Who needs humans to run computers?

Role of Big Data and Analytics in running Tomorrow's Computers illustrated with Today's Examples

Maris van Sprang, 2015
IBM, The Netherlands

Agenda

- Preface
- Grand Challenges
- Autonomous computer
- Roadmap
- Conclusions

Preface / Disclaimer

- This presentation is my view
- It is based on experience I have with my clients
- I have combined it with more general work inside and outside IBM
- It is not an IBM official view (but I am working on it)

Agenda

- Preface
- Grand Challenges
- Autonomous computer
- Roadmap
- Conclusions

Grand Challenges from IBM

- IBM Grand Challenges
  - Goal is attractive but is obviously difficult
  - Just-not-impossible
  - Multi-year effort
  - Cross organization: IBM Research, Software, Hardware and Services
  - SMART goal, to register progress and to determine success
  - Not directly for a client, so pure investment
  - Even if it is not successful, 'frontiers of the possible will have been pushed outward'

Chess program Deep Blue defeats world champion Kasparov in 1997

- For decades, the chess world was divided whether a computer would *ever* be able to beat the human world champion
- Hardware was not fast enough for brute force algorithms
- Software was not good enough to mimic the way humans play chess
- Deep Blue was built by IBM as a Grand Challenge and defeated world champion Gary Kasparov in 1997
- Since then hardware has shrunk from 1 full rack to a '*mobile phone*'

Watson defeats human top Jeopardy! players in 2011

- Grand Challenge in 2005 to make a computer that understands Natural Language sufficiently to answer spoken questions (NLP) and to beat human Jeopardy! top players
- Cognitive computer able to deal with incomplete, conflicting or partial incorrect input data
- Watson had to 'learn' about many different knowledge fields: sports, geography, history, arts, etc
- Internally launched in 2005, progress was slow but in 2011 Watson was strong enough to beat the human champion players
- Its original size of 10 racks POWER 750 servers with 2880 cores has shrunk to only 3 *'pizza boxes'*

What might a next Grand Challenge from IBM be?

- Many possibilities are conceivable:
  - A computer that you can have discussions with
  - Intelligent computer understanding to some point what you want
  - Sentient computer that is aware of its surroundings
  - Self-conscious computer
  - Human computer implants
  - …….
  - How about a computer that …..

That keeps running
- when it hasn't yet finished its tasks
- without too much human attention
- is a little bit more robust than today's computers

→ Autonomous

In this presentation focus is on Non-Functional aspects of Autonomous computers

Functionality

- Autonomy with respect to Functionality has to deal with many more aspects such as incorrect or incomplete input or application coding errors
- This will require meta information about the purpose of a service

Non-Functional Aspects

- Non-Functional Requirements include many different aspects and in general are perceived as Quality
- Autonomy of Quality implies that computers Self-manage the Quality to the intended / designed level

Non-Functional Requirements
- Availability
- Reliability
- Stability
- Scalability
- Performance
- Security
- Maintainability
- (and others)

Note: The term computer or -sytem in this presentation is meant to include the whole stack: from hardware up to applications.

Agenda

- Preface
- Grand Challenges
- Autonomous computer
- Roadmap
- Conclusions

Q: Why should we want Autonomous computers?

Trading halted for half a day on the biggest US exchange for financial options following an outage caused by software problems.

Airline canceled more than 700 flights and another 765 flights are delayed due to a software outage – Blamed ticketing partner while the real problem was on their end

Not surprisingly, many angry customers poured out their wrath via social networking after the largest video streaming company had a software outage for more than 20 hours

A leading freight company lost $120 million in revenue because IT was unaware that critical warning messages were associated with their key freight delivery application. They were unable to deliver packages for an entire day due to downtime.

Software problem led to two days of downtime at the largest bank in Europe has tarnished their image as the most reliable banking website.

Downtime is more costly than ever before

- ~ 5-7 M$ / hr brokerage,
- ~ 2-3 M$ / hr credit card,
- ~ 660 k$ Mobile CSP,
- ~ 90 k$ / hr Airline Res

The Bottom Line:

In Today's World, the App can never go DOWN!!!

Q: Why should we want Autonomous computers?
A: Because we have to make computers much more reliable

- Society becomes more and more dependent on computer systems and impact of failures has grown
- But these systems have become more complicated and have to operate in an environment that becomes more and more dynamic
- Pressure to build and release new functionality reduces attention for Quality
- Testing cannot cover all conceivable failure modes in the allotted time
- Every human action has some probability of causing a new error e.g. by a typing error or it may be too late

- A step increase is needed in making more robust and reliable computer systems
- One possibility is to use computers to manage computers
- Use self-managing computers and take human activities as much as possible out of the loop

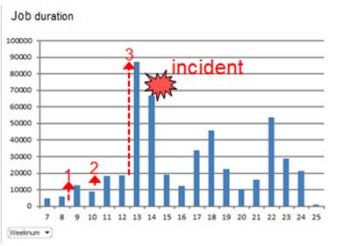
Q: How can we make computers run more reliably?
A: Let's have a look at some of the root causes of failures

- Disks become 100% utilized
- Log files of databases cannot be written anymore
- Maximum number of connections is used
- Buffer space become too small
- CPU utilization becomes too high
- Application changes were introduced but not completely tested. *(but, OK, one only tests for known possible errors and cannot test all potential errors)*
- An incorrect state was reached but no alert was raised because it was not a known failure mode
- Alerts have been raised but not reacted upon (or too late, or incorrectly)

Why have alerts?

A recent change in payments in the Netherlands switched to reading data from chip instead of a magnetic carrier. This led to an unanticipated growth of transaction data causing full file systems

For a full day a bank showed incorrect account- and transaction data to clients because batch processing was delayed too much. Reason: slowed down job.



Agenda

- Preface
- Grand Challenges
- Autonomous computer
- Roadmap
- Conclusions

How to build a self-managing computer system?

- Simple question:
  "Can it be known upfront whether a computer finishes in time, given some input, program and SLA goal?"

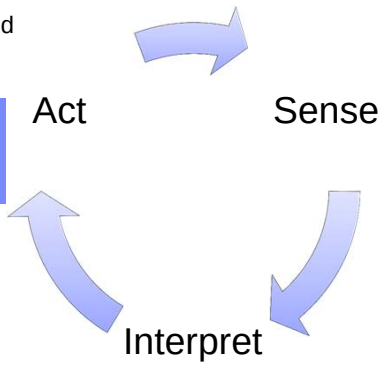
- NO !
  In fact, we even cannot know, in general, if it ever will stop.

But what we can do is make the system like a ***car-navigation system***:
- consider with high frequency what the best route is
- keep the end goal in mind
- stay alert for road conditions and traffic jams

An autonomous computer requires a feedback control loop and …. what recent trends mean for this

3
Actions that are automatically executed make changes to the computer system

Software Defined Environments



1
Data that is sampled from the environment:
- detect threats to quality
- verify effect of actions

Big Data

Analytics

2 Model that interprets data, compares it with target states and translates delta's into corrective actions

1 Developments in Big Data have caused that data size or speed are no limiting factors anymore

Lemma's

1. Without data, one essentially operates blindfolded
2. "Finding a needle in a haystack" is not an excuse for not using the data
3. If one cannot find signals of a particular event in the available data, then he didn't look in the right data and needs to capture more data.
4. More is better

Logs & Monitor output

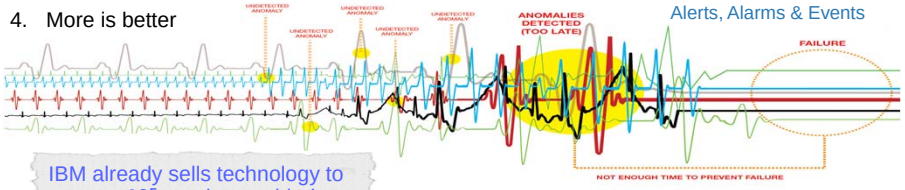
Redbooks

Documentation

Transactions

Performance Metrics

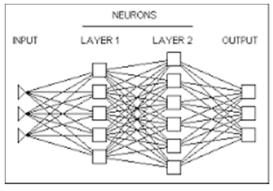
Alerts, Alarms & Events



IBM already sells technology to support 10^5 metrics per blade

2 Analytics is the foundation to interpret data and determine if actions are needed and to assess effects of taken actions

- Fixed rules, such has programming, need to be avoided
- More suitable is Cognitive Computing leveraging Machine Learning techniques, such as Neural Nets
- The system can be in a number of states and the Neural Net determines the actual state while using all metrics as input
- One of the states is "Normal behavior" which is dynamically determined by Behavior Learning methods
- Success of states is determined by SLA's related to throughput, response times, maximum finish times but also by total energy usage. So this is an Optimization problem.
- A second Machine Learning engine determines actions to bring the system from unfavorable states towards favorable states and learns from already taken actions



3 Software Defined Environments (SDE) could be made fully configurable by automated processes

SDE Characteristics

- Software defined abstraction of compute, storage and network resources
- Aggregation of infrastructure heterogeneous resources and sharing among different workloads and tenants
- Infrastructure rapid and automatic reconfiguration to meet functional and non-functional requirements.
- Automatic, policy –based, assignment of resources to workloads
- Integration, automation and optimization of IT resources
- Autonomic and proactive management of workload

Quality aspects

Agile & Flexible

- Workloads and Abstracted infrastructure have configurable capacity & performance characteristics (without reboots)

Optimized

- Total capacity of environment is optimally used: maximum number of workloads is served
- Individual workloads are optimized: SLA's are met

Adaptive & Autonomous

- Built in continuous configuration optimization to meet workloads SLA's

Role of Big Data, Analytics and SDE in Maturity Levels towards Autonomic Computing

Maturity Level	1. Foundation	2. Extended Virtualization	3. Extended Automation	4. Integrating Capabilities	5. Autonomic Computing
Goals	• Understand on workload level what happened before or during an incident	• Understand if workloads meet their SLA's. • Determine mapping between workload and environment	• Understand how the SDE environment operates • Improve the SDE environment	• Predict when goals are not met to enable proactive measures • Generate improvement suggestions	• Determine the configuration to optimize the overall system thus balancing SDE- and environment goals.
Rules & Control	• Standards • Fire and Forget	• Policies • Guidelines • Multiple independent goals	• Static thresholds • Partly dependent goals • Track and Adjust	• Non-linear forecasting • Using observed normal behavior	• Multiple conflicting goals • Behavior learning system • Adaptive
Data	• Workload characteristics and their execution	• Workload SLA's. • Environment capabilities.	• Environment goals • Environment behavior • Overall system metrics		• Workload plans
Way of work	• Ad hoc, when needed , manual	• Data collection, ETL and integration is automated. • Results are used during a manual activity.	• Results can be used as part of other automation processes	• Results are used in manual improvement actions.	• Results can be used in an automated way.

The aspects are incremental: each maturity level adds capabilities on top of the lower levels.

Agenda

- Preface
- Grand Challenges
- Autonomous computer
- Roadmap
- Conclusions

Conclusion

- There is a need to make computers run much more reliable
- This can be realized by making computers run autonomous with respect to Quality
- Recent developments in Big Data, Analytics and Software Defined Environments enable running computers more autonomic
- Realizing this goal requires a combined approach in many fields and is not ‘just a new version of software’

> "We can't solve problems by using the same kind of thinking we used when we created them."
>
> Albert Einstein