XenSummit Asia

November 2-3, 2011

Seoul, Korea

Engaging the Xen Developer Community

George Dunlap

george.dunlap@eu.citrix.com

Citrix Systems UK

Sponsored by:

SAMSUNG & LIBERTAS JUSTITIA VERITAS KOREA UNIVERSITY & kt

& github SOCIAL CODING

Introduction

Hardware Vendors

Software vendors

Academic

Public cloud providers

Private cloud providers

Why might people not be involved?

Don’t know what there is to do

Don’t know the benefits of engaging the community

Don’t know how to engage community effectively

Goal:
Enable you to engage developer community

Outline

- Ways that you can contribute to Xen development
- Benefits and costs
- Practical steps
    - Good bug reports
    - Good questions
    - Good patches
- Running an open development project

How can you contribute?

How you can contribute

Testing on your hardware / software

- Submitting bug fixes

Suggesting interfaces / features

- Submitting patches for interface changes

Development

Running an out-of-tree incubator project

Testing

- Need testing on a wide array of hardware / software
  - Test the functionality you use
  - Performance testing as well
- What to test
  - Xen release candidates
  - Linux pvops release candidates
  - xen-unstable (occasionally)

Reporting

Good bug report is good

Better yet: Patches!

Features

Developers don’t see the learning curve

Suggest changes / clarifications to interface

Suggest new features which might be useful to you

Better yet: Patches!

Upstreaming your changes

- Standard practice: Freeze and fork
- Main xen branch moves forward
- What to do with a fork?
  - Stay forked
  - Forward-port patches
  - Back-port new features / fixes
  - Submit upstream

Why not upstream?

- A lot of extra work
- Time constraints
  - Product deadlines
  - Academic deadlines
- Required changes not acceptable upstream

Good bug reports

- What to include
  - Detailed hardware, software, steps to reproduce
  - As much error as you can
  - Serial console
- Wiki page: ReportingBugs

Asking a question is asking someone else to work for you

Asking good questions

- Do as much work as you can yourself first
  - Search Google
  - Read the code
- Context
  - Who are you?
  - What are you trying to accomplish
- Make it specific
- Show how you’ve tried to solve it yourself
- Wiki page: AskingXenDevelQuestions

Sending code upstream

Wiki page: SubmittingXenPatches

Three people to think about

- Person reviewing the patch
  - Does it do the right thing?
  - Are there any mistakes?
- Person scanning through changesets
  - Do I need to look at this patch?
- Archaeologist
  - 6 months, 1 year, 2 years, 5 years
  - Why is the code the way it is now?

Guidelines

- Break your change into a series of patches
  - One logical change per patch
- No regressions
  - Don’t fix a bug in one patch in a later patch!
- Separate clean-up from functional changes
  - Even if it’s just a one-line change
- One-line summary easy to scan
- Description says what the patch does and why

Requirements

- Patch must apply
  - Beware of mailer mangling
  - Mercurial PatchBomb extension
- Signed-off-by
- Coding style

Interpreting responses

American Culture: “Praise sandwich”

Hacker culture: Direct and to the point

- “I don’t think this is the right way to do this”
- “This is ugly”
- “This is unmaintainable”

Detailed criticism means “I want this patch accepted”

Ignoring is much worse than criticism

Open development

Project lifecycle

- Initial code dump
  - Basic functionality and framework
- Incubation
  - Active development towards maturity
  - Growing community development
- Project maturity
  - Ready to be used by others
  - Actively maintained
- Retired / Archived

Types of projects

- In-tree
  - “Tech preview”
  - Fairly self-contained
  - Example: Paging, page sharing, Remus
- Out-of-tree
  - Major changes
  - Example: xen-arm

Open development: Theory

- Criteria
  - Movement towards upstreaming
  - Encouragement of a development community
- Principles
  - Anyone should be able to influence
  - Influence should correlate to contribution

Open development: Practice

- One public shared repository
- Handful of committers
  - One to begin with
  - Committers added based on contribution
- All patches posted and discussed publicly
- Technical decisions made in public
  - Private discussions must end with public proposal

Outline

- Ways that you can contribute to Xen development
- Benefits and costs
- Practical steps
  - Good bug reports
  - Good questions
  - Good patches
- Running an open development project

Goal:
Enable you to engage developer community

Questions?