Modeling and Extracting Load Intensity Profiles

Jóakim von Kistowski, Nikolas Herbst, Samuel Kounev,
Daniel Zoller, Andreas Hotho

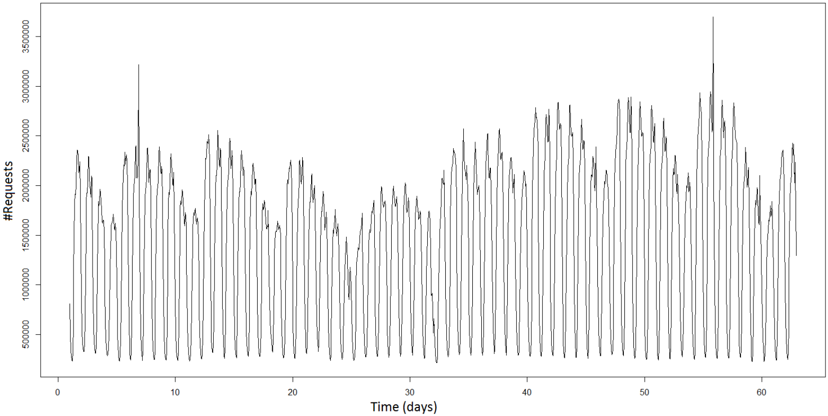
Chair for Software-Engineering, University of Würzburg

SEAMS 2015, Firenze, Italy
May 19th, 2015

http://descartes.tools/limbo
http://research.spec.org/tools/limbo

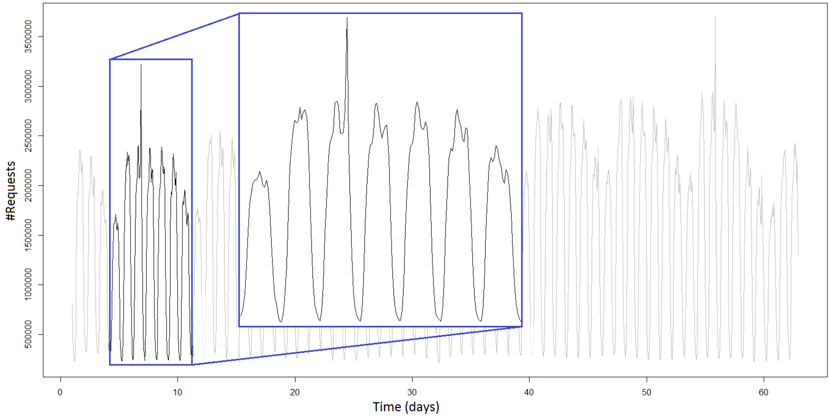
Motivation

- Page Requests for the German Wikipedia



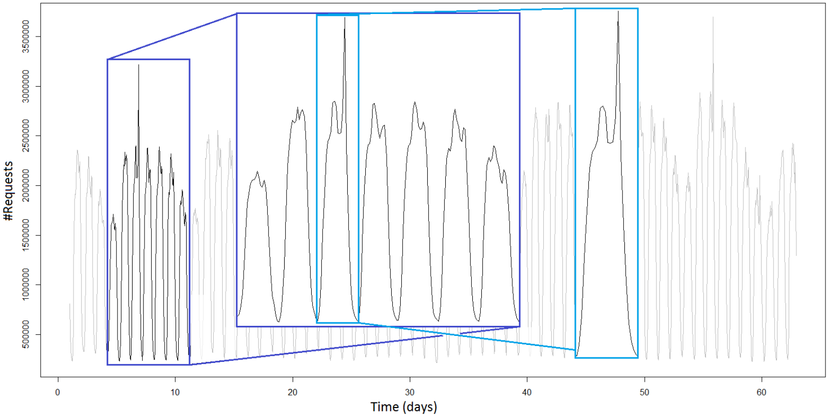
Motivation

- Page Requests for the German Wikipedia

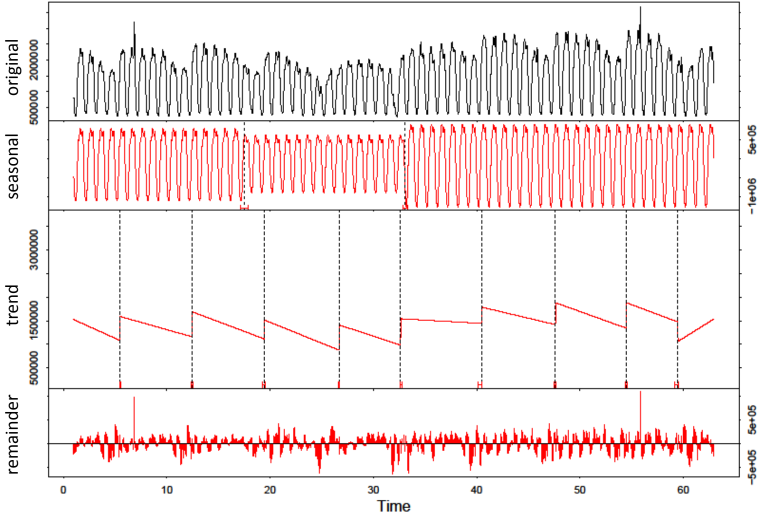


Motivation

- Page Requests for the German Wikipedia



Motivation



Additive decomposition into seasonal part, trend, and remainder. Created using *BFAST* [1].

Related Work

User Behavior Models (e.g. using Markov Chains)

- van Hoorn et al. (2008): probabilistic, intensity-varying workloads
- Roy et al. (2013): workload volatility of a streaming system

Workload Models

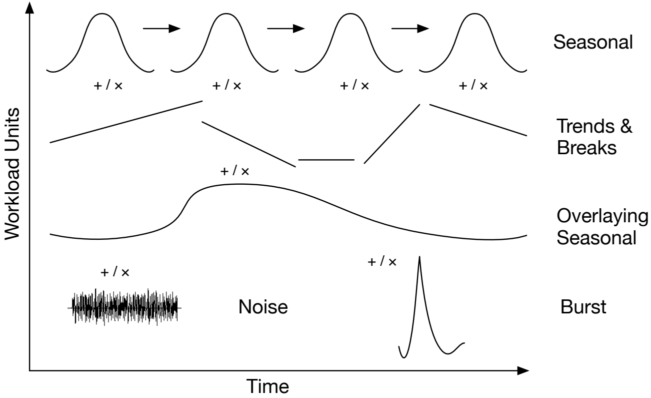
- Barford et al. (1998): file popularity and distribution (web)
- Casale et al. (2012): bursts
- Beich et al. (2010): data popularity and user classes (cloud)

Statistical Models

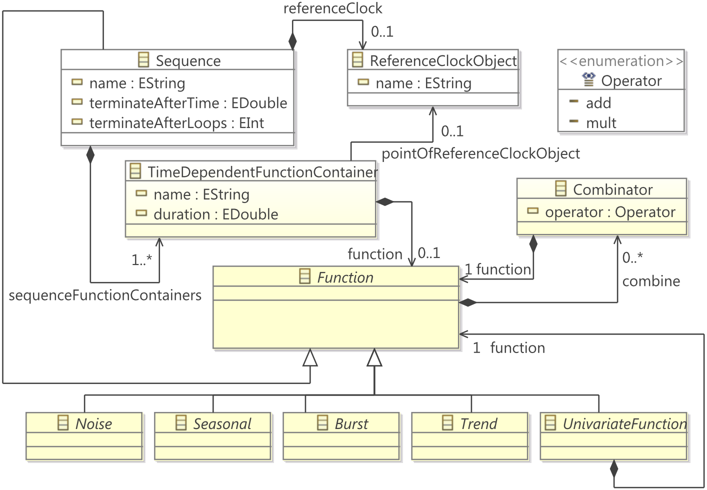
- Feitelson (2002): workload representativity through statistical characteristics

Descartes Load Intensity Model (DLIM)

- Describes arrival rate variations over time
- Provides structure for piece-wise mathematical functions
- Independent of work/request type

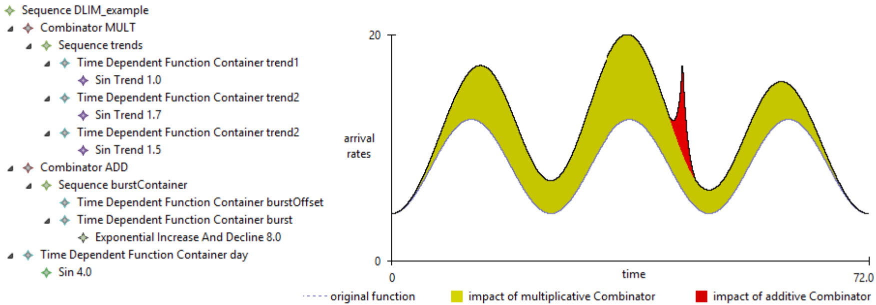


Descartes Load Intensity Model (DLIM)



DLIM Example Instance

- Created using LIMBO eclipse plugin [1]
- Contains *Seasonal* part, *Trends*, and *Burst*



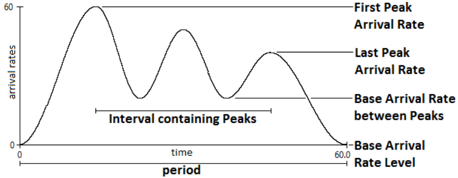
[1] LIMBO: `http://descartes.tools/limbo`

high-level DLIM

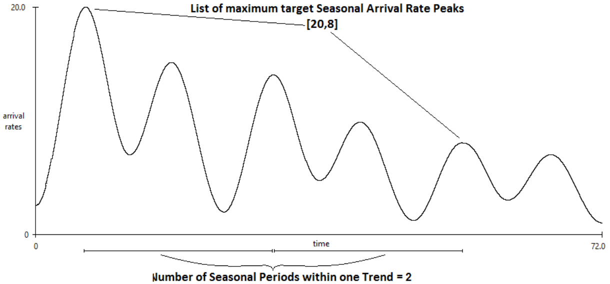
- Benefits of DLIM:
  - Powerful and expressive
  - Easy derivation of arrival rates or request time-stamps
- Drawbacks of DLIM:
  - Instances can become complex
  - Large trees may be unintuitive
- Solution: **high-level DLIM**
  - Fewer parameters for load intensity profile description
  - Strictly structured into single *Seasonal*, *Trend*, recurring *Burst*, and *Noise* parts

hl-DLIM Seasonal and Trend parts

- hl-DLIM *Seasonal* part:

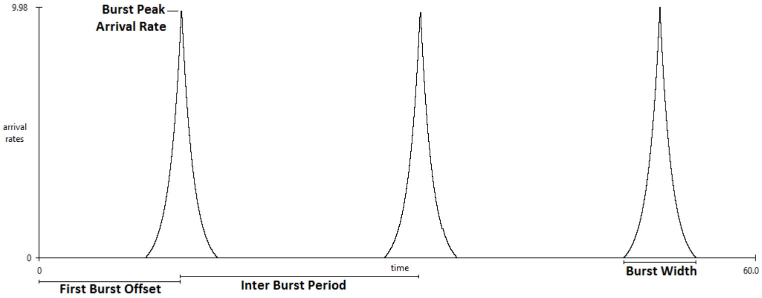


- hl-DLIM *Trend* part:



hl-DLIM Burst and Noise parts

- hl-DLIM *Burst* part:



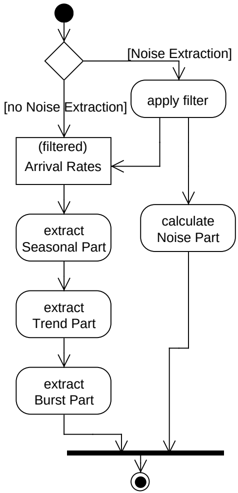
- hl-DLIM *Noise* part:

Uniform Distribution

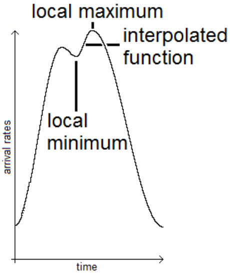
- Minimum Noise Rate
- Maximum Noise Rate

Automated Model Instance Extraction

- Automated process for extracting DLIM or hl-DLIM instances from existing arrival rate traces
- Structured into *Seasonal*, *Trend*, *Burst*, and *Noise* part extraction
- Noise reduction and extraction is optional and separate

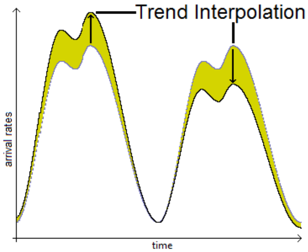


Automated Model Instance Extraction II



- **Seasonal Part**:
  - Extracts median local min/max within *Seasonal* iterations
  - Interpolates using DLIM *Functions*

- **Trend Part**:
  - Adds at each maximum *Seasonal* Peak to trend-list



Automated Model Instance Extraction III

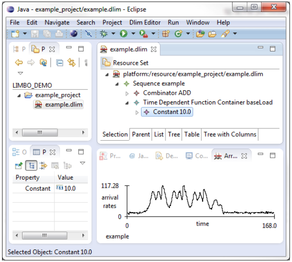
- **Burst Part**:
  - Bursts are detected at strong positive deviations from predicted *Seasonal* and *Trend* behavior
  - Peak is set to match arrival rate in trace
- **Noise Part**:
  - Before Extraction: High frequencies are reduced using a gaussian filter
  - After Extraction: Reduced noise (normal) destribution is added to model instance

Automated Model Instance Extraction IV

- **Simple DLIM Extraction Process (s-DLIM)**:
  - Uses a single Trend-List to describe one overlying *Trend Part*
  - Extracts a DLIM instance
- **Periodic DLIM Extraction Process (p-DLIM)**:
  - Uses a multiple recurring Trend-Lists to describe repeating trends
  - Extracts a DLIM instance
- **high-level DLIM Extraction Process (hl-DLIM Extraction)**:
  - Modified version of the Simple Model Instance Extraction Process
  - Extracts an hl-DLIM instance

LIMBO

- EMF-based modeling platform
- Uses DLIM for load intensity description
- New Model Creation Wizard based on hl-DLIM
- Allows arrival rate and request time-stamp generation
- Visualizes and compares arrival rate profiles
- Provides automated model instance extraction
- Accepted into the SPEC RG tool repository[a]



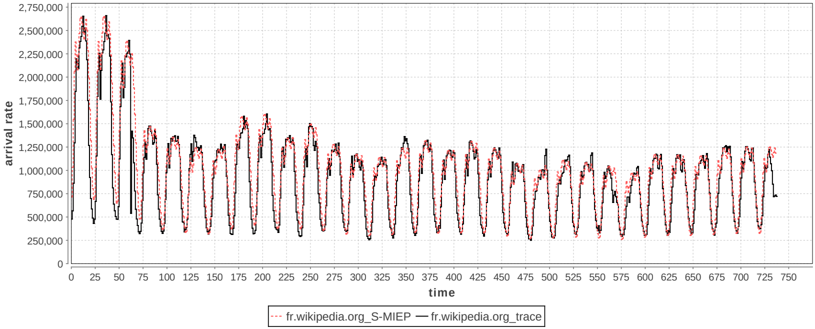
[a] `http://research.spec.org/tools/limbo`

Model Extraction Accuracy Evaluation

- Evaluated using 9 real world web server traces
  - Traces contain tuples of arrival rate / time
  - All traces 2 weeks or longer
  - All traces have human user influence
- Error metric: median arrival rate deviation
- s-DLIM and hl-DLIM extraction applied to all traces
- p-DLIM to traces longer than one month

s-DLIM I

- Most accurate of the extraction processes
- Does not require noise reduction
- Median deviation across all traces: **12.4%**



French Wikipedia s-DLIM result, median deviation: 7.6%

s-DLIM II

Trace	s-DLIM relative median error (%)	BFAST relative median error (%)
ClarkNet-HTTP	12.409	12.243
NASA-HTTP	18.812	-
Saskatchewan-HTTP	26.492	-
WorldCup98	12.979	-
BibSonomy	21.479	-
de.wikipedia.org	8.538	11.223
fr.wikipedia.org	7.6	8.511
ru.wikipedia.org	9.912	5.809
wikipedia.org	4.855	2.302

- s-DLIM performs on average **8354** times faster than BFAST

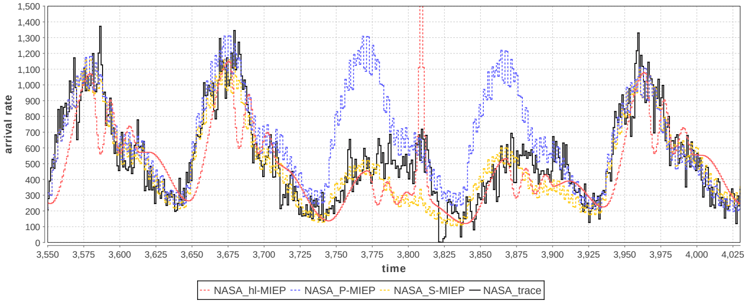
p-DLIM and hl-DLIM

- Both processes are less accurate than s-DLIM
- hl-DLIM: “recurring bursts” - limitation may lead to phantom bursts
- p-DLIM: ignores small deviations from recurring periodic patterns

	s-DLIM	**p-DLIM**	**hl-DLIM**
median deviation (%)	12.4	36.0	15.7

Observations

- **s-DLIM** is the most accurate of the extraction processes
- **p-DLIM** works well for regular load intensity profiles
- **hl-DLIM** heavily relies on noise reduction
- Challenge: Seasonal pattern drift in long traces
  - Extraction uses one seasonal pattern for approximation



Summary

- Two Meta-Models for load intensity variation description
  - **DLIM:** Powerful and expressive
  - **hl-DLIM:** Abstract and concise
- Modeling Platform: **LIMBO**
  - Enables creation of custom load intensity variations for open workload based benchmarking
  - Provides automated load intensity profile extraction
- Automated model instance extraction:
  - **s-DLIM**: most accurate, median deviation: 12.4%
  - **p-DLIM**: good for regular profiles, median deviation: 36.0%
  - **hl-DLIM**: relies on noise reduction, median deviation: 15.7%
- LIMBO is open-source[2] and already being used in different contexts.

[2] LIMBO: `http://descartes.tools/limbo`

Thank you for your Interest!

- Our future work on LIMBO:
  - Extraction of multiple and overlaying seasonal patterns
  - Change detection
  - Advanced calibration and noise reduction
  - Include additional meta-information
- Ideas for integration/extension
  - Extend user-session models to use LIMBO (or reverse)
  - Extend architecture level performance models (PCM / DML) to use DLIM instances or LIMBO timestamps
  - Use DLIM models for improving anomaly detection accuracy

References

J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor, "Detecting trend and seasonal changes in satellite image time series," *Remote Sensing of Environment*, vol. 114, no. 1, pp. 106 – 115, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S003442570900265X

N. R. Herbst, S. Kounev, A. Weber, and H. Groenda, "BUNGEE: An Elasticity Benchmark for Self-Adaptive IaaS Cloud Environments," in *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015)*, May 2015.

A. van Hoorn, M. Rohr, and W. Hasselbring, "Generating probabilistic and intensity-varying workload for web-based software systems," in *Proceedings of the SPEC international workshop on Performance Evaluation: Metrics, Models and Benchmarks*, ser. SIPEW '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 124–143. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-69814-2_9

S. Roy, T. Begin, and P. Goncalves, "A complete framework for modelling and generating workload volatility of a vod system," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, 2013, pp. 1168–1174.

P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '98/PERFORMANCE '98. New York, NY, USA: ACM, 1998, pp. 151–160. [Online]. Available: http://doi.acm.org/10.1145/277851.277897

G. Casale, A. Kalbasi, D. Krishnamurthy, and J. Rolia, "Burn: Enabling workload burstiness in customized service benchmarks," *IEEE Transactions on Software Engineering*, vol. 38, no. 4, pp. 778–793, 2012.

A. Beitch, B. Liu, T. Yung, R. Griffith, A. Fox, and D. A. Patterson, "Rain: A workload generation toolkit for cloud computing applications," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-14, Feb 2010. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-14.html

D. Feitelson, "Workload modeling for performance evaluation," in *Performance Evaluation of Complex*