, 2002.

A.Y. Burtsev, L.B. Ryzhyk, Y.A. Timoshenko. Operating system architecture based on distributed objects. *Scientific bulletin of NTUU "KPI"*, #4, pp.5-12, 2002 [[PDF](#)].

A.Y. Burtsev, L.B. Ryzhyk. Operating system architecture with process migration support. *Proc. 3rd International Conference on Programming "UkrProg2002"*, Kyiv, Ukraine, pp.516-521, 2002. Abstract [[PDF](#)].

A.Y. Burtsev, L.B. Ryzhyk. Operating system for parallel computing. *Proc. 3rd International Conference of Students, Postgraduate Students and Young Scientists on System Analysis and Information Technologies*, Kyiv, Ukraine, pp.23-27, 2001. Conference Theses [[PDF](#)].

A.Y. Burtsev, L.B. Ryzhyk. Agent-oriented approach to software development. *Proc. 2nd National Conference of Students, Postgraduate Students and Young Scientists on System Analysis and Information Technologies*, Kyiv, Ukraine, pp.130-134, 2000. Abstract [[PDF](#)].

Theses

A.Y. Burtsev (in co-authorship with Leonid Ryzhyk). Distributed operating system architecture based on distributed

shared objects, *M.Sc. thesis*, National Technical University of Ukraine “Kyiv Polytechnic Institute”, 2002. Abstract [[PDF](#)].

A.Y. Burtsev (in co-authorship with Leonid Ryzhyk). Intelligent software agents, *B.Sc. thesis*, National Technical University of Ukraine “Kyiv Polytechnic Institute”, 2000.

Talks

Manycorers: Challenges for OS. Given at the [Multi-Core Discussion Colloquium](#), School of Computing, University of Utah, October 2007. [[PDF](#)]

Implementation of μ ITRON Embedded Operating System Specification on top of L4 Microkernel. Given at DiSy: Operating Systems, Embedded and Distributed Systems Research Group, University of New South Wales, Sydney, Australia, November 2004. [[PDF](#)]

The E1 Distributed Operating System Project. Given together with Leonid Ryzhyk at DiSy: Operating Systems, Embedded and Distributed Systems Research Group, University of New South Wales, Sydney, Australia, October 2004. [[PDF](#)]

Personal info

Born February 8, 1979. Ukrainian citizen. Language proficiency: Russian (native language), Ukrainian (native language), English (fluent), German (basic proficiency).

Other interests

Jazz music, reading, painting, taekwondo, mountain biking, skiing.